DEEP-LEARNING-BASED SHORT-TERM PHOTOVOLTAIC POWER GENERATION FORECASTING USING SELF-ORGANIZATION MAP NEURAL NETWORK AND PROBABILISTIC COMPUTATION

NITIKORN JUNHUATHON

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in Electrical Engineering
Suranaree University of Technology
Academic Year 2022

การคาดการณ์กำลังผลิตไฟฟ้าระยะสั้นจากเซลล์แสงอาทิตย์ด้วยการเรียนรู้เชิงลึกโดยใช้โครงข่ายประสาทเทียมแบบจัดกลุ่มเอง และการคำนวณเชิงความน่าจะเป็น

นายนิธิกร จันทร์หัวโทน

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2565

DEEP-LEARNING-BASED SHORT-TERM PHOTOVOLTAIC POWER GENERATION FORECASTING USING SELF-ORGANIZATION MAP NEURAL NETWORK AND PROBABILISTIC COMPUTATION

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for The Degree of Doctor of Philosophy.

Thesis Examining Committee

Somchat Jiriwibhakorn

(Assoc. Prof. Dr. Somchat Jiriwibhakorn)
Chairperson

(Assoc. Prof. Dr. Keerati Chayakulkheeree)
Member (Thesis Advisor)

(Prof. Dr. Thanatchai Kulworawanichpong)
Member

(Asst. Prof. Dr. Uthen Leeton)
Member

Tosaphol Ratniyomchai

(Asst. Prof. Dr. Tosaphol Ratniyomchai)
Member

Chatchai Jothityangkoon

(Assoc. Prof. Dr. Chatchai Jothityangkoon)
Vice Rector for Academic Affairs
and Quality Assurance

(Assoc. Prof. Dr. Pornsiri Jongkol)
Dean of Institute of Engineering

นิธิกร จันทร์หัวโทน : การคาดการณ์กำลังผลิตไฟฟ้าระยะสั้นจากเซลล์แสงอาทิตย์ด้วยการเรียนรู้เชิงลึกโดยใช้โครงข่ายประสาทเทียมแบบจัดกลุ่มเองและการคำนวณเชิงความน่าจะเป็น (DEEP-LEARNING-BASED SHORT-TERM PHOTOVOLTAIC POWER GENERATION FORECASTING USING SELF-ORGANIZATION MAP NEURAL NETWORK AND PROBABILISTIC COMPUTATION)
อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.กีรติ ชยะกุลคีรี 169 หน้า

คำสำคัญ: การคาดการณ์กำลังผลิตไฟฟ้าจากเซลล์แสงอาทิตย์/การเรียนรู้ของเครื่อง/ความน่าจะเป็น

วิทยานิพนธ์ฉบับนี้นำเสนอ กระบวนการสำหรับการคาดการณ์กำลังผลิตไฟฟ้าระยะสั้นรายชั่วโมงจากเซลล์แสงอาทิตย์ด้วยการเรียนรู้เชิงลึก (Deep Learning, DL) โดยใช้โครงข่ายประสาทเทียมแบบจัดกลุ่มเอง (Self-Organizing Map, SOM) และการคำนวณเชิงความน่าจะเป็น โดยแบ่งการศึกษาออกเป็นสามส่วน ประกอบด้วย การศึกษาเชิงเปรียบเทียบเพื่อหาแบบจำลองและขั้นตอนที่เหมาะสม จากนั้นนำเสนอกระบวนการสำหรับเพิ่มความแม่นยำของเทคนิคการคาดการณ์โดยการที่ขับเคลื่อนด้วยข้อมูลโดยใช้เทคนิคแผนที่ที่จัดการด้วยตัวเองสำหรับจำแนกชุดข้อมูลที่เวลาใด ๆ ที่มีลักษณะความสัมพันธ์ของข้อมูลคล้ายกันให้อยู่ในกลุ่มเดียวกัน เพื่อนำไปเป็นอินพุตของแบบจำลองและนำเสนอการใช้การคำนวณเชิงความน่าจะเป็นเพื่อระบุขอบเขตความผิดพลาดที่อาจจะเกิดขึ้น ในการจำลองเชิงเปรียบเทียบได้แบ่งเป็น 6 กรณีทดสอบ คือ 1) ผลกระทบของการปรับไฮเปอร์พารามิเตอร์ 2) ผลกระทบของฟังก์ชันการเปิดใช้งาน 3) ผลกระทบของการทำให้เป็นมาตรฐาน 4) ผลกระทบของฤดูกาลและการเลือกชุดทดสอบ 5) ผลกระทบของวิธีการตรวจสอบความถูกต้อง และ 6) ผลกระทบของชุดข้อมูลที่ไม่สมบูรณ์ โดยใช้ชุดข้อมูลทดสอบสองชุด โดยกรณีที่ 1 ถึง 5 จะใช้ระบบโซลาร์เซลล์ที่ติดตั้งบนหลังคาโรงงานขนาด 14 MWp และ กรณีที่ 6 ใช้ระบบโซลาร์เซลล์ที่ติดตั้งบนผิวน้ำ ที่มหาวิทยาลัยเทคโนโลยีสุรนารี ขนาด 1.5 MWp จากผลการจำลองการเชิงเปรียบเทียบพบว่ากรณีที่ดีที่สุดของระบบโซลาร์เซลล์ที่ติดตั้งบนหลังคา ที่สุ่มเลือกวันสำหรับทดสอบมา 30 วัน คือ การใช้แบบจำลองโครงข่ายประสาทเทียมแบบหลายชั้น เปอร์เซ็นต์เฉลี่ยความผิดพลาดสมบูรณ์ (Mean Absolute Percentage Error, MAPE) อยู่ที่ 8.504 เปอร์เซ็นต์ ที่ใช้วิธีการแบ่งชุดข้อมูลแบบ 70 เปอร์เซ็นต์แรกสำหรับสอนและ 30 เปอร์เซ็นต์หลัง สำหรับตรวจสอบผล กรณีที่ดีที่สุดของระบบโซลาร์เซลล์แบบติดตั้งบนผิวน้ำ คือ การใช้แบบจำลองโครงข่ายประสาทเทียมแบบหลายชั้นเช่นกันเนื่องจากสามารถรักษาประสิทธิภาพการคาดการณ์ในทุก ๆ กรณี โดยมีเปอร์เซ็นต์เฉลี่ยความผิดพลาดสมบูรณ์อยู่ที่ 19.052 เปอร์เซ็นต์ เนื่องจากข้อมูลมีความแตกต่างค่อนข้างมาก นอกจากนี้

ยังได้นำกระบวนการเพิ่มประสิทธิภาพมาเพิ่มความแม่นยำของการคาดการณ์ด้วยเทคนิคแผนที่ที่จัดการด้วยตัวเอง มาทดสอบกับระบบที่ 1 พบว่าสามารถลดค่าเปอร์เซ็นต์เฉลี่ยความผิดพลาดสมบูรณ์ไปอยู่ที่ 4.90 เปอร์เซ็นต์ และได้ใช้การคำนวณเชิงสถิติเพื่อระบุช่วงความน่าจะเป็นของการผลิตไฟฟ้าจากเซลล์แสงอาทิตย์ได้อย่างแม่นยำโดยมีค่าความน่าจะเป็นของการครอบคลุมช่วงเวลาการคาดการณ์ (Prediction Interval Coverage Probability, PICP) เป็น 1 ซึ่งหมายถึงครอบคลุม 100 เปอร์เซ็นต์

สาขาวิชา วิศวกรรมไฟฟ้า
ปีการศึกษา 2565

ลายมือชื่อนักศึกษา .....
ลายมือชื่ออาจารย์ที่ปรึกษา .....

NITIKORN JUNHUATHON : DEEP-LEARNING-BASED SHORT-TERM PHOTOVOLTAIC POWER GENERATION FORECASTING USING SELF-ORGANIZATION MAP NEURAL NETWORK AND PROBABILISTIC COMPUTATION. THESIS ADVISOR: ASSOC. PROF. KEERATI CHAYAKULKHEREE, D.Eng., 169 PP.

Keyword: PV forecasting/Machine learning/Probabilistic

This thesis presents a process for forecasting the short-term hourly power output of Photovoltaic cells using deep learning (DL) and self-organizing neural networks (Self-Organizing Map, SOM), along with probabilistic calculations. The study is divided into three parts: a comparative study to identify suitable models and methods; a solution for increasing the accuracy of data-driven forecasting techniques by utilizing self-organizing mapping to group datasets with similar correlation characteristics as inputs to the model; and the use of probabilistic calculations to determine the extent of prediction error. The comparative study was conducted using six test cases to examine the impact of hyperparameter tuning, activation function, normalization, seasonality, test set selection, and incomplete data sets. Two test datasets were used: a 14 MWp rooftop solar system for cases 1-5 and a 1.5 MWp surface-mounted photovoltaic system at Suranaree University of Technology for case 6. The results showed that the best-performing system was the rooftop solar system, with a multilayer feedforward neural network model producing an average mean absolute percentage error (MAPE) of 8.504% using a 70:30 data split for training and testing. In contrast, the surface-mounted system achieved an average MAPE of 19.052% due to the high variability of the available data. Furthermore, an optimization process was proposed to enhance forecasting accuracy by utilizing SOM techniques. The results demonstrated that the average MAPE could be reduced to 4.90% using the rooftop solar system in case 1.

Additionally, probabilistic calculations were used to accurately identify the probability range of electricity generation from solar cells, with a Prediction Interval Coverage Probability (PICP) of 1 indicating full coverage of the forecasting period with a 100% probability.

School of Electrical Engineering

Academic Year 2022

Student's Signature .....

Advisor's Signature .....

ACKNOWLEDGEMENT

Writing a Ph.D. thesis has been a challenging and fulfilling journey, and it would not have been possible without the support and guidance of numerous people. I would like to express my heartfelt gratitude to all those who have contributed to my research and my personal growth.

Firstly, I would like to thank my supervisor, Assoc. Prof. Dr. Keerati Chayakulkheeree, for their continuous support, guidance, and encouragement throughout my research journey. Their vast knowledge, expertise, and constructive criticism have been instrumental in shaping my ideas and approach toward research. I am grateful for their patience, understanding, and belief in me.

I am also deeply indebted to Assoc. Prof. Dr. Somchat Jiriwibhakorn, Prof. Dr. Thanatchai Kulworawanichpong, Asst. Prof. Dr. Uthen Leeton and Asst. Prof. Tosaphol Ratniyomchai for their valuable insights, encouragement, and assistance throughout my research. Their enthusiasm and motivation have been a constant source of inspiration for me, and I am grateful for their guidance and support.

I would like to acknowledge the financial support provided by Suranaree University of Technology for my research. Without their support, this research would not have been possible.

Lastly, I would like to thank my family, for their support, and encouragement throughout my Ph.D. journey.

In conclusion, I am grateful to all those who have contributed to my research, my personal growth, and my overall success. Thank you all for your support and encouragement.

NITIKORN JUNHUATHON

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

LIST OF TABLES

Table **Page**

LIST OF TABLES (Continued)

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES (Continued)

Figure **Page**

LIST OF ABBREVIATIONS

AEDP 2018	alternative energy development plan 2018-2037
AnEn	analog ensemble
ANN	artificial neural networks
ARD	automatic relevance determination
BR	bayesian regression
CNN	convolution neural network
CEEMD	complementary ensemble empirical mode decomposition
CNQF	nearest neighbor's quantile filter
CV	cross-validation
DL	deep learning
DP	diurnal persistence
DER	distributed energy resource
DQR	direct quantile regression
e-MVFTS	evolving multivariate fuzzy time series
EPPO	energy policy and planning office
FNN	feedforward neural network
GA	genetic algorithms
GEFCom2012	2012 global energy forecasting competition

LIST OF ABBREVIATIONS (Continued)

GPR	gaussian process regression
IGIVA	data-optimized using improved grey ideal value approximation
KDE	kernel density estimation
LS-SVR	least square support vector regression
LightGBM	light gradient-boosting machine
LSTM	long short-term memory
MAE	mean absolute error
MAPE	mean absolute percentage error
MBE	mean bias error
ML	machine learning
ML-FNN	multi-layer feedforward neural network
NARX	nonlinear autoregressive network with exogenous inputs
NN	neural network
NREL	national renewable energy laboratory
NWP	numerical weather prediction
PDF	probability density function
PDPP	partial daily pattern prediction
PDP 2018	Thailand unveiled the s power development plan 2018-2037
PI	prediction interval

LIST OF ABBREVIATIONS (Continued)

PICP	prediction interval coverage probability
PIF	probabilistic irradiance forecasting
PINAW	prediction interval normalized average width
PIT	determining the dependability
PV	photovoltaics
QELM	quantile extreme learning models
QESN	quantile echo state networks
QR	quantile regression
QRF	quantile regression forest
RES	renewable energy sources
RF	random forest
RMSE	root mean square error
RNN	recurrent neural network
SVM	support vector machine
SVR	support vector regression
SOM	self-organizing map
TCM	time correlation modification
TEDA	typicality and eccentricity of data analytics
TS-SOM	tree-structured self-organizing map
WPD	wavelet packet decomposition
XGBoosting	extreme gradient boosting

CHAPTER 1
INTRODUCTION

1.1 Background

The persistent growth of the economy has resulted in a commensurate increase in demand for electricity, necessitating the need for electrical systems that exhibit reliability and quality. This is because a fault in the electrical system could have catastrophic implications for the entire economy of the country. In light of the pressures posed by these challenges and carbon emissions, renewable energy sources (RES) have been widely integrated into power grids. However, the inherently intermittent nature of RES electricity poses several technical issues to the power system, underscoring the need to estimate photovoltaic (PV) output accurately to ensure the dependable functioning and economical dispatch of the power systems. Among the various energy sources available, RES is regarded as one of the most cost-effective and environmentally friendly options for electrical systems. To enhance the stability of the power transmission system, it is imperative to manage the power system to maximize its benefits and increase competition in the energy sector while ensuring maximum reliability and performance. Moreover, the use of RES in the distribution system can reduce the infrastructure costs involved in generating, transmitting, distributing, and using electricity, making it an essential component of the country's economic growth and industrial development. In 2018, Thailand unveiled the Power Development Plan 2018-2037 (PDP 2018) and Alternative Energy Development Plan 2018-2037 (AEDP 2018), which outlines the country's energy future for the next two decades.

The central aim of both plans is to increase the power capacity of distributed energy resources (DER) through RES to reduce pollution from conventional power generation and enhance reliability. The PDP 2018 underscores the need to procure

electricity from renewable energy sources in keeping with the country's remaining renewable energy potential and support the changing behavior of electricity consumers, including disruptive technology in electrical energy. This will occur while adhering to the COP21 agreement and will comprise biomass, biogas, solar, and floating solar, combined with hydropower, and other renewable energy sources to maintain the retail price of electricity. The PDP 2018 also accounts for future energy conservation measures that will be cost-competitive with renewable energy plants and reliable. The new renewable energy power plants and energy conservation measures of PDP 2018 are presented in Table 1.1.

Table 1.1 New renewable energy power plants of PDP 2018

Renewable energy	Contract capacity (MW)	Reliable capacity (MW)
Solar power	10,000	4,250
Biomass	3,376	2,296
Biogas	546	325
Solar floating and hydro power plant	2,725	1,158
Wind power	1,485	189
Industrial waste	44	26
Electrical energy conservation measures	-	4,000
Total as of 2037	18,176	12,244

(Source: PDP2018, EPPO)

AEDP2018 seeks to augment the proportion of renewable and alternative energy in the form of electricity, heat, and biofuels, constituting 30 percent of the total energy consumption by 2037. A detailed breakdown of the energy and alternative energy production targets for each fuel type is provided in Table 1.2.

Table 1.2 Comparison of alternative energy production targets of AEDP 2018

Fuel type	Production capacity (MW)			
	AEDP2015		AEDP2018	
	Goal (MW)	Done (MW)	Goal (MW)	Cumulative demand[1] (MW)
Solar energy	6,000	2,849	9,290	12,139
Solar floating energy	-	-	2,725	2,725
Biomass	5,570	2,290	3,500	5,790
Wind energy	3,000	1,504	1,485	2,989
Biogas	1,280	382	1,183	1,565
Community waste	500	500	400	900
Industrial waste	50	31	44	75
Small hydro energy	376	239	69	308
Large hydro energy	2,906	2,920	-	2,920
Total Capacity (MW)	19,684	10,715	18,696	29,411
Total energy production (GWh)	65,582	37,757	52,864	85,652
Total energy demand (GWh)	326,119	329,119	250,204	250,204
Electricity from RES per demand (%)	20.11	10.04	21.14	34.23
Electricity from RES per final energy (%)	4.27	2.13	3.55	5.75

Note: 1 is Cumulative demand that can be calculated from done from AEDP 2015 plus 2018

(Source: AEDP2018, EPPO)

Table 1.1 and Table 1.2 indicate that power generation from photovoltaics (PV) represents the most substantial component of power capacity that must be augmented due to the necessity to scrutinize and enhance PV system operation technologies.

1.2 Statement of the problem

To realize the objectives outlined in the PDP2018 and AEDP2018, it is imperative to investigate and implement the photovoltaic (PV) power generation forecasting approach in modern power system operation. Multiple approaches have been proposed to achieve this task, broadly classified into three main groups: physical, statistical, and hybrid approaches. While physical and statistical approaches possess varying strengths and drawbacks, physical approaches utilize theoretical simulation models to compute the PV system power generation based on its fundamental design variables. On the other hand, statistical approaches encompass all data-driven methodologies, ranging from conventional statistical modeling to advanced machine learning algorithms. Mayer and Gróf (2021). Subsequently, hybrid techniques were proposed to mitigate the limitations of each method by integrating two distinct methodologies, namely a physical and a statistical approach or multiple statistical models. The literature suggests that statistical approaches are primarily employed for PV power prediction. (Antonanzas et al., 2016). These data-driven methods are reliant on historical irradiance, weather, and production data. Additionally, these models do not necessitate a comprehensive understanding of the PV system's parameters. Nonetheless, the precision of data-driven forecasting is substantially influenced by the quality, resolution, and accuracy of the training dataset, and even state-of-the-art deep learning systems exhibit restricted accuracy if the historical data provided is limited to less than 1-3 years and fails to encompass various weather and PV conditions. (Wang, Qi, & Liu, 2019). The literature has documented various statistical approaches, such as artificial neural networks (ANN) (Liu, Fang, Zhang, & Yang, 2015), and long short-term memory (LSTM) (Kim, Ko, & Kim, 2019) , to predict PV production. However, in practice, the measured data may be incomplete, while the traditional methods rely on the completeness of the PV-generating dataset. According to the Korea Meteorological Administration (Kim et al., 2019), roughly 19.0 percent of data was missing in 2017. A flawed dataset impedes the application of machine learning-based PV forecasting models or significantly reduces the accuracy of forecasting models. Despite the gravity

of the problem of missing data and a dearth of multiple inputs, relatively few research studies have addressed this issue. Furthermore, it is challenging to ascertain the efficacy of any forecasting model in predicting the output close to the actual value. Therefore, techniques to enhance the accuracy of PV power generation forecasting and probabilistic forecasting, which can bridge this gap by delineating the range and probability of electricity production during various periods, are necessary.

As mentioned earlier, it is essential to investigate solutions that can facilitate more accurate PV generation predictions for practical applications. Firstly, widely adopted models will be comparatively analyzed to assess their accuracy using performance matrices. Subsequently, an alternative forecasting framework will be proposed to enhance the efficiency of PV power generation. Finally, probabilistic calculations will be implemented to determine the range of firm PV power generation forecasting for the test system.

1.3 Objective of the study

The objective of PV (photovoltaic) forecasting is to accurately predict the amount of energy that will be generated by a solar PV system at a given time in the future. This information is useful for a variety of stakeholders, including grid operators, energy traders, and solar power plant owners and operators. Accurate PV forecasting helps to improve the reliability and stability of the electricity grid by allowing grid operators to better anticipate fluctuations in renewable energy supply. It can also help energy traders to optimize their energy trading strategies and solar power plant owners and operators to better manage their power output and revenue streams. Overall, the main objective of PV forecasting is to provide reliable and accurate information on the expected energy output of a solar PV system, which can be used to inform decision-making and improve the efficiency and effectiveness of the renewable energy sector.

The primary goal of this study comprises of the following objectives:

1.3.1 To undertake a comparative analysis of commonly employed forecasting models with the aim of ascertaining their accuracy.

1.3.2 To enhance the precision of PV forecasting for a select set of input variables.

1.3.3 To employ probabilistic forecasting techniques in order to determine the reliable range of PV power generation.

1.4 Structure of thesis

The present thesis is composed of six chapters, and the subsequent section provides concise overviews of each chapter.

Chapter I serves as the introductory section of this thesis, wherein the background information, problem statement, objectives, and overall structure of the thesis are outlined.

Chapter II provides an in-depth discussion of the various applications of power forecasting in distribution systems. This includes an overview of machine and deep learning, as well as a review of recent research in photovoltaic power forecasting. The chapter covers both point and interval forecasting methods, including a description of the fundamental workflow from data collection to the implementation of forecasting models. In addition, the chapter explores widely used methods and adopted predictive models. Finally, the chapter concludes with a detailed description of validation methods and performance metrics.

Chapter III presents a comparative study workflow between widely used forecasting model that were mentioned in chapter II, which includes the datasets used for this study, data preprocessing, and visualization techniques. The chapter also explores the impact of hyperparameter tuning on the forecasting models and evaluates the performance of the deployed models.

Chapter IV presents the proposal of a Self-Organizing Map (SOM) to enhance the clustering efficiency of a nonlinear problem. The findings of this study suggest that SOM is an effective method for addressing this type of challenge, as it can depict the relationship between two or more parameters through numerous states. Additionally,

this research proposes an alternate technique for enhancing the performance of a Deep Learning (DL)-based forecasting model with few inputs by utilizing a SOM to estimate an unmeasured and related factor as one of the inputs.

Chapter V outlines the probabilistic forecasting process, which is utilized to attain a high level of accuracy in predicting intervals.

Chapter VI described on conclusion of thesis.

1.5 Thesis overview

The overview of this thesis is illustrated in Figure 1.1.

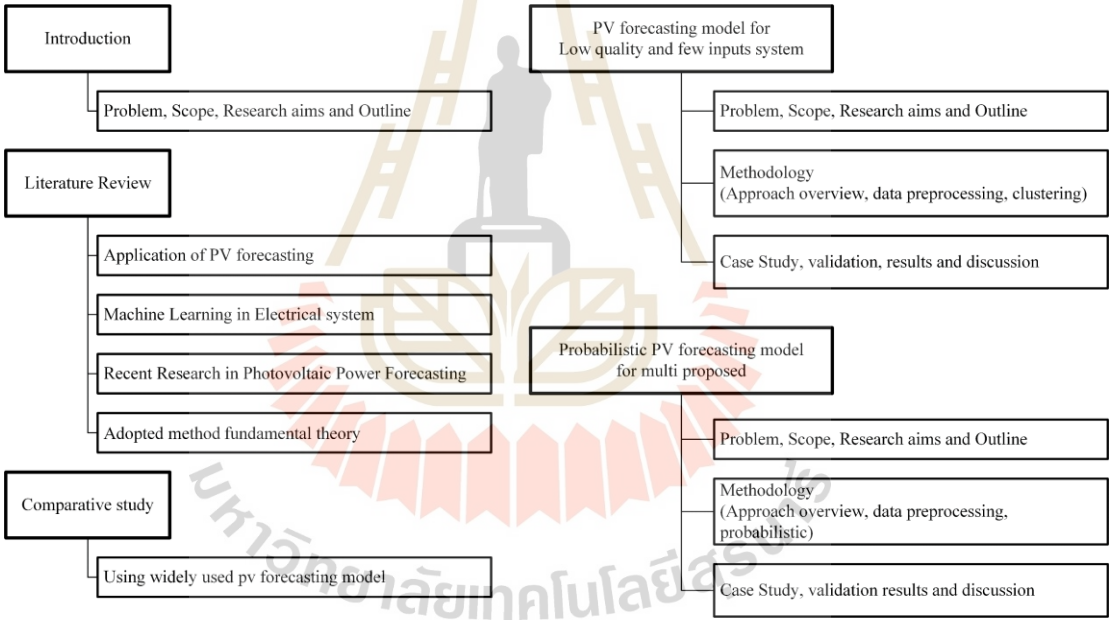


Figure 1.1 Thesis overview

1.6 References

Antonanzas, J., Osorio, N., Escobar, R., Urraca, R., Martinez-de-Pison, F. J., & Antonanzas-Torres, F. (2016). Review of photovoltaic power forecasting. *Solar Energy, 136*, 78-111. doi:https://doi.org/10.1016/j.solener.2016.06.069

Kim, T., Ko, W., & Kim, J. (2019). Analysis and Impact Evaluation of Missing Data Imputation in Day-ahead PV Generation Forecasting. *Applied Sciences, 9*(1). doi:10.3390/app9010204

Liu, J., Fang, W., Zhang, X., & Yang, C. (2015). An Improved Photovoltaic Power Forecasting Model With the Assistance of Aerosol Index Data. *IEEE Transactions on Sustainable Energy, 6*(2), 434-442. doi:10.1109/TSTE.2014.2381224

Mayer, M. J., & Gróf, G. (2021). Extensive comparison of physical models for photovoltaic power forecasting. *Applied Energy, 283*, 116239. doi:https://doi.org/10.1016/j.apenergy.2020.116239

Wang, K., Qi, X., & Liu, H. (2019). A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network. *Applied Energy, 251*, 113315. doi:https://doi.org/10.1016/j.apenergy.2019.113315

CHAPTER 2
LITERATURE REVIEWS

In order to identify and determine the appropriate methodology, a thorough review of related research and methods is necessary. Chapter 2 addresses this need by providing an overview of recent research in several key areas, including the background of power forecasting, the application of PV forecasting, machine learning for electrical systems, recent research in photovoltaic power forecasting, and the fundamental theory behind adopted methods.

2.1 Background of power forecasting

Power forecasting is the process of predicting future electricity production or consumption, typically at a particular location and time. It involves the use of various mathematical and statistical models, as well as data analysis techniques, to estimate the amount of energy that will be generated or consumed over a given time period. Power forecasting is used in a variety of applications, including grid management, energy trading, and renewable energy integration, among others. Accurate power forecasting is critical for ensuring the stability and reliability of the electrical grid and maximizing the efficiency of energy systems. Energy forecasting is a process of predicting the future demand and supply of energy, typically at a particular location and time. It involves the use of various mathematical and statistical models, as well as data analysis techniques, to estimate the amount of energy that will be consumed or produced over a given time period. Energy forecasting can be applied to various energy sources, including electricity, gas, and renewable energy, among others. Accurate energy forecasting is essential for efficient energy management, as it helps energy providers to plan for future energy supply and demand, manage energy prices, and make informed investment decisions. Energy forecasting is used in a variety of

applications, including energy trading, load management, and renewable energy integration, among others. The main difference between power forecasting and energy forecasting lies in the scope of their predictions. Power forecasting specifically focuses on the prediction of electricity production or consumption, while energy forecasting encompasses a broader range of energy sources, such as oil, gas, and renewable energy sources, in addition to electricity. Power forecasting is typically used in the management of electrical grids, energy trading, and renewable energy integration, among other applications, whereas energy forecasting is used in broader energy management contexts, including the planning and optimization of energy supply and demand, the management of energy prices, and the planning of energy infrastructure. In summary, power forecasting is a subset of energy forecasting that specifically focuses on the prediction of electricity production and consumption, while energy forecasting encompasses a wider range of energy sources and applications.

The history of forecasting in electrical systems dates back to the early 20th century when the demand for electricity began to rise rapidly. At that time, electrical utilities needed to be able to predict the amount of electricity that would be needed in order to ensure that they could generate and distribute enough power to meet demand. The history of forecasting in electrical systems can be traced back to the early days of the electricity industry. As the demand for electricity grew and power grids became more complex, accurate forecasting became increasingly important for efficient system operation, planning, and resource allocation. This history can be broadly divided into several key phases, each marked by significant advancements in methodologies, technology, and data collection. In the early days of electricity (late 19th century - early 20th century), During this period, electrical systems were relatively simple, and forecasting techniques were primarily based on simple linear models and intuition. Utilities relied on load duration curves and basic statistical techniques to estimate demand for electricity. For the growth of statistical methods (mid-20th century):

As electricity demand increased and power systems became more interconnected, there was a growing need for more accurate and reliable forecasting methods. Statistical techniques, such as regression analysis and time series analysis, began to be applied to electrical load forecasting. This led to the development of more sophisticated short-term and long-term forecasting models. With the advent of computer technology (the 1960s - 1980s), the introduction of computers revolutionized forecasting techniques in electrical systems. With increased computational power, more complex and efficient forecasting models were developed, including the Box-Jenkins method (ARIMA) and exponential smoothing state space models. Additionally, the advent of digital computers enabled the collection, storage, and processing of large amounts of data, which further improved the accuracy and reliability of forecasting models. For the development of artificial intelligence and machine learning techniques (1990s - early 21st century), during this period, advances in artificial intelligence and machine learning led to the development of new forecasting techniques, such as artificial neural networks (ANNs), fuzzy logic, and support vector machines (SVMs). These methods were capable of capturing complex nonlinear relationships between input variables and offered improved accuracy and adaptability compared to traditional statistical methods. For big data and advanced analytics (2010s - present), The ongoing digital revolution and widespread use of smart grid technologies, such as advanced metering infrastructure (AMI) and phasor measurement units (PMUs), have generated large volumes of data in the power sector. With the help of big data analytics and high-performance computing, advanced forecasting models like deep learning, ensemble methods, and hybrid models have been developed. These models are capable of handling high-dimensional and complex data, leading to improved forecasting accuracy and reliability. Throughout its history, forecasting in electrical systems has evolved in response to the growing complexity and interconnectedness of power grids, as well as advancements in technology and data collection methods. Today, accurate forecasting plays a crucial

role in ensuring the reliable, efficient, and sustainable operation of power systems around the world.

Load forecasting is the process of predicting the amount of electricity demand that will be required at a future time. There are several different types of load forecasting, each with its own approach and methodology. Some of the most common types of load forecasting are: Short-term load forecasting (STLF) predicts electricity demand for a period of up to one week ahead. This type of forecasting is used for day-to-day operation of the power system, such as scheduling generation and transmission resources. Medium-term load forecasting (MTLF) predicts electricity demand for a period of up to one year ahead. This type of forecasting is used for mid-term planning of the power system, such as determining the need for new transmission lines or power plants. Long-term load forecasting (LTLF) predicts electricity demand for a period of more than one year ahead. This type of forecasting is used for long-term planning of the power system, such as developing energy policies and making investment decisions. Peak load forecasting: Peak load forecasting predicts the maximum amount of electricity demand that will occur during a specific time period, such as a day or a week. This type of forecasting is used to plan for the highest levels of demand and ensure that there is enough generation and transmission capacity available to meet it. Weather-sensitive load forecasting takes into account weather patterns and other external factors that can affect electricity demand, such as holidays and special events. This type of forecasting is used to plan for the impact of weather conditions on demand, such as high levels of air conditioning use during heatwaves. Customer load forecasting predicts the electricity demand for individual customers or groups of customers. This type of forecasting is used to plan for the needs of specific customer segments, such as industrial or residential customers. Spatial load forecasting is a type of load forecasting that predicts the amount of electricity demand for different geographical areas within a power system. This type of forecasting takes into account the unique characteristics of each area, such as population density, weather patterns, and economic activity. Spatial load forecasting is important because electricity demand

can vary significantly from one area to another, depending on factors such as the types of customers in the area, the time of day, and the season. By predicting the demand for each area, utilities can plan for the resources needed to meet that demand and ensure that there is enough capacity available to avoid power outages and other disruptions. Average load forecasting is a type of load forecasting that predicts the average amount of electricity demand for a particular time period, such as an hour, day, or week. This type of forecasting is important because it provides a baseline for estimating the resources needed to meet expected demand. Average load forecasting is typically based on historical data, which is used to identify patterns and trends in electricity demand. For example, a utility might analyze electricity demand data from the same day in previous years to identify trends in demand based on factors such as weather patterns, economic activity, and population growth. Overall, load forecasting is a critical tool for ensuring the reliable and efficient operation of the power system. By predicting future electricity demand, utilities can plan for the resources needed to meet that demand and ensure that there is enough capacity available to avoid power outages and other disruptions.

Wind power forecasting is the process of predicting the amount of electricity that will be produced by wind turbines at a future time. This is important for ensuring the reliable and efficient operation of wind power systems, as well as for integrating wind power into the larger electrical grid. The background of wind power forecasting can be traced back to the early days of wind energy research when scientists began to explore the potential of wind power as a renewable energy source. In the early days, wind power forecasting was done manually, using simple statistical techniques to predict the amount of energy that would be produced by a wind turbine in a given time period. As wind power systems became more common in the 1980s and 1990s, computer-based forecasting models began to be developed. These models used more sophisticated techniques such as time series analysis and neural networks to predict wind energy production. The introduction of these models made it possible to make more accurate forecasts, which in turn helped to improve the reliability and efficiency

of wind power systems. During the 2000s and 2010s, the focus of wind power forecasting shifted from just predicting energy production to also predicting the impact of weather patterns on energy production. This required the development of new forecasting techniques that could take into account the variable nature of weather patterns and their impact on wind energy production. Today, wind power forecasting plays a critical role in the operation of wind power systems. By predicting the amount of energy that will be produced at a future time, utilities and system operators can plan for the resources needed to meet that demand and ensure that there is enough capacity available to avoid power outages and other disruptions. As wind power continues to grow as a source of renewable energy, it is likely that even more advanced forecasting techniques will be developed in the future.

Photovoltaic (PV) forecasting is the process of predicting the amount of electricity that will be produced by a solar power system at a future time. This is important for ensuring the reliable and efficient operation of solar power systems, as well as for integrating solar power into the larger electrical grid. The background of PV forecasting can be traced back to the early days of solar energy research when scientists began to explore the potential of solar power as a renewable energy source. In the early days, PV forecasting was done manually, using simple statistical techniques to predict the amount of energy that would be produced by a solar panel in a given time period. As solar power systems became more common in the 1980s and 1990s, computer-based forecasting models began to be developed. These models used more sophisticated techniques such as time series analysis and neural networks to predict solar energy production. The introduction of these models made it possible to make more accurate forecasts, which in turn helped to improve the reliability and efficiency of solar power systems. During the 2000s and 2010s, the focus of PV forecasting shifted from just predicting energy production to also predicting the impact of weather patterns on energy production. This required the development of new forecasting techniques that could take into account the variable nature of weather patterns and their impact on solar energy production. Today, PV forecasting plays a critical role in

the operation of solar power systems. By predicting the amount of energy that will be produced at a future time, utilities and system operators can plan for the resources needed to meet that demand and ensure that there is enough capacity available to avoid power outages and other disruptions. As solar power continues to grow as a source of renewable energy, it is likely that even more advanced forecasting techniques will be developed in the future.

In conclusion, power forecasting is a critical tool for ensuring the reliable and efficient operation of power systems. By predicting the amount of electricity demand or production at a future time, utilities and system operators can plan for the resources needed to meet that demand and ensure that there is enough capacity available to avoid power outages and other disruptions. The history of power forecasting can be traced back to the early 20th century when simple statistical techniques were used to predict electricity demand. With the advent of computers, more sophisticated forecasting methods were developed, including artificial intelligence and machine learning algorithms. In recent years, the focus of power forecasting has shifted to the integration of renewable energy sources such as solar and wind power, which require new forecasting techniques that can take into account the variable nature of these energy sources. Overall, power forecasting is an important tool for ensuring the reliability and sustainability of power systems, and its importance is likely to grow as renewable energy sources become more common.

2.2 Applications of Power forecasting

The utilization of photovoltaic (PV) forecasting in distribution systems offers a range of benefits, including capacity firming, battery size determination, and energy market management. In a recent study by (Keerthisinghe, Mickelson, Kirschen, Shih, & Gibson, 2020), different forecasting techniques such as persistent forecasts, long short-term memory (LSTM), encoder-decoder LSTM, and multi-layer feed forward neural network (ML-FNN) were compared to firm capacity. The study found that both ML-FNN and LSTM, which are deep learning (DL) models, outperformed the conventional

model by reducing the annual battery energy throughput the most. Similarly, (Beltran, Cardo-Miota, Segarra-Tamarit, & Pérez, 2021) utilized DL-based irradiance forecasts to determine battery size for solar capacity stabilization, demonstrating that this methodology increases the predictability of PV output and enables PV capacity firming. (Visser, AlSkaif, & van Sark, 2022) proposed a comparison of day-ahead solar power forecasting algorithms for PV systems with variable geographical distribution, which revealed that DL-based forecasting methods outperformed other models in terms of both performance and economics. These findings highlight the potential of short-term PV forecasting models to enhance grid benefits, with ML-based forecasting models demonstrating superior performance compared to other methods. Consequently, the application of ML in electrical systems and PV forecasting will be discussed in subsequent sections.

In power system planning studies, the time horizon can be classified into long-term and short-term. Long-term planning studies typically involve generation and transmission expansion planning, policy development, and investment decisions over several decades. On the other hand, short-term planning studies focus on issues such as unit commitment, economic dispatch, power flow, and day-ahead markets, with a time horizon of up to one year (Seifi & Sepasian, 2011). Therefore, the duration of forecasting can be defined as shown in Tables 2.1 to 2.3.

Table 2.1 Classification of load forecasting methods according to the time period (Wang, Guo, & Huang, 2011)

Methods	Time horizon	Applications
very short-term load forecasting	few mins	- distribution's schedule - generation forecasting
short-term load forecasting	few hours	- distribution's schedule - generation forecasting
medium-term load forecasting	few days to a month	- seasonal load forecasting
long-term load forecasting	>1 year	generation growth planning

Table 2.2 Wind generation forecasting methods according to the period (Soman, Zareipour, Malik, & Mandal, 2010)

Time horizon	Range	Applications
very short-term	1 sec to 30 minutes	- electricity market clearing - regulation actions
short term	30 minutes to 6 hours ahead	- economic load dispatch planning - load increment/decrement decisions
medium-term	6 hours to 1 day ahead	- generator online/offline decisions - operational security in the day-ahead electricity market
long-term forecasting	1 day to 1 week or more ahead	- unit commitment decisions - reserve requirement decisions - maintenance scheduling to obtain optimal operating cost

Table 2.3 PV generation forecasting methods according to the period (Akhter, Mekhilef, Mokhlis, & Mohamed Shah, 2019)

Time horizon	Range	Applications
very short-term	1 sec to 1 hr	- real-time electricity dispatch - optimal reserves - power smoothing
short term	1 hr to 24 hr	- increase the security of the grid
medium-term	1 week to 1 month	- maintains the power system planning and maintenance schedule by predicting the available electric power shortly
long-term forecasting	1 month to many years	- helps in electricity generation planning, transmission, and distribution authorities in addition to energy bidding and security operations.

This thesis categorizes the forecasting horizon into four different time periods, namely Nowcasting (Intra hour), Short-term forecast (Intraday), Medium-term forecast, and Long-term forecast. Intra-hour forecasting requires investigation of a few seconds to several minutes and is crucial for real-time decision-making, particularly in applications such as distributed load dispatching and energy storage planning. In networks with high penetration of renewable energy, short-term forecasting using renewable energy resources and related storage enhances grid stability, especially in unexpected islanding/fault scenarios. Although numerical weather prediction (NWP)-based methods suffer performance degradation, typically, historical and/or meteorological information are used for nowcasting assessments. Recent advancements in image processing of captured sky images have the potential to yield promising results. Short-term forecasting estimates PV power generation for up to seven days, enabling unit commitment, rescheduling, and dispatch of electricity

supply, making it beneficial in constructing a PV-integrated energy management system. It also improves grid operation security. Medium-term forecasting uses a time frame of more than a week to a month, enabling the projection of the future availability of electric power to facilitate power system design and maintenance schedule. Long-term forecasting estimates PV power generation from one month to one year and is useful for planning electricity production, transmission, and distribution organization and for energy bidding and security operations.

2.3 Machine/Deep Learning for forecasting

Machine learning (ML) is a subfield of artificial intelligence (AI) concerned with the evolution of systems that can be trained or improved by the data they receive. The term "machine learning" was coined in 1959 by American scientist Arthur Lee Samuel, who defined it as "a field of study devoted to the study of a computer's ability to learn without being explicitly programmed". ML is a data-driven approach that enables mechanisms to evaluate information without explicit programming. DL, also known as deep learning, is one of the neural networks distinct from conventional artificial neural networks (ANNs) in that it consists of multiple hidden layers, intricate interconnection structures, and various transition operators. In recent decades, several machine-learning architectures have been developed, facilitating the proliferation of deep learning. With the increasing use of deep learning in various fields, numerous techniques and algorithms have been developed for training DLs (Kingma & Ba, 2015). DL can effectively perform without feature engineering, which is the process of extracting significant features from the dataset, and this is a key difference between ML and DL. Generally, DL models require substantially more data to be effectively trained. As previously mentioned, machine learning is a statistical method that captures insights from a dataset without being primarily ordered. However, the ability to do so requires an information source on which the model is "trained". Following this initial step of information extraction from statistical information, the machine learning model can be used to provide accurate forecasts/insights throughout the process,

which is known as the "inference" mode. Training data must be appropriately standardized/normalized to provide the appropriate features that enable the artificial neural network (ANN) to be efficiently trained. ML/DL methods can be applied in various categories in the electrical system or related topics. Nonetheless, each ML/DL method has different strengths and drawbacks. To select the appropriate model that works well with the research objective, in-depth details of recent research on the PV power forecasting model will be discussed in the next section.

2.4 Recent Research in Photovoltaic Power Forecasting

In order to develop a comprehensive understanding of photovoltaic power forecasting, a review paper will be examined as presented in Table 2.4. The review paper aims to analyze and discuss the most popular and effective methods utilized in this field. Photovoltaic (PV) power forecasting is a critical aspect of renewable energy generation, and its importance has grown exponentially in recent years due to the increased deployment of solar panels worldwide. Accurate PV power forecasting is essential for optimal energy management, grid integration, and maintenance scheduling, among other purposes. To achieve this, a variety of forecasting techniques have been developed, ranging from conventional statistical models to advanced artificial intelligence (AI) and machine learning (ML) techniques. These techniques have become increasingly popular due to their effectiveness in dealing with the inherent complexities of PV power forecasting, such as weather fluctuations and solar panel degradation. The review paper will provide a comprehensive overview of the various methods utilized in the field of PV power forecasting and will serve as a useful reference for researchers, practitioners, and policymakers seeking to understand the state-of-the-art techniques in this domain.

Table 2.4 Literature Review of Review Paper in PV forecasting field

Ref.	Notes
(Das et al., 2018)	The study examined the effectiveness of ML and statistical models from historical data. ANNs and SVMs were found to be the most adaptable models, while GAs were preferred for hyperparameter optimization. These findings emphasize the importance of advanced techniques in PV power forecasting for better energy management. The study's results provide valuable insights for renewable energy researchers and practitioners, paving the way for future improvements in PV power forecasting models.
(H. Wang et al., 2020)	This article comprehensively reviews AI-based solutions for solar energy forecasting, evaluating methods such as DL and optimization and identifying obstacles and research goals. These include developing probabilistic prediction models, improving model explainability, and estimating cloud behavior. Continued research is essential for effective solar energy management, and the study provides valuable insights for renewable energy researchers and practitioners. The review's findings pave the way for future advancements in solar energy forecasting.
(Mellit, Massi Pavan, Ogliari, Leva, & Lughi, 2020)	This article reviews publications between 2008 and 2019 on ML, DL, and hybrid models for solar energy output forecasting. The review focuses on point forecasting, with ANNs and SVMs being the most popular methods. These methods enable accurate solar energy output forecasting for efficient energy management and grid integration. The findings provide insights into current techniques for solar energy forecasting, paving the way for advancements in renewable energy.

Table 2.4 Literature Review of Review Paper in PV forecasting field (continue)

Ref.	Notes
(Carrera & Kim, 2020)	This study compares various forecasting approaches for PV power output using a 1.5-day prediction horizon. ML models are evaluated with a k-fold cross-validation process and grid search to find optimal hyperparameters. Data from weather forecasts and observations validate each model. The XGBoosting algorithm outperforms others due to its ability to handle non-linear relationships and interactions between input variables. This study provides insights for researchers and practitioners in renewable energy, improving PV power output forecasting models.
(Rajagukguk, Ramadhan, & Lee, 2020)	This study evaluates three DL approaches for forecasting solar irradiance and PV power production: LSTM, CNN+LSTM, and ED-LSTM models. The CNN+LSTM model performs best in predicting solar irradiance and PV power production. The study emphasizes the importance of using RMSE for comparing outcomes. RMSE provides a reliable error rate measure, enabling researchers and practitioners to assess model accuracy. The study's findings provide insights for developing accurate and reliable forecasting models for solar irradiance and PV power production, contributing to optimal energy management and grid integration.

Table 2.4 provides an overview of various machine learning (ML) and deep learning (DL) models proposed for PV power generation forecasting, as well as the preprocessing techniques used in these studies. The reviewed articles demonstrate that ML and DL models outperform conventional techniques in short-term PV power forecasting, and provide insight into which methods are most effective for this task. Table 2.4 presents a concise summary of the key findings. In this thesis, we will discuss the literature review on PV power generation forecasting in two groups: 1) point-forecast ML-based methods and 2) interval (or probabilistic)-forecast ML/DL-based

methods. The review will analyze and compare the strengths and limitations of these different methods, providing valuable insights for researchers and practitioners seeking to develop accurate and reliable PV power forecasting models. The aim of this analysis is to contribute to the ongoing efforts to improve renewable energy management and grid integration, ultimately enabling a more sustainable energy future.

Table 2.5 Reviews on PV power generation forecasting

Ref.	**Forecasting horizon & Resolution**	**Parameters (Historical & forecast)**	**Method & notes**
Point-forecast ML-based methods			
(Fekri, Ghosh, & Grolinger, 2020)	Not applicable	Not applicable	The R-GAN has been employed to generate realistic datasets suitable for training energy forecasting models.
(Li, Zhou, Lu, & Yang, 2020)	One hour ahead ,5 min	Metrological data Power production	In this study, a comparison is conducted between a hybrid deep learning (DL) model that integrates wavelet packet decomposition (WPD) and DL models.

Table 2.5 Reviews on PV power generation forecasting (continue)

Ref.	Forecasting horizon & Resolution	Parameters (Historical & forecast)	Method & notes
(Niu, Wang, Sun, Wu, & Xu, 2020)	1–150 steps ahead ,5 min	Metrological data Power production	In order to generate ultimate forecasts, an artificial neural network (ANN) constructs a hybrid model by leveraging historical photovoltaic (PV) power data that have been decomposed using the Complementary Ensemble Empirical Mode Decomposition (CEEMD) algorithm, and weather information that has been selected through the use of a random forest (RF) approach and optimized using the Improved Grey Ideal Value Approximation (IGIVA) method.
(H. Zhou et al., 2019)	1–8 steps ahead ,7.5 min	Metrological data Power production	An ensemble model is constructed for temperature and power series, comprising two Long Short-Term Memory (LSTM) models, each equipped with attention mechanisms.

Table 2.5 Reviews on PV power generation forecasting (continue)

Ref.	Forecasting horizon & Resolution	Parameters (Historical & forecast)	Method & notes
(S. Zhou, Zhou, Mao, & Xi, 2020)	1–4 weeks ,10 min	Irradiance Power production	In order to address the issue of limited training data, a technique based on sequential model-based optimization is employed to optimize hyperparameters of LSTM model that features shared-optimized layers. Furthermore, transfer learning is incorporated into this LSTM model, with a source domain consisting of historical solar irradiance data and a target domain comprised of power production data.
(Severiano, Silva, Weiss Cohen, & Guimarães, 2021)	1,2 and 8 steps , every 15 min	Solar energy & Wind energy	In this study, a mechanism known as TEDA is utilized in conjunction with Evolving Multivariate Fuzzy Time Series and Data Analytics to distinguish typicality and eccentricity. The pyFTS module in Python was employed to implement the model, which offers a novel

Table 2.5 Reviews on PV power generation forecasting (continue)

Ref.	Forecasting horizon & Resolution	Parameters (Historical & forecast)	Method & notes
			means of detecting concept drift.
(F. Wang et al., 2020)	Day ahead ,15 min	Direct normal irradiance (DNI) and temperature Power production	In this study, a group or ensemble is created by integrating a Long Short-Term Memory (LSTM) recurrent neural network (RNN) with a Time Correlation Modification (TCM) model. The coefficients of the TCM model are calibrated through the utilization of a partial daily pattern prediction (PDPP) framework.
(Zhao et al., 2021)	One day ahead ,30 min	Metrological data Power production Power from a physical model	The AML model comprises three regression techniques, namely Elastic Net CV regression, Gradient Boosting Regression, and RF Regression. The selection of appropriate features for the region-specific base models is carried out using a modified genetic algorithm (GA) approach. In addition, the forecast of power generation at the base level

Table 2.5 Reviews on PV power generation forecasting (continue)

Ref.	Forecasting horizon & Resolution	Parameters (Historical & forecast)	Method & notes
			is enhanced through the incorporation of a physical model, thus elevating the performance of the final model.
(Chang, Li, & Zomaya, 2020)	1–12 steps ahead ,30 min	Metrological data	In order to cluster weather patterns, Light Gradient-Boosting Machine (LightGBM) models were employed in tandem with temporal pattern aggregation and Time Series Self-Organizing Map (TS-SOM) techniques. The resulting approach demonstrated noteworthy performance not only in terms of accuracy but also in relation to training and inference time, even on edge devices.
(Hossain & Mahmood, 2020)	12 to 24 h , hourly	Metrological data Power production	In this study, the Long Short-Term Memory (LSTM) model leverages a synthetic irradiance forecast that is generated using a k-MEANS classification algorithm. This approach results in a 33%

Table 2.5 Reviews on PV power generation forecasting (continue)

Ref.	Forecasting horizon & Resolution	Parameters (Historical & forecast)	Method & notes
			improvement in accuracy compared to the use of an hourly sky forecast, and a 44% enhancement relative to utilizing a daily sky forecast.
(Pan & Tan, 2019)	1–24 steps ahead hourly	Metrological data Power production	In this study, an ensemble approach is employed for Random Forest (RF) models, utilizing ridge regression, in addition to preliminary cluster analysis of weather predictions.
(Leva, Dolara, Grimaccia, Mussetta, & Ogliari, 2017)	1–24 steps ahead , hourly	Metrological data Irradiance measurement Power production	In this study, the clear sky model was utilized to pre-process data prior to its utilization with the Artificial Neural Network (ANN) model. Furthermore, various periods of the year were examined, and the simulation outcomes were analyzed for both partially cloudy and cloudy days.
(Nkuriyingoma & Selcuklu, 2021)	Depending on metrological data	Metrological data from the station	In order to achieve this objective, Nonlinear AutoRegressive with

Table 2.5 Reviews on PV power generation forecasting (continue)

Ref.	Forecasting horizon & Resolution	Parameters (Historical & forecast)	Method & notes
	, hourly	Irradiance measurement Power production	eXogenous input (NARX) models were proposed. The activation function, also known as the transfer function, plays a crucial role in adjusting the output amplitude of the neural network model, as it is responsible for translating the input signals into the corresponding output signals. The most commonly utilized transfer functions include sigmoid (logistic and hyperbolic tangent), linear, and Gaussian.
(Lateko, Yang, & Huang, 2022)	Depend on metrological data (7 days) , hourly	Metrological data Power production	In this study, Linear regression, Support Vector Regression (SVR), and an ensemble of trees were compared with the proposed forecasting methods. The proposed method employed a combination of Support Vector Machines (SVM) and K-means, resulting in higher performance compared to

Table 2.5 Reviews on PV power generation forecasting (continue)

Ref.	Forecasting horizon & Resolution	Parameters (Historical & forecast)	Method & notes
			the other compared methods. To achieve this task, Gaussian Process Regression (GPR) was proposed, with consideration given to some unique inputs that were tested using correlation analysis.
(Fen et al., 2017)	Depend on metrological data (3 days) , hourly	Metrological data , Clearness index , Hourly temperature difference, Sunshine duration, Power production	In this study, Linear regression, Support Vector Regression (SVR), and an ensemble of trees were compared with the proposed forecasting methods. The proposed method employed a combination of Support Vector Machines (SVM) and K-means, resulting in higher performance compared to the other compared methods. To achieve this task, Gaussian Process Regression (GPR) was proposed, with consideration given to some unique inputs that were tested using correlation analysis.

Table 2.5 Reviews on PV power generation forecasting (continue)

Ref.	Forecasting horizon & Resolution	Parameters (Historical & forecast)	Method & notes
Interval-forecast ML/DL-based methods			
(du Plessis, Strauss, & Rix, 2021)	1–6 h ahead (21 steps) 15 min	Weather sensor data PV power data	In this study, a comparison was conducted between Artificial Neural Network (ANN), Long Short-Term Memory Recurrent Neural Network (LSTM-RNN), and Gate Recurrent Unit Recurrent Neural Network (GRU-RNN) models.
(Carriere, Vernay, Pitaval, & Kariniotakis, 2020)	30 min–36 h ahead 30 min	Historical Weather Historical power Forecast altitude & azimuth sun position	The Analog Ensemble (AnEn) model utilizing Numerical Weather Prediction (NWP), satellite, and in situ data was employed to forecast results for a horizon ranging from 5 to 36 hours.
(Wen et al., 2020)	1,3,6 h ahead	Historical Weather Historical power	In this study, a hybrid model was constructed by combining Radial Basis Artificial Neural Networks (RBANN) with Particle Swarm Optimization (PSO). The Prediction Interval (PI) was determined through the utilization of Bootstrap with Quantile Regression (QR)

Table 2.5 Reviews on PV power generation forecasting (continue)

Ref.	Forecasting horizon & Resolution	Parameters (Historical & forecast)	Method & notes
			results. It was found that the utilization of Bootstrap resulted in superior PI reliability diagrams.
(Huang & Wei, 2020)	1–24 h ahead Hourly data	Historical Weather Historical power	To address the non-differentiable loss functions problem associated with Quantile Convolutional Neural Networks (QCNN), a two-stage training strategy was implemented in this study.
(Najibi, Apostolopoulou, & Alonso, 2021)	1–24 h ahead Hourly data	Historical Weather Historical power azimuth sun position	In this study, Gaussian Process Regression (GPR) was utilized with the Matern 5/2 kernel function on pre-clustered data (using k-means clustering).

While a range of Artificial Neural Network (ANN) architectures and other Machine Learning (ML) techniques have been utilized in this field, earlier research has predominantly focused on shallow architectures such as multilayer perceptron (MLP) networks. In recent years, however, there has been a shift towards more advanced Deep Learning (DL) techniques, such as Long Short-Term Memory (LSTM) networks. In order to accurately evaluate the performance of forecasting models, an understanding of performance metrics is necessary. The most commonly employed performance

measurements are the mean absolute error (MAE) and the root mean square error (RMSE). It is important to note that the scenario under consideration has a significant impact on the model's performance. It would be unfair to compare outcomes from frameworks applied to different scenarios, where the scenario encompasses various attributes of the plant under independent inquiry (e.g., dimension, structure in terms of the number of strings, cell type, etc.), environmental factors, the size of the training and testing datasets, and feature preprocessing and/or extraction. This holds true not only for absolute measurements such as MAE or RMSE but also for relative performance measures such as Mean Average Percentage Error (MAPE), which are more suitable for comparing the results of models across different plants. A detailed description of the measures utilized to evaluate models can be found in (Zhang et al., 2015).

Over the past few years, numerous machine learning frameworks have been developed to simplify the process of developing and deploying machine learning models in production. Many of these frameworks support AML (Automated Machine Learning), which is a technique that enables the automatic selection, training, and optimization of a machine learning model, or an ensemble of machine learning models. In a recent study conducted by (Zhao et al., 2021). an AutoML approach was proposed for creating an ensemble that utilized an improved genetic algorithm (GA) optimization technique to select the best attributes for each region. The proposed approach combined historical data from photovoltaic (PV) plants, weather data, and the output of a physical model to forecast generated power, utilizing features such as tilted solar irradiance, PV panel temperature, and ambient temperature. The dataset covered 2016 and 2017 and was recorded every thirty minutes. The researchers trained a multi-regional model using Elastic Net CV regression, Gradient Boosting Regression, and RF Regression, which was subsequently applied to data from different plant locations. The study is one of the few to examine the effectiveness of AutoML in forecasting PV output, and interestingly, the models used in the ensemble have not been widely used in the industry before. Historical data from PV power plants in

Hokkaido, Japan were used to train the models from January 1, 2016 to December 31, 2017, but only one month (December 2017) was utilized for testing purposes.

In a recent study by (Severiano et al., 2021). , the aspect category problem was introduced for the first time in the field of energy forecasting, as evidenced by a review of relevant literature. Although the focus of the study was on the forecasting of solar and wind energy, the methods employed in the research are potentially applicable in the context of PV energy generation, and the study utilized a public dataset. The research utilized the Evolving Multivariate Fuzzy Time Series (e-MVFTS) approach to forecast time series and evaluated its effectiveness in the context of solar and wind energy by utilizing a publicly available dataset from the United States National Renewable Energy Laboratory (NREL) for solar energy data, and the 2012 Global Energy Forecasting Competition (GEFCom2012) for wind energy data, which is now accessible through the Kaggle platform. The proposed method integrates a forecasting model based on Fuzzy Time Series with an evolving clustering method based on Typicality and Eccentricity Data Analytics (TEDA), enabling it to adapt to concept drift that occurs in time series and to automatically handle changes in the data distribution.

(Li et al., 2020) proposed a novel combination method of Wavelet Packet Decomposition (WPD) and Long Short-Term Memory (LSTM) networks in their research. This method incorporates historical information regarding power and weather but does not consider future irradiance predictions in the model. The WPD technique is applied to a photovoltaic power series to generate four new sub-series, which are then fed into individual LSTMs. The outputs from each LSTM are combined using linear weighting to produce the final forecast. Moreover, each LSTM generates sequential forecasts.

In a study conducted by (Liu, Zhao, Wang, Sun, & Wennersten, 2019), historical photovoltaic power data, past weather conditions, and artificially generated weather forecasts using k-means clustering were utilized to develop multi-step predictions with an LSTM network. The results indicate that the proposed LSTM model outperformed

RNNs, GRNNs, and ELM models. In comparison to utilizing an hourly sky forecast, the accuracy of the model was found to improve by 33-44.6% when a synthetic irradiance prediction was employed.

The concept of Transfer Learning (TF) was first introduced in the field of photovoltaic (PV) generation forecasting, as proposed by (S. Zhou et al., 2020), In Deep Learning (DL), TF is a commonly used technique where pre-trained DL model, which is complex and successful, is employed to transfer its domain knowledge to a new but similar domain. In the context of image classification/recognition, TF for Convolutional Neural Networks (CNN) has been extensively utilized. The initial layers of a CNN can learn basic features of image collections, such as edges, shapes, and textures. Only the last one or two layers of a CNN are responsible for the most complex classification of vectorized visual input. This approach is more efficient than freezing the weights of early layers and training only the last layers for a specific task in the target domain. In PV power forecasting, TF involves transferring data from a pre-trained LSTM model, which is trained on historical irradiance time series, to a PV power time series to overcome the lack of data in the target domain. The study concluded that TF can be extremely advantageous for a new plant lacking sufficient historical data.

(Chang et al., 2020) propose an ensemble technique, LightGBM, along with a Bayesian optimization algorithm to determine the optimal time steps for temporal pattern grouping, and a clustering-based training framework based on a tree-structured self-organized map (TS-SOM) for short-term forecasting of photovoltaic power output. The effectiveness of this approach is demonstrated in a power generation environment that includes an edge computing platform (Raspberry Pi 3B). Utilizing historical weather conditions, the proposed model consists of three functional steps: Bayesian-optimized temporal pattern aggregation, weather clustering using TS-SOM, and model training with LightGBM. The authors demonstrate that their proposed approach outperforms well-known deep learning alternatives such as GRNN and LSTM by significantly reducing both training and inference time.

(Niu et al., 2020) propose a hybrid machine learning-based approach for short-term forecasting of PV generation capacity. The approach involves using an RF model to rank the input weather-related features, followed by an Improved Grey Ideal Value Approximation (IGIVA) model that uses the RF outcomes as weight values to identify similar days of various meteorological types and enhance the data for training. Subsequently, a Complementary Ensemble Empirical Mode Decomposition (CEEMD) methodology is used to decompose the original power series, and an Artificial Neural Network (ANN) is trained using the dynamic factor Particle Swarm Optimization (DIFPSO) method to create short-term PV power forecasts.

There has been a relatively low number of studies focusing on probabilistic forecasting in recent years, compared to studies on point forecasting. Global forecasting challenges like the M3 and M4 challenges have contributed to the development of probabilistic forecasting techniques, highlighting concepts such as Prediction Intervals (PI) and probability coverage, and introducing measurements like pinball loss that are more appropriate for this type of forecasting. Interested readers can refer to (Hong et al., 2016; Hyndman, 2020; Makridakis, Spiliotis, & Assimakopoulos, 2018). for further information on these forecasting challenges.

In their study, (du Plessis et al., 2021) have introduced a novel approach for point prediction with a confidence interval (CI) that considers uncertainties in available forecasts. The CI is computed using a bootstrap method based on expected changes and the level of certainty for each forecast. It is worth noting that CI and prediction interval (PI) are distinct concepts, with CI being smaller than PI. The primary focus of this study is on short-term forecasting for a range of 1-6 hours. The proposed method is unique in its application to a large-scale multi-megawatt PV system (specifically a 75 MW plant with 84 inverters), where a macro-level modeling approach provides a slight improvement in accuracy compared to the conventional inverter-level modeling approach.

In their study, (Najibi et al., 2021), employ a Multi-Layer Feed Forward Neural Network (ML-FNN), a long short-term memory (LSTM) network, and a gated recurrent unit (GRU) in their proposed model. The authors conduct a probabilistic analysis of the accuracy of a Gaussian process regression model with Matérn 5/2 kernel function using the same criteria for confidence interval (CI) estimation. The proposed model, like many in the field of photovoltaic output prediction, employs weather data and past photovoltaic output as inputs. The data is clustered into four groups based on solar output and time using k-means clustering. The authors validate their proposed model using data from five different PV plants with both a five-fold cross-validation technique and a hold-out process with 30 randomly selected test days.

The studies by (Carriere et al., 2020; Huang & Wei, 2020) aim to develop accurate probabilistic solar output forecasting models that emphasize prediction intervals (PI). In addition to the traditional point forecastings metrics like RMSE and MAE, the researchers introduce PI coverage probability and prediction interval normalized average width (PINAW) as new metrics to evaluate the reliability of predictions and the width of the PIs. The research considers an hourly day-ahead forecasting horizon and uses a CNN-based quantile regression (QR) approach with a two-stage training strategy to address the non-differentiable loss function of QR. The proposed model outperforms other models like quantile extreme learning models (QELM), quantile echoes state networks (QESN), direct quantile regression (DQR), and RBML-ML-FNN.

The researchers in (Wen et al., 2020) investigated probabilistic forecasting by proposing a hybrid model that involves a wavelet transform applied to historical power output, followed by an RBML-FNN trained using the PSO approach for point prediction. To calculate the prediction interval (PI), the indirect bootstrap method is utilized. The performance of the proposed model is evaluated against the direct and indirect quantile regression (QR) approaches using reliability diagrams. The comparison

demonstrates that the bootstrap method is crucial for identifying the best-performing model.

The study conducted by (González Ordiano, Gröll, Mikut, & Hagenmeyer, 2020) evaluates a novel approach for probabilistic forecasting using data from the 2014 Global Energy Forecast Competition (GEFCom2014). The proposed approach, known as the nearest neighbor's quantile filter (CNQF), addresses the challenges associated with training quantile regressions using gradient-based optimization by modifying the training set. The modified training set is then used to train a generic regression model that directly outputs the conditional empirical q-quantile given by the training neighbors. The results indicate that the proposed method achieves pinball loss levels that are comparable to those of the GEFCom14 competition winners, with a difference of less than 1 percent.

Based on the literature review, several forecasting horizons and sampling have been researched. However, for the purpose of this study, the most applicable forecasting horizon is days ahead, with an hourly sampling frequency. This is because the required input parameters can be obtained from readily available weather data sources. Additionally, such a forecasting horizon and sampling frequency can be useful in the energy market and capacity-firming applications. Regarding the input parameters for the forecasting model, the available meteorological data is commonly used with data-driven methods such as Artificial Neural Networks (ANN), Nonlinear Autoregressive Networks with eXogenous inputs (NARX), Long Short-Term Memory (LSTM), Linear Regression (LR), Support Vector Regression (SVR), Ensemble learning, Gaussian Process Regression (GPR), and Duranial Persistence (DP). These methods will be further discussed and utilized in the present study.

2.5 Fundamental forecasting workflow

In order to develop a forecasting model, there exist various methodologies. However, in this study, the four-step approach for creating a forecasting model will be

employed, as it provides a clear and systematic framework that is easy to understand and implement. The four steps include data importation, data preprocessing and analysis, predictive modeling, and deployment of forecasting models, as illustrated in Figure 2.1.

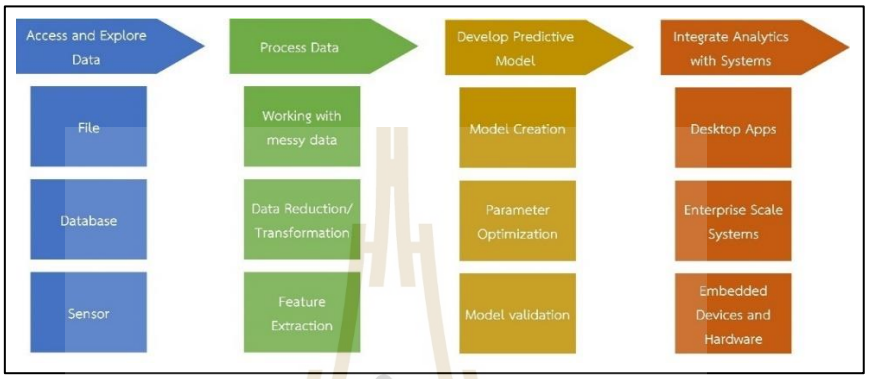


Figure 2.1 Fundamental workflow to create forecasting models

2.5.1 Import data

The initial step in the data analysis process is importing the necessary data, including historical data and real-time data from various sources such as sensors, the web, and databases.

2.5.2 Preprocessing and analyzing data

The next step after importing data is to preprocess and analyze it. The data will be converted to a suitable data type that can be used in the model, such as timetable, table, cell, or struct, among others. Relevant data from various sources will be selected and organized into a dataset for model training, validation, and testing. Additionally, this process involves removing unnecessary data or noise from the dataset, managing missing data, outliers, and resampling irregular data to a uniform format.

Various techniques are utilized to analyze the dataset, such as group summary computations, transform by group, resample or aggregate data in the

timetable, resolve duplicate or irregular times, split data into groups and apply function, and data visualization techniques such as heat maps, geo-bubbles, word clouds, box plots, scatter plots, and exploration of connections.

2.5.3 Predictive modeling

In order to develop the predictive model, it is essential to have knowledge of selecting the appropriate model for addressing the problem at hand. There exist several ways to approach a predictive modeling problem, such as curve fitting, classification, regression, deep learning, system identification, and econometric time-series modeling (e.g., ARIMA, GARCH, etc.), or designing a custom model. Afterward, the data needs to be prepared for machine learning, which involves removing infrequent data, partitioning the data into training and testing sets, and defining validation methods (such as hold-out or cross-validation). The subsequent step involves training and testing the model with the partitioned data and then evaluating it using performance metrics. If the model does not provide the expected results, the hyperparameters of the model should be reconfigured to achieve better outcomes.

2.5.4 Deploying forecasting model

In the process of deploying a forecasting model, the chosen deployment option should align with the desired goals of the project. As highlighted by previous studies, there are numerous options for deploying a model, ranging from desktop applications to web applications, and even generating C and C++ code for deployment on various platforms, including GPUs and FPGAs. Other options include Java, Python, .NET applications, and the MATLAB production server.

The selection of a deployment option should be based on the specific needs of the project, such as the desired level of scalability, performance, and accessibility. For example, if the project requires real-time forecasting with low latency, a desktop or server-based application may be the most suitable option. Alternatively,

if the goal is to make the model accessible to a large number of users, a web application may be more appropriate.

Furthermore, the deployment process should ensure that the model is integrated with the existing infrastructure and systems and that it is updated regularly with new data to ensure that it continues to produce accurate and reliable forecasts. Additionally, the deployment process should also consider the security and privacy implications of deploying the model, and ensure that appropriate measures are in place to protect sensitive data.

Overall, the deployment of a forecasting model is a crucial step in ensuring that the model can be utilized effectively to generate accurate and reliable forecasts. The selection of an appropriate deployment option and the careful consideration of deployment-related factors can greatly impact the success of the project.

2.6 Adopted predictive models

This section presents a comprehensive overview of the various prediction models utilized in this study. The objective of this study is to compare and evaluate the performance of different prediction models for photovoltaic (PV) power generation forecasting. Linear regression is a widely used statistical technique for predicting a numerical value based on a linear relationship between the input variables and the target variable. In this study, linear regression is applied as a baseline model for comparison. Support Vector Regression (SVR) is a machine learning algorithm that uses a nonlinear kernel function to map the input variables to a higher dimensional space, where a linear regression model is then applied. The SVR algorithm aims to minimize the margin of the regression function while still maintaining a certain level of error tolerance. Ensemble learning combines multiple prediction models to improve the overall accuracy of the forecast. This study employs two types of ensemble models, namely, bagging and boosting. Bagging is a technique that combines multiple models

by averaging their predictions while boosting is a technique that combines multiple models by sequentially training new models on the residual errors of the previous models. Deep Learning techniques have shown remarkable performance in a wide range of applications, including PV power generation forecasting. This study utilizes three types of Deep Learning models, namely, and Long Short-Term Memory (LSTM). ML-FNN is a Multi-Layer Feed Forward Neural Network that consists of multiple layers of perceptrons, LSTM is a type of Recurrent Neural Network (RNN) that is designed to handle sequential data with long-term dependencies. In addition to the above models, a benchmark model is also included in the study. This model uses a simple average of the previous day's power output as the forecast for the next day.

Finally, the evaluation of the performance of the forecast models is described. The metrics used for evaluation include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

2.6.1 Linear regression

1) Multivariate Linear Regression

In this research, the first model studied is a Multivariate Linear Regression (MLR) model. This technique is commonly used for solar forecasting due to its simplicity. The MLR model predicts photovoltaic (PV) power output by establishing a linear relationship between a matrix (*X*) consisting of (*n*) predictors and (*m*) timestamps and the power output (y^{mlr}). The model is characterized by a vector of regression coefficients β:

$$\hat{y}^{mlr} = \beta_1 x_1 + \beta_2 x_2 + ... + \beta_i x_i + \varepsilon \,, \tag{2.1}$$

where ε demonstrates the uncertainty, minimizing discrepancies between the actual (*y*) and expected ($\hat{y}^{mlr}$) power output yields β is the coefficients:

2) Linear regression with interact

An interaction effect exists in regression when the influence of an independent variable on a dependent variable varies with the value(s) of one or more other independent variables.

$$\hat{y}^{mlr} = \beta_1 x_1 + \beta_2 x_2 + ... + \beta_i x_i + \beta_3 x_1 x_2 + \beta_{i+1} x_i x_1 + \beta_{i+2} x_i x_2 + \varepsilon \,, \qquad (2.2)$$

3) Robust linear regression

Robust linear regression is a preferred approach to standard linear regression as it is less sensitive to outliers. Standard linear regression uses least-squares fitting to determine the model parameters that connect the response data to the predictor data using one or more coefficients. However, outliers can have a significant impact on the fit as squaring the residuals multiplies the impact of extreme data points. This can invalidate model assumptions and result in unreliable parameter estimates, confidence intervals, and other statistics.

Robust regression uses iteratively reweighted least squares to assign weights to each data point, making the technique less susceptible to outliers than conventional linear regression. Weighted least squares incorporate the weight as an additional scale element in the fitting process, which improves the fit. Preexisting weight functions, such as Tukey's bisquare function, can be used to calculate the weights. The iteratively reweighted least-squares method automatically and repeatedly calculates the weights. Initially, the algorithm assigns equal weight to each data point and estimates the model coefficients using ordinary least squares. At each iteration, the algorithm computes the weights, assigning a lower weight to locations that deviated the most from regression models in the previous iteration. The method calculates the model coefficients using the least-squares method, aiming to find the curve that best fits most of the data while minimizing the effects of outliers. The algorithm stops iterating when the estimated coefficient values converge within a

specified tolerance. Table 2.6 provides an overview of the process for the iteratively reweighted least-squares method.

Table 2.6 Reweighted least-squares methods

Step	Descriptions
1	The initial step in robust linear regression is to estimate the weights, followed by utilizing weighted least squares to fit the model.
2	The calculation of adjusted residuals can be expressed as follows: $$r_{adj} = \frac{r_i}{\sqrt{1-h_i}}, \quad (2.3)$$ The calculation of adjusted residuals involves the use of the expression where r_i refers to the least-squares residuals and h_i represents the leverage values for the least-squares fit. The adjustment of residuals is done to reduce the weight of high-leverage data points that have a considerable impact on the least-squares fit.

Table 2.6 Reweighted least-squares methods (continue)

Step	Descriptions
3	When standardized residuals are modified, the resulting standardized modified residuals are given by: $$u = \frac{r_{adj}}{Ks} = \frac{r_i}{Ks\sqrt{1-h_i}}, \quad (2.4)$$ In the equation above, K is a scaling constant and s is an estimation of the standard deviation of the error term, calculated as s = MAD/0.6745, where MAD is the median absolute deviation of the residuals from their median. The constant 0.6745 renders the estimate for the normally distributed independent. If the predictor data matrix *X* includes *p* columns, the program excludes the p absolute deviations with the smallest values while calculating the median.
4	To obtain the robust weights w_i based on *u*, the following equation is used to calculate the weights: $$w_i = \begin{cases} (1-u_i^2)^2 & ,\lvert u_i \rvert < 1 \\ 0 & ,\lvert u_i \rvert \geq 1 \end{cases}, \quad (2.5)$$
5	Estimate the robust regression coefficients β. The weights adjust the following expression for the parameter estimates β as follows $$\beta = \left(X^T W T\right)^{-1} X^T W y, \quad (2.6)$$ where *W* is the diagonal weight matrix, *X* is the predictor data matrix, and *y* is the response vector.
6	Computing the least-squares weighted error $$e = \sum_1^n w_i \left(y_i - \hat{y}_i\right)^2 = \sum_1^n w_i r_i^2, \quad (2.7)$$ where w_i are the weights, y_i is the observed responses, $\hat{\boldsymbol{y}}_i$ are the fitted responses, and r_i are the residuals.
7	If the convergence criteria are met or the maximum allowable number of iterations is reached, the iteration process is terminated.

4) Stepwise linear regression

Stepwise regression is a systematic method of adding and removing predictor variables from a linear or generalized linear model based on their statistical significance in explaining the response variable. This technique involves comparing the predictive ability of progressively larger and smaller models.

Table 2.7 Conclusion of Regression models

Regression model type	Interpretability	Model Flexibility
Linear	Easy	Very low
Interactions Linear	Easy	Medium
Robust Linear	Easy	Very low, less sensitive to outliers, slow to train
Stepwise Linear	Easy	Medium

2.6.2 Support Vector Regression

Support Vector Regression (SVR) is a kernel-based approach used for forecasting, which evolved from the Support Vector Machine (SVM) that is frequently used to solve classification problems. Like SVM, SVR employs hyperplanes to establish the relationship between the predictor and target variables. In the present study, we explore an SVR model with a linear kernel known as least square SVR (LS-SVR), which is represented as follows:

$$\hat{y}^{SVR} = \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) K(x_i, x_j) + b, \tag{2.8}$$

where $\hat{y}^{svr}$ represents the goal variable (PV power output), $(\alpha_i - \alpha_i^*)$ represents the difference between the Lagrange multipliers, and *b* represents the bias. The kernel function for a linear SVR is denoted by $K(x_i, x_j)$ as follows:

$$K(x_i, x_j) = \exp\left(-\frac{\left\| x_i - x_j \right\|^2}{2\sigma^2} \right), \tag{2.8}$$

where σ is a hyperparameter from the manual set.

2.6.3 Gaussian process regression

Gaussian process regression (GPR) models are nonparametric kernel-based probabilistic models. Consider taking the training dataset $\{(x_i, y_i); i = 1, 2, ..., n\}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, where $\mathbb{R}^d$ and $\mathbb{R}$ are unknown distributions. Given the incoming input vector x_{new} and the training data, a GPR model predicts the value of the response variable y_{new}. A linear regression model has the following form:

$$y = x^T \beta + \varepsilon, \tag{2.9}$$

where $\varepsilon \sim N(0, \sigma^2)$. Using the data, the error variance σ^2 and β coefficients are calculated. A GPR model describes the response by incorporating Gaussian process (GP) latent variables, $f(x_i), i = 1, 2, ..., n$ and specified basis functions, *h*. The covariance function of the latent variables represents the smoothness of the response, while the basis functions map the inputs x into a *p*-dimensional feature space, see (Fen et al., 2017) for more details.

2.6.4. Ensembles of Trees

Decision trees are commonly used in both classification and regression tasks to predict responses based on a series of decisions made on input variables. The tree is traversed from the root node to a leaf node, where the predicted response is stored. Nominal responses such as "true" or "false" are provided by classifier trees, while regression trees provide numerical responses. Figure 2.2 illustrates an example of a decision tree.

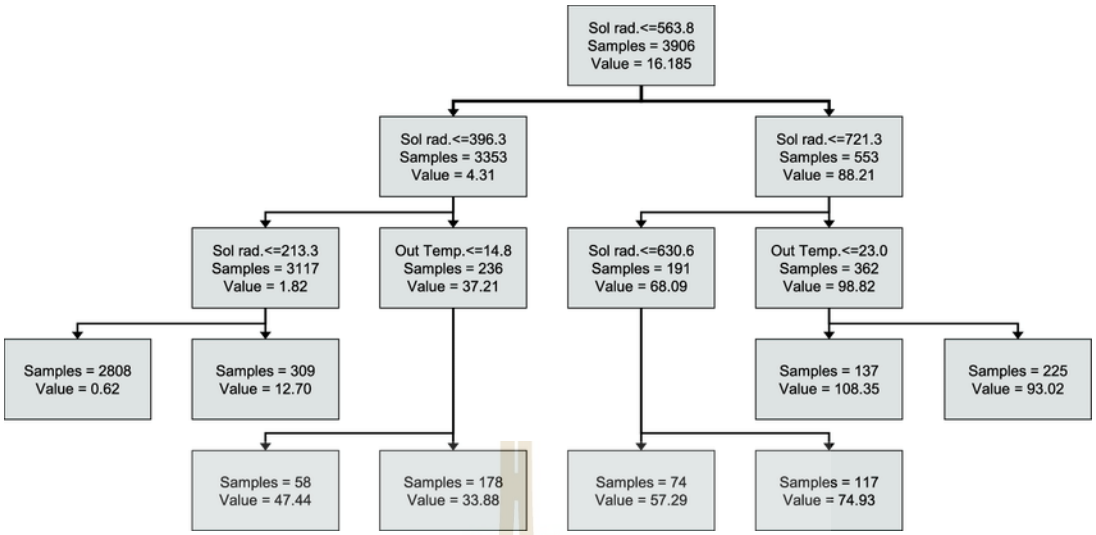


Figure 2.2 Decision tree for predicting energy gain from solar collector (Ahmad, Reynolds, & Rezgui, 2018)

Ensemble methods have been shown to outperform single decision trees in terms of predictive performance. The basic principle behind ensemble methods is the combination of multiple weak learners to create a strong learner. Two commonly used ensemble decision trees are bagging and boosting. Bagging, also known as Bootstrap Aggregation, is utilized when the objective is to minimize the variance of a decision tree. This method involves generating several subsets of data from a training sample that are selected at random with replacement. Each subset of data is then used to train its decision tree, resulting in a collection of distinct models. The forecasts from several trees are averaged to produce a more reliable result than that of a single decision tree.

Boosting, on the other hand, is an ensemble strategy for generating a set of predictors. In this method, learners are taught progressively, with novice models being fit to the data before analyzing it for flaws. In other words, we fit successive trees (random sample) to minimize the net error of the previous tree at each stage. The summary of trees is presented in Table 2.8.

Table 2.8 Summarize Ensembles of Trees

Regression model type	Interpretability	Ensemble method	Model Flexibility
Boosted trees	Hard	Least-squares boosting	Medium to high
Bagged trees	hard	Bootstrap aggregating	High

2.6.5 Deep learning

In addition to the aforementioned models, this research employs three distinct types of Neural Networks (NNs). NNs possess a distinctive architecture that enables them to model intricate nonlinear relationships between input and output variables without the need for any preconceived notions about the relationship between them. The NN architecture consists of an input layer that receives the input data, an output layer that produces the predictions, and a specific number of hidden layers that transform the input data. These hidden layers consist of multiple nodes where the data is processed and transformed.

1) Multi-Layer Feed Forward Neural Network

Multi-Layer Feed Forward Neural Network (ML-FNN) is a type of neural network where the information flows in a forward direction only, from the input layer through the hidden layers to the output layer. Each neuron in a layer is connected to all neurons in the previous layer, and the output of each neuron is determined by a weighted sum of the inputs, followed by the application of an activation function as described in (Jiriwibhakorn, 2022). ML-FNN, on the other hand, can be with multiple hidden layers. Each layer is fully connected to the previous layer, and the weights of each connection are learned during the training process. ML-FNN are typically used for supervised learning tasks, such as classification or regression. ML-FNN have been successfully applied to a variety of tasks, including image recognition, speech recognition, and natural language processing.

In this study, the ML-FNN is one of the neural network architectures examined. The ML-FNN is designed to transmit data unidirectionally from one layer's node to every node in the following layer. The activation function, weight, and bias within each node perform the data transformation. The primary objective of implementing the ML-FNN is to train it to achieve optimal weights and biases, which can enhance its performance. To adjust the weights and biases of the training ML-FNN, the Levenberg-Marquardt backpropagation, an advanced version of the gradient descent method, is used. The learning error computation function can be MAE, RMSE, MAPE, or MSE, with MSE utilized in this study. The computation of MSE is shown in equation (2.10), and Figure 2.3 illustrates the workflow diagram of the ML-FNN and the hidden node.

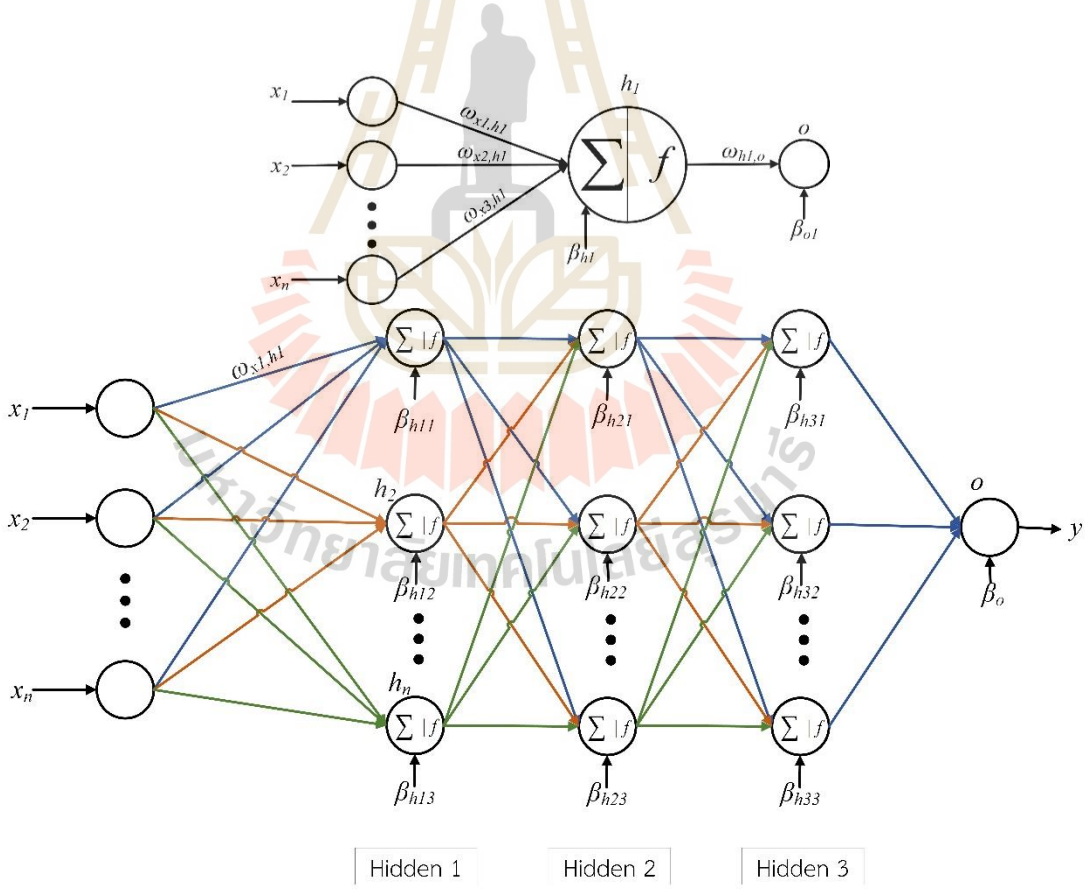


Figure 2.3 Structure of ML-FNN

$$E_j = \sum_{i=1}^{n} MSE\ (y^j_{forecast,i}, y^j_{actual,i}), \tag{2.10}$$

where E_j is training error at training iteration j^{th}, is actual output at sample i^{th}, x_i is input, n is the number of the training sample, and forecasting output at the training sample j^{th} can be calculated following equation 2.11

$$y_j = \sum_{j=1}^{n} \omega_{k,j} f(h_k) + \beta_j, \tag{2.11}$$

where $\omega_{k,j}$ is weights from hidden node k^{th} to output node at iteration j^{th}, β_j is the bias of output node at j^{th}. $f(h_k)$ is the outcome of hidden h_k. The fitness function is estimated using the training error (E). The fitness function can be calculated as follows:

$$Fitness(x) = Minimize\ E(x), \tag{2.12}$$

2) LSTM

In this study, the LSTM architecture is considered as a second option. Unlike the ML-FNN which processes samples individually, the LSTM processes a sequence of samples simultaneously. Each sample is independently processed by the RNN model, and the output of the previous sample is passed to every layer. The LSTM RNN can retain several previous outputs of nodes in all layers, including the output layer. To guide the preservation of information, three gates are included in the data transformation steps performed by the nodes: an input gate, an output gate, and a forget gate. These gates determine what information is retained, discarded, and provided as input to the next sample. The output of these gates is determined by an activation function, weight, and bias, much like the node operations. Similar to the

ML-FNN, the output of the forecast in the LSTM model can be described as a combination of the input data from the previous layer and data from previous samples. The LSTM block's architecture is depicted in Figure 2.4.

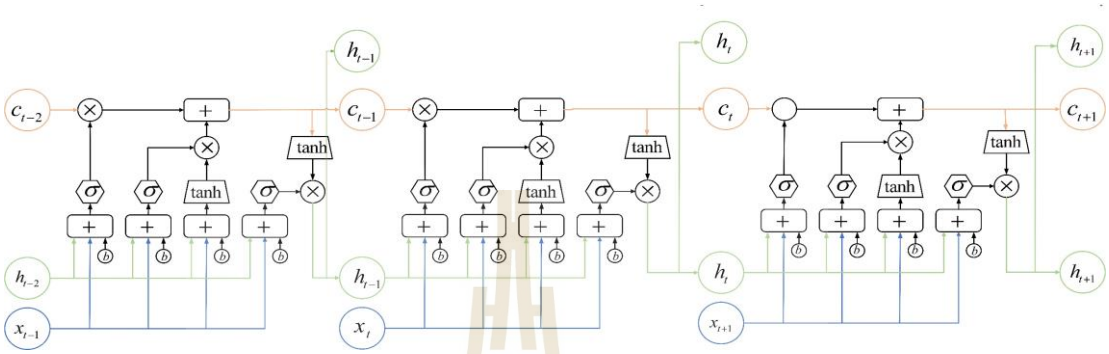


Fig. 2.4 The structure of the LSTM memory block

Given an input (x_t | t = 1, ..., T) with j_{th} frames, where x_t is the static feature of the T frame, the standard LSTM is used to learn a sequence of hidden states (h_t |t = 1, ..., T) to describe the dynamic of this input. The standard LSTM mainly consists of an input gate, forget gate, output gate, input modulation gate, and memory cell state, and one common LSTM unit at time step j can be repressed as follows:

$$f_t = \sigma(W_{hf} X_t + W_{hf} h_{t-1} + W_{cf} c_{t-1} + b_f), \quad (2.13)$$

$$i_t = \sigma(W_{hf} X_t + W_{hi} h_{t-1} + W_{ci} c_{t-1} + b_i), \quad (2.14)$$

$$c_t^k = f_t\, c_{t-1} + i_t\, \tanh(W_{xc} X_t + W_{hc} h_{t-1} + b_c), \quad (2.15)$$

$$o_t^k = \sigma(W_{xo} X_t + W_{ho} h_{t-1} + W_{co} c_{t-1} + b_o), \quad (2.16)$$

$$h_t^k = o_t\, \tanh(c_t), \quad (2.17)$$

$$s_t = g_t\, (i_t + c_{t-1}(f_t)), \quad (2.18)$$

where i_t , f_t , o_t , g_t , and c_t are the input gate, forget gate, output gate, input modulation gate, and memory cell state, respectively; σ is a sigmoid function. The weight matrices W_x and W_h, along with the bias vector b, are used in the LSTM

model. The input gate, represented as "i_t," regulates how much newly received input data at time step t should contribute to updating the memory cell. The forget gate, "f_t," controls how much the previous state (c_{t-1}) should be taken into account when deriving the current state (c_t). Finally, the output gate, "o_t," is responsible for determining how the LSTM unit's output at time step t should be generated based on the current state of the memory cell (c_t).

3) NARX

The Nonlinear Auto-Regressive Neural Network was a simpler version of NARX, which is an advanced implementation. The output regressor in the former was obtained using only one delayed feedback loop, whereas the latter uses m-tapped delay lines in both input and output signals n time. In the case of NARX, the parametric equation includes exogenous values. This information was reported by (Di Piazza, Di Piazza, & Vitale, 2016). Exogeneous values are included in NARX's parametric equation as follows.

$$y(t) = f\left[x(t-0),...,x(t-d);y(t-1),...,y(t-d)\right], \tag{2.19}$$

where d denotes the past value of output $y(t)$ and another series input $x(t)$ at sample t^{th}. The structure of NARX is shown in Figure 2.5

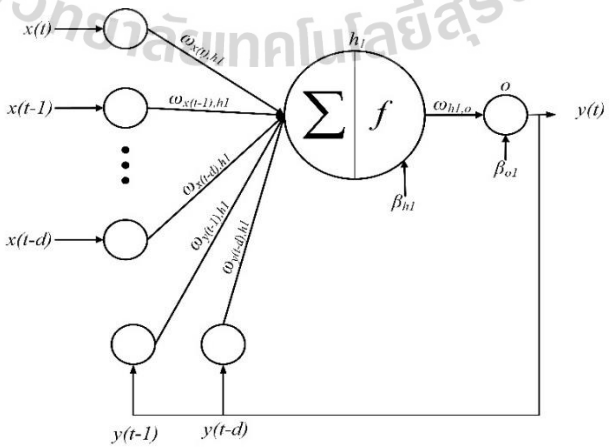
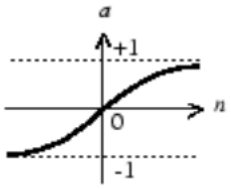
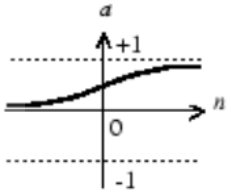
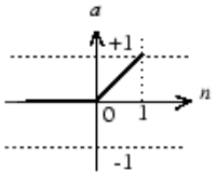


Fig. 2.5 NARX Architecture

Table 2.9 Activation function

No.	Activation functions	Formulation	Graph
1	Hyperbolic tangent sigmoid	$a = \tan sig(n)$ $a = \frac{2}{(1+e^{-2n})} - 1$ (2.20)	
2	Logarithmic sigmoid	$a = \log sig(n)$ $a = \frac{1}{(1+e^{-n})}$ (2.21)	
3	Positive linear	$a = poslin(n)$ $poslin(n) = \begin{cases} n & n >= 0 \\ 0 & n <= 0 \end{cases}$ (2.22)	

In machine learning and deep learning models, activation functions play a crucial role in the functioning of artificial neural networks. They allow each neuron to create a weighted sum of its inputs and transfer the resulting scalar value through a specified function. Various activation functions are used in ML/DL models, and some of the commonly used ones are described in Table 2.9, which provides a brief explanation of each function's characteristics and usage.

2.6.6 Benchmark model

Lastly, to offer context for the accuracy achieved by the prediction models provided, we evaluate one benchmark model.

1) Diurnal Persistence (DP)

Initially, we incorporate a Diurnal Persistence (DP) model in which the PV prediction matches the most recent available daily series data. Because PV production values are only accessible up to noon on day T, we examine the production values for T+1 as shown in Eq. 2.24

$$\hat{y}^{dp}(t) = y(t-48), \tag{2.24}$$

where time t varies between h = 0 and h = 24.

2.7 Validation methods

To evaluate the performance of forecasting models, it is important to use appropriate validation methods and variables. In this research, two widely used techniques in the machine learning field are employed: K-fold cross-validation and Holdout. K-fold cross-validation involves dividing the data into K equally sized subsets, training the model on K-1 subsets, and using the remaining subset for testing. This process is repeated K times, with each subset serving as the testing data exactly once. Holdout validation, on the other hand, involves randomly dividing the data into two sets: one for training and one for testing. The model is trained on the training set and then evaluated on the testing set. In addition to validation methods, various validation factors should be considered. (Bragança, Colonna, Oliveira, & Souto, 2022). identify several factors that can impact the performance of forecasting models, including the size and quality of the data set, the selection of input variables, the model architecture, the training algorithm, and the hyperparameters. These factors should be carefully considered and optimized to ensure that the forecasting model performs well and accurately predicts future outcomes.

2.7.1 Holdout (Training-test split)

The hold-out method is a simple way to divide the data into two separate subsets: the training set and the test set (Arlot & Celisse, 2010). This partitioning approach usually assigns 70 to 80 percent of the data for training and 30 to 20 percent for testing. It is beneficial since it requires less computational effort, but it can lead to a pessimistic estimator since the classifier is trained on only a portion of the data broader (Kohavi, 1995). The accuracy of the model depends on the choice of subjects for evaluation and the number of samples used for testing. If the data are re-

divided, the model's conclusions may change, and accuracy may be affected (Gholamiangonabadi, Kiselov, & Grolinger, 2020).

2.7.2 K fold cross validation

K-fold cross-validation (Kf-cv) is a statistical technique utilized in machine learning to evaluate the effectiveness of models. This technique is widely used to compare and choose a model for a specific predictive task, as it is easy to understand, implement, and yields less biased skill estimates compared to other approaches. This method involves dividing the dataset into k disjoint, roughly equal-sized folds, and then using each fold as a test set for a classification model produced from the remaining k-1 folds. The total performance is then computed as the mean of the k accuracies derived from k-CV (Wong, 2015). It should be noted that there is no universally superior cross-validation approach, and the method should be tailored to the specific context. However, this approach can be computationally expensive when k values are high, and the sample size is large (Arlot & Celisse, 2010).

2.7.3 Performance metric

Evaluating the accuracy of a forecast is essential for comparing its effectiveness with benchmark approaches, typically the naive method, and existing methods. However, there are numerous metrics available, and the appropriate ones should be selected based on the characteristics of the time series, such as the presence of zero values or the performance of the benchmark approach. This section aims to provide guidance on selecting the appropriate metrics for the point or probabilistic forecasts.

To assess the performance of a model, multiple performance indicators should be used, as each metric has unique characteristics. For example, the root-mean-square error (RMSE) heavily penalizes outliers due to the squared errors. In contrast, the mean absolute error (MAE) demonstrates the accuracy of a forecast relative to observations by calculating the average error between them using the following formula:

$$MAE = \frac{\sum_{i=1}^{n} | y_{actual,i} - y_{forecast,i} |}{n}, \quad (2.25)$$

where $y_{actual,i}$ and $y_{forecast,i}$ are the actual power output and forecast power output, respectively, at sample i^{th}, and n is the sample number. This indicator helps compare forecasts based on the same time series. However, since it is scale-dependent, it cannot be used for predictions of distinct time series owing to the intrinsic scale variations. In addition, a high number of relatively minor mistakes might mask a small number of substantial errors, which can be problematic if the prediction shows noise.

Mean square error (MSE) and root mean square error (RMSE) are defined as the following:

$$MSE = \frac{\sum_{i=1}^{n} \left(y_{actual,i} - y_{forecast,i} \right)^2}{n}, \quad (2.26)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(y_{actual,i} - y_{forecast,i} \right)^2}{n}}, \quad (2.27)$$

Similarly, their application is constrained by size dependence. In addition, the squared error makes these measures more susceptible to outliers than the MAE. Nonetheless, these measures are extensively used owing to their theoretical importance in statistical research and because they give immediate insight into the error variance and standard deviation, respectively.

As noted earlier, the metrics presented in equations (2.26) to (2.27) are inadequate for evaluating forecast accuracy across various time series and are insufficient without prior knowledge of the studied PV power plant. Percentage error measurements can facilitate comparisons of forecasts across different temporal and geographic dimensions. Several denominators can be used to normalize the inaccuracy of PV power forecasting. It was determined subjectively that MAE normalized by

average output would be preferred, but normalizing by capacity would be more acceptable if both MAE and RMSE were to be used. A similar principle may apply to load forecasting, although no supporting literature was found. Normalized by capacity, the mean absolute percentage error (MAPE) and normalized root mean square error (NRMSE) can be calculated as follows:

$$MAPE = \sum_{i=1}^{n} \frac{\left| y_{actual,i} - y_{forecast,i} \right|}{y_{actual,i}} \times 100\% , \tag{2.28}$$

$$NRMSE = \sqrt{\sum_{i=1}^{n} \left(\frac{\left| y_{actual,i} - y_{forecast,i} \right|}{y_{actual,i}} \right)^2} \times 100\% , \tag{2.29}$$

The MAPE measure is commonly used in forecasting due to its simplicity and widespread recognition. On the other hand, NRMSE, like RMSE, is more sensitive to outliers than MAPE. Equations (2.26) and (2.27) are sometimes normalized by the rated power rather than the measured value, which has the advantage of not having an absolute zero.

In order to assess forecasting biases, such as overestimation or underestimation, the mean bias error (MBE) is commonly used. This metric provides an immediate indication of the average bias in a model. A large and positive MBE indicates a significant overestimate, while a large and negative MBE indicates a significant underestimate. However, it should be noted that MBE is dependent on the scale of the data and does not provide information about the error distribution. Despite these drawbacks, MBE is still considered useful as it can be reduced or eliminated through post-processing or considered directly by the utility. The MBE can be expressed as follows:

$$MBE = \frac{1}{n} \sum_{i=1}^{n} (y_{actual,i} - y_{forecast,i}) , \tag{2.30}$$

The coefficient of determination, R^2, is a statistical measure that indicates the degree to which a statistical model fits the data and shows the extent to which the variance of the errors and the variance of the observed values correspond. R^2 is given by:

$$R^2 = 1 - \frac{\sigma(y_{actual} - y_{forecast})}{\sigma(y_{forecast})}, \tag{2.31}$$

In the context of short-term forecasting, the methods described in the previous section can be applied, but additional metrics have also been developed for probabilistic forecasting. However, the lack of well-established assessment methodologies is one of the primary reasons for the immaturity of probabilistic irradiance forecasting (PIF). Perfect reliability of a probabilistic prediction occurs when the probability derived from the quantiles of the forecast model and the actual probability is the same. Any deviation from this reduces the forecast's reliability, and it is linked to the forecast's bias, where high predictability corresponds to low bias. The dependability of the model can be determined by creating a time series that tracks instances of over- or under-prediction. If this series is near the diagonal, the model's dependability is considered good. Another way to assess dependability is to examine the histograms of the probability integral transformation (PIT). If the probabilistic prediction is accurate, the PIT histograms are uniform by definition. The purpose of a probabilistic prediction is to ensure that the probability distribution of data falls within the prediction interval. The prediction interval coverage probability (PICP) is computed to determine whether this is true, which can be expressed as follows:

$$PICP = \frac{1}{n}\sum_{i=1}^{n} c_i, \tag{2.32}$$

where c_i is defined as:

$$c_i = \begin{cases} 1, \ if \ x_i \in [L_i, U_i] \\ 0, \ if \ x_i \notin [L_i, U_i] \end{cases}, \tag{2.33}$$

where L_i and U_i indicate the lower and upper limits, respectively, of the prediction interval. From the definition of C_i, we may derive that a high value for *PICP* indicates that a larger proportion of findings fall inside the prediction interval, which is desirable. The *PICP* measurement is a quantitative statement of dependability and should be more than the actual confidence level since they are invalid and should be disregarded if they are lower.

Nonetheless, if one considers the performance of the forecast based purely on the PICP, it is feasible to choose a wide range between lower bound Li and upper bound Ui, so that the scope probability is artificially increased while the deviation of the forecast is unacceptable and decision makers are provided with little useful information. The informativeness of prediction intervals is in reality governed by their breadth. Therefore, the PICP should be simultaneously examined with the prediction interval normalized average width (PINAW), a metric that quantitatively evaluates the width of the prediction intervals. Following is a description of the PINAW:

$$PINAW = \frac{1}{nR} \sum_{i=1}^{n} (U_i - L_i), \tag{2.34}$$

where R is used to standardize the average width of the prediction interval and reflects the highest forecast value minus the lowest forecast value.

2.8 Conclusion

This chapter serves as an introduction to the application of PV forecasting. Within this context, the use of machine learning (ML) and deep learning (DL) has been widely discussed within the realm of electrical research. Specifically, recent research in PV power forecasting has been reviewed. Following this discussion, the adopted method has been presented, which incorporates forecasting methods alongside a

performance matrix for evaluation purposes. The accurate forecasting of PV power is a crucial topic within the renewable energy industry, as it aids in the integration of PV power into the electrical grid and enhances the operation of PV power plants. ML and DL techniques have become popular in the context of PV power forecasting due to their ability to derive predictions from complex patterns within historical data. Incorporating methods such as time series analysis, statistical modeling, and ML/DL models, the forecasting techniques utilized in PV power forecasting are further evaluated using a performance matrix, which measures the effectiveness of the forecast by utilizing metrics such as mean absolute error, root means squared error and correlation coefficient.

In conclusion, this chapter provides an overview of the application of ML/DL in PV power forecasting and highlights recent advancements in this field.

2.9 References

Ahmad, M., Reynolds, J., & Rezgui, Y. (2018). Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees. *Journal of Cleaner Production,* 203, 810-821. doi:10.1016/j.jclepro.2018.08.207

Akhter, M. N., Mekhilef, S., Mokhlis, H., & Mohamed Shah, N. (2019). Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques. *IET Renewable Power Generation,* 13(7), 1009-1023. doi:https://doi.org/10.1049/iet-rpg.2018.5649

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys,* 4, 40-79.

Beltran, H., Cardo-Miota, J., Segarra-Tamarit, J., & Pérez, E. (2021). Battery size determination for photovoltaic capacity firming using deep learning irradiance forecasts. *Journal of Energy Storage,* 33, 102036. doi:https://doi.org/10.1016/j.est.2020.102036

Bragança, H., Colonna, J. G., Oliveira, H. A. B. F., & Souto, E. (2022). How Validation Methodology Influences Human Activity Recognition Mobile Systems. *Sensors,* 22(6). doi:10.3390/s22062360

Carrera, B., & Kim, K. (2020). Comparison Analysis of Machine Learning Techniques for Photovoltaic Prediction Using Weather Sensor Data. *Sensors,* 20(11). doi:10.3390/s20113129

Carriere, T., Vernay, C., Pitaval, S., & Kariniotakis, G. (2020). A Novel Approach for Seamless Probabilistic Photovoltaic Power Forecasting Covering Multiple Time Frames. *IEEE Transactions on Smart Grid,* 11(3), 2281-2292. doi:10.1109/TSG.2019.2951288

Chang, X., Li, W., & Zomaya, A. Y. (2020). A Lightweight Short-Term Photovoltaic Power Prediction for Edge Computing. *IEEE Transactions on Green Communications and Networking,* 4(4), 946-955. doi:10.1109/TGCN.2020.2996234

Das, U. K., Tey, K. S., Seyedmahmoudian, M., Mekhilef, S., Idris, M. Y. I., Van Deventer, W., . . . Stojcevski, A. (2018). Forecasting of photovoltaic power generation and model optimization: A review. *Renewable and Sustainable Energy Reviews,* 81, 912-928. doi:https://doi.org/10.1016/j.rser.2017.08.017

Di Piazza, A., Di Piazza, M. C., & Vitale, G. (2016). Solar and wind forecasting by NARX neural networks. *Renewable Energy and Environmental Sustainability,* 1, 39. doi:10.1051/rees/2016047

du Plessis, A. A., Strauss, J. M., & Rix, A. J. (2021). Short-term solar power forecasting: Investigating the ability of deep learning models to capture low-level utility-scale Photovoltaic system behaviour. *Applied Energy,* 285, 116395. doi:https://doi.org/10.1016/j.apenergy.2020.116395

Fekri, M. N., Ghosh, A. M., & Grolinger, K. (2020). Generating Energy Data for Machine Learning with Recurrent Generative Adversarial Networks. *Energies,* 13(1). doi:10.3390/en13010130

Fen, L., Chunyang, L., Yong, Y., Quanquan, Y., Jinbin, Z., & Lijuan, W. (2017). Short-term photovoltaic power probability forecasting based on OLPP-GPR and modified clearness index. *The Journal of Engineering,* 2017(13), 1625-1628. doi:https://doi.org/10.1049/joe.2017.0607

Gholamiangonabadi, D., Kiselov, N., & Grolinger, K. (2020). Deep Neural Networks for Human Activity Recognition With Wearable Sensors: Leave-One-Subject-Out Cross-Validation for Model Selection. *IEEE Access,* 8, 133982-133994. doi:10.1109/ACCESS.2020.3010715

González Ordiano, J. Á., Gröll, L., Mikut, R., & Hagenmeyer, V. (2020). Probabilistic energy forecasting using the nearest neighbors quantile filter and quantile regression. *International Journal of Forecasting,* 36(2), 310-323. doi:https://doi.org/10.1016/j.ijforecast.2019.06.003

Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., & Hyndman, R. J. (2016). Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond. *International Journal of Forecasting,* 32(3), 896-913. doi:https://doi.org/10.1016/j.ijforecast.2016.02.001

Hossain, M. S., & Mahmood, H. (2020). Short-Term Photovoltaic Power Forecasting Using an LSTM Neural Network and Synthetic Weather Forecast. *IEEE Access,* 8, 172524-172533. doi:10.1109/ACCESS.2020.3024901

Huang, Q., & Wei, S. (2020). Improved quantile convolutional neural network with two-stage training for daily-ahead probabilistic forecasting of photovoltaic power. *Energy Conversion and Management,* 220, 113085. doi:https://doi.org/10.1016/j.enconman.2020.113085

Hyndman, R. J. (2020). A brief history of forecasting competitions. *International Journal of Forecasting,* 36(1), 7-14. doi:https://doi.org/10.1016/j.ijforecast.2019.03.015

Jiriwibhakorn, S. (2022). *Forecasting Machines in Power Systems (ISBN:* 978-616-593-370-4). Retrieved from www.se-ed.com

Keerthisinghe, C., Mickelson, E., Kirschen, D. S., Shih, N., & Gibson, S. (2020). Improved PV Forecasts for Capacity Firming. *IEEE Access,* 8, 152173-152182. doi:10.1109/ACCESS.2020.3016956

Kingma, D. P., & Ba, J. (2015). *Adam: A Method for Stochastic Optimization*. http://arxiv.org/abs/1412.6980

Kohavi, R. (1995). *A study of cross-validation and bootstrap for accuracy estimation and model selection.* Paper presented at the Ijcai.

Lateko, A., Yang, H.-T., & Huang, C.-M. (2022). Short-Term PV Power Forecasting Using a Regression-Based Ensemble Method. *Energies,* 15, 4171. doi:10.3390/en15114171

Leva, S., Dolara, A., Grimaccia, F., Mussetta, M., & Ogliari, E. (2017). Analysis and validation of 24 hours ahead neural network forecasting of photovoltaic output power. *Mathematics and Computers in Simulation,* 131, 88-100. doi:https://doi.org/10.1016/j.matcom.2015.05.010

Li, P., Zhou, K., Lu, X., & Yang, S. (2020). A hybrid deep learning model for short-term PV power forecasting. *Applied Energy,* 259, 114216. doi:https://doi.org/10.1016/j.apenergy.2019.114216

Liu, L., Zhao, Y., Wang, Y., Sun, Q., & Wennersten, R. (2019). A Weight-Varying Ensemble Method for Short-term Forecasting PV Power Output. *Energy Procedia*.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting,* 34(4), 802-808. doi:https://doi.org/10.1016/j.ijforecast.2018.06.001

Mellit, A., Massi Pavan, A., Ogliari, E., Leva, S., & Lughi, V. (2020). Advanced Methods for Photovoltaic Output Power Forecasting: A Review. *Applied Sciences,* 10(2). doi:10.3390/app10020487

Najibi, F., Apostolopoulou, D., & Alonso, E. (2021). Enhanced performance Gaussian process regression for probabilistic short-term solar output forecast. *International Journal of Electrical Power & Energy Systems,* 130, 106916. doi:https://doi.org/10.1016/j.ijepes.2021.106916

Niu, D., Wang, K., Sun, L., Wu, J., & Xu, X. (2020). Short-term photovoltaic power generation forecasting based on random forest feature selection and CEEMD: A case study. *Applied Soft Computing,* 93, 106389. doi:https://doi.org/10.1016/j.asoc.2020.106389

Nkuriyingoma, O., & Selcuklu, S. (2021). Solar power plant generation forecasting using NARX neural network model: A case study. *International Journal of Energy Applications and Technologies,* 8, 80-92. doi:10.31593/ijeat.870088

Pan, C., & Tan, J. (2019). Day-Ahead Hourly Forecasting of Solar Generation Based on Cluster Analysis and Ensemble Model. *IEEE Access,* 7, 112921-112930. doi:10.1109/ACCESS.2019.2935273

Rajagukguk, R. A., Ramadhan, R. A. A., & Lee, H.-J. (2020). A Review on Deep Learning Models for Forecasting Time Series Data of Solar Irradiance and Photovoltaic Power. *Energies,* 13(24). doi:10.3390/en13246623

Seifi, H., & Sepasian, M. (2011). *Electric Power System Planning: Issues, Algorithms and Solutions* (Vol. 49).

Severiano, C. A., Silva, P. C. L., Weiss Cohen, M., & Guimarães, F. G. (2021). Evolving fuzzy time series for spatio-temporal forecasting in renewable energy systems. *Renewable Energy*.

Soman, S., Zareipour, H., Malik, O. P., & Mandal, P. (2010). *A review of wind power and wind speed forecasting methods with different time horizons*.

Visser, L., AlSkaif, T., & van Sark, W. (2022). Operational day-ahead solar power forecasting for aggregated PV systems with a varying spatial distribution. *Renewable Energy,* 183, 267-282. doi:https://doi.org/10.1016/j.renene.2021.10.102

Wang, F., Xuan, Z., Zhen, Z., Li, K., Wang, T., & Shi, M. (2020). A day-ahead PV power forecasting method based on LSTM-RNN model and time correlation modification under partial daily pattern prediction framework. *Energy Conversion and Management,* 212, 112766. doi:https://doi.org/10.1016/j.enconman.2020.112766

Wang, H., Liu, Y., Zhou, B., Li, C., Cao, G., Voropai, N., & Barakhtenko, E. (2020). Taxonomy research of artificial intelligence for deterministic solar power forecasting. *Energy Conversion and Management,* 214, 112909. doi:https://doi.org/10.1016/j.enconman.2020.112909

Wang, X., Guo, P., & Huang, X. (2011). A Review of Wind Power Forecasting Models. *Energy Procedia,* 12, 770-778. doi:https://doi.org/10.1016/j.egypro.2011.10.103

CHAPTER 3
COMPARATIVE STUDY OF DATA-DRIVEN-BASED SHORT-TERM PHOTOVOLTAIC POWER GENERATION FORECASTING MODELS: SELECTION OF HYPERPARAMETER AND VALIDATION METHODS

3.1 Background

In the context of designing a distribution grid that incorporates a PV system, accurate forecasting is crucial to optimize efficiency. To address this issue, various techniques have been proposed, but the selection of an appropriate method requires a thorough understanding and knowledge of the available options. This chapter presents a comparative study that examines the efficacy and benefits of supervised photovoltaic power forecasting methods that can be applied appropriately to various systems.

The study will test predictive models from chapter 2 using holdout validation and k-fold cross-validation. Performance metrics will be used to evaluate each method. A MATLAB program will simulate the study and the results, strengths, and limitations of each method will be discussed. The objective is to comprehensively analyze available forecasting techniques for photovoltaic power generation, identify their strengths and weaknesses, and aid in selecting an appropriate technique. The study contributes valuable insights to the field and can inform the development of efficient forecasting schemes for photovoltaic systems in distribution grids. It includes six cases to assess factors impacting forecasting model performance. The cases investigated are as follows: 1) the impact of hyperparameter tuning, 2) the impact of activation function selection, 3) the impact of normalization techniques, 4) the impact of seasonal and test set selection, 5) the impact of validation methods, and 6) the impact of incomplete datasets.

3.2 Comparative study workflow

The four steps consist of importing data, visualizing and preprocessing data, predictive modeling, deploying the forecasting model, and evaluating the model as shown in Figure 3.1.

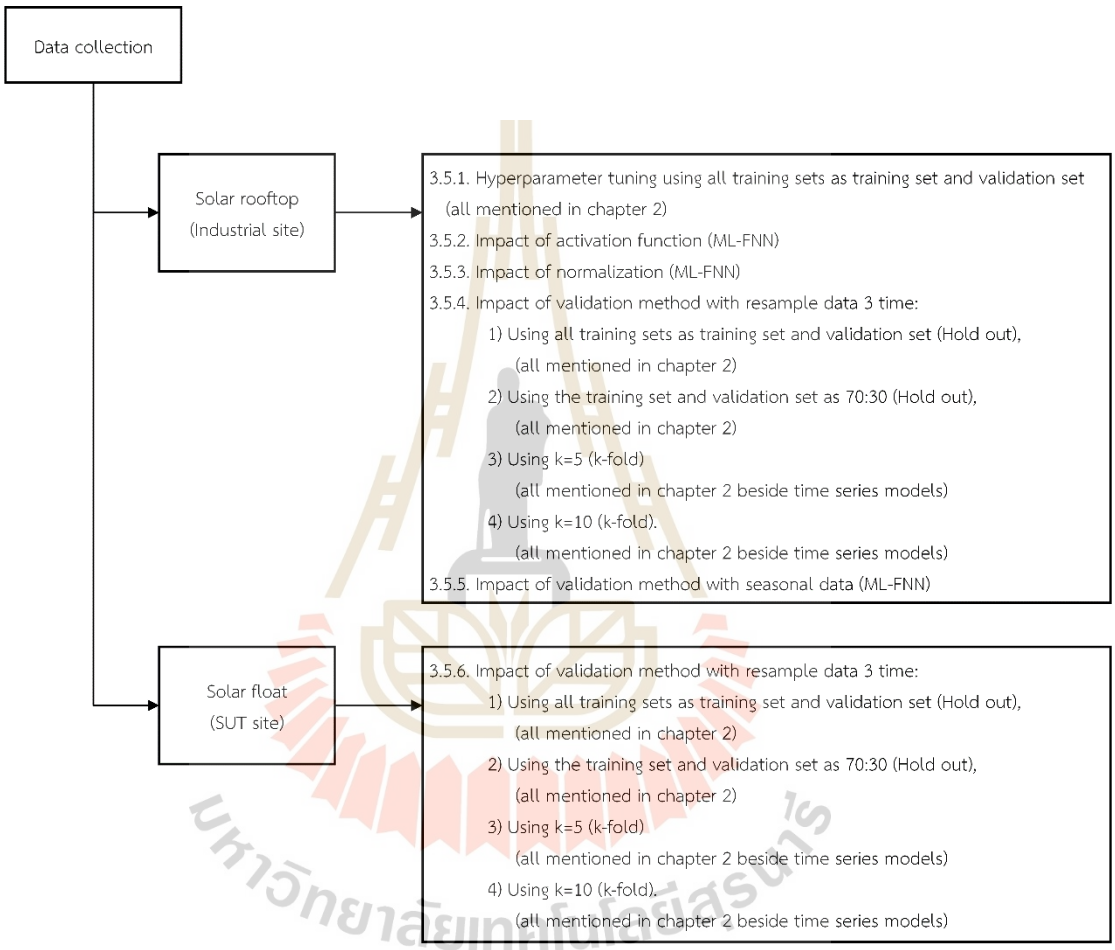


Figure 3.1 Case of comparative study

3.3 Imported dataset

The are two datasets that were used in this study: (1) Solar rooftop dataset (Industry) and (2) Solar floating plant dataset (at Suranaree university of technology).

The first dataset contains one year's worth of historical data from a 14 MWp rooftop solar power plant in Nakhon Ratchasima province, Thailand. The data includes

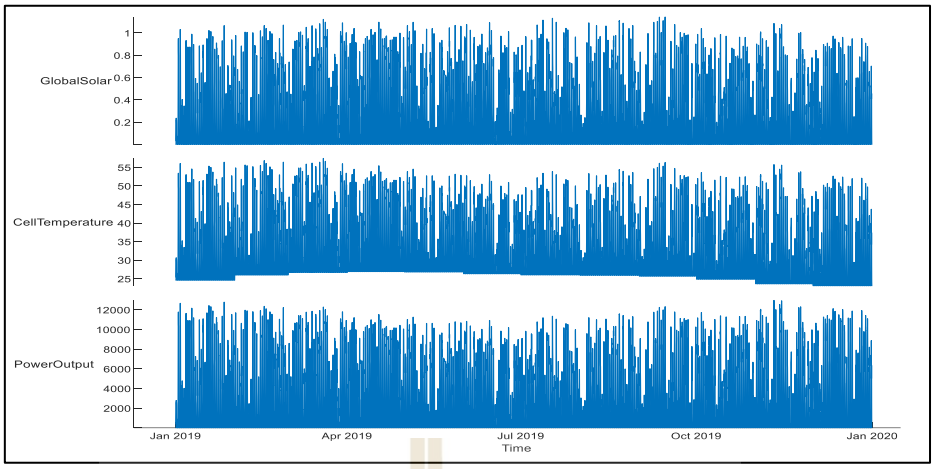
time, irradiance, temperature, and power output (Junhuathon & Chayakulkheeree, 2021; Mongkol Treekijjanon, 2020). The second dataset contains eight months of historical data from a 1.5 MWp solar floating plant at Suranaree University of Technology (SUT), also located in Nakhon Ratchasima province. The data for this plant includes time, irradiance, ambient temperature, wind speed, and power output. Table 3.1 provides a summary of the datasets, including their parameters, sample sizes, and resolutions.

Table 3.1 Dataset description

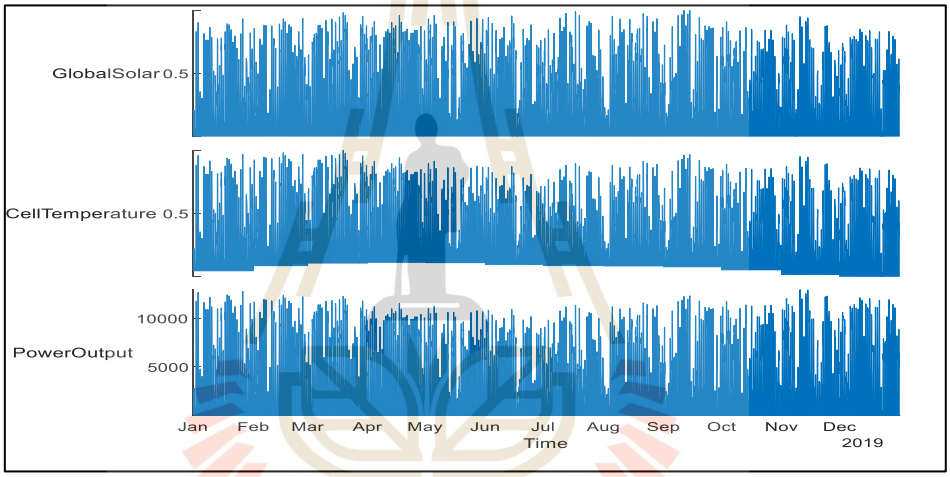
Detail	Industrial site	SUT site
Parameters	Time, Ambient temperature, irradiance	Time, Ambient temperature, irradiance, wind speed
Sample	8,760	6,651
Resolutions	hourly	hourly

3.4 Data preprocessing and visualization

For the industrial site, the dataset was standardized into 0 to 1 for every variable. However, this dataset is unnecessary to remove outliers, missing data, and NAN data because the dataset is completely preprocessed before. The dataset before standardization and after standardization will be shown in Figure 3.2 (a) and (b), respectively.



(a)



(b)

Figure 3.2 Industrial site dataset visualization (a) before normalization (b) after normalization

As shown in figure 3.2 (a), this dataset consists of solar irradiance (kW), temperature (^{o}C), and power generation (kW) from the PV system. The dataset is rather complete.

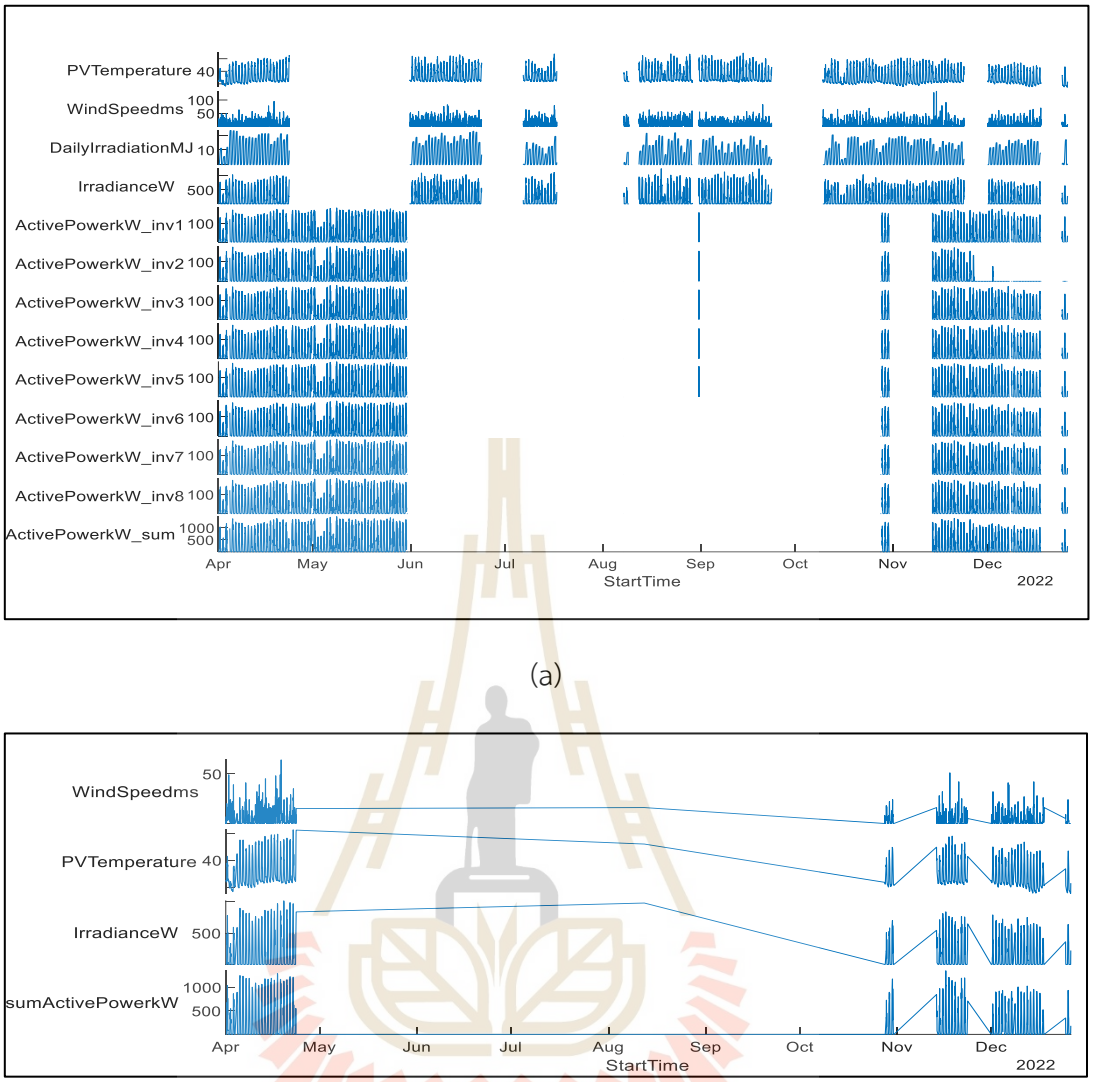


Figure 3.3 SUT site dataset visualization (a) original dataset (b) preprocessed dataset

As shown in Figure 3.3, outlier, missing data, and not a number (NaN) data need to be managed because this dataset has a problem of data collecting process for a huge period. The dataset before standardization and after standardization will be shown in Figure 3.3 (a) and (b), respectively. From Figure 3.3 (a), temperature (^{o}C), wind speed (m/s), and solar irradiance (W) were collected from a local measurement station and power generation was collected from 8 inverters (Inverter rated power:175

kW). There are different huge gaps among collecting devices. Then, the dataset was preprocessed and selected the intersect period as shown in Figure 3.3 (b).

3.5 Case studies

The cases investigated are as follows: 1) the impact of hyperparameter tuning, 2) the impact of activation function selection, 3) the impact of normalization techniques, 4) the impact of seasonal and test set selection, 5) the impact of validation methods, and 6) the impact of incomplete datasets.

3.5.1 Impact of hyperparameter tuning

To maximize the accuracy of the data-driven-based forecasting method, the widely implemented forecasting methods were used with industrial dataset, and each dataset. Before splitting the dataset into the training set and validation set, the test set had been chosen randomly to evaluate the performance of forecasting models.

To achieve the best condition in data-driven-based forecasting models, besides using the appropriate dataset to train and validate, the hyperparameter of models should be adjusted to the appropriate value. The hyperparameter tuning results are shown in Table 3.2 and more details of adjustment can be seen in appendix A.

Table 3.2 Hyperparameter tuning from data set in case 1

No.	Methods	Hyperparameters
1	LR: Linear	Preset: Linear Terms: Linear Robust option: Off

Table 3.2 Hyperparameter tuning from data set in case 1 (continue)

No.	Methods	Hyperparameters
2	LR: Interactions linear	Preset: Interactions Linear Terms: Interactions Robust option: Off
3	LR: Robust linear	Preset: Robust Linear Terms: Linear Robust option: On
4	LR: Stepwise linear regression	Preset: Stepwise Linear Initial terms: Linear Upper bound on terms: Interactions Maximum number of steps:1000
5	Optimized Ensemble of Trees	Ensemble method: Bag Minimum leaf size: 39 Number of learners: 24 Number of predictors to sample: 3
6	Optimized SVR	Kernel function: Linear Box constraint: 105.0635 Epsilon: 238.7449 Standardize data: true
7	Optimized GPR	Basis function: Linear Kernel function: Isotropic Exponential Kernel scale: 0.054297 Sigma: 0.0010932 Standardize: false

Table 3.2 Hyperparameter tuning from data set in case 1 (continue)

No.	Methods	Hyperparameters
8	ML-FNN (Manual, See more details in Appendix A)	Number of fully connected layers: 3 First layer size: 20, Second layer size: 20 Third layer size: 20 Activation function: Poslin
9	NARX (Manual, See more details in Appendix A)	Number of fully connected layers: 3 First layer size: 20, Second layer size: 20 Third layer size: 20 Activation function: Poslin
10	LSTM (Manual, See more details in Appendix A)	Number of fully connected layers: 3 First layer size: 20, Second layer size: 20 Third layer size: 20 Solver: adam Initial Learn Rate: 0.03 Learn Rate Schedule: piecewise Learn Rate Drop Factor: 0.7 Learn Rate Drop Period: 100 Max Epochs: 300 Mini Batch Size: 24
11	Benchmark model: DP	-

Note: Hyperparameters of Ensemble of trees, SVR, and GPR, were optimized by Bayesian optimization.

3.5.2 Impact of activation function

Activation functions are a critical component in machine learning and deep learning models, as they introduce nonlinearity into the model, allowing it to capture complex patterns and relationships within data. The key impacts of activation functions in these models are: firstly, activation functions introduce nonlinearity into the model, enabling it to learn and capture complex nonlinear relationships in the

data, without which a neural network would only be a linear regression model. Secondly, activation functions impact gradient propagation in the model, as different activation functions possess varying gradients, which affect the error signals propagating back through the network during training. Activation functions such as the sigmoid and hyperbolic tangent functions are vulnerable to the vanishing gradient problem, where the gradients become very small, making it difficult for the network to learn. Thirdly, certain activation functions, such as the rectified linear unit and its variants, can induce sparsity into the model by setting some of the activations to zero, which reduces the number of parameters in the model and improves its efficiency. Lastly, activation functions also affect the output range of the model, with the sigmoid function outputting values between 0 and 1, while the hyperbolic tangent function outputs values between -1 and 1. The performance of the model may be impacted based on the problem's nature being solved.

In summary, the choice of activation function can have a significant impact on the performance of a machine learning or deep learning model. It is important to choose an appropriate activation function for the problem at hand, taking into account its nonlinearity, gradient propagation, sparsity, and output range. Therefore, this thesis aim to compare between commonly used activation function as shown in Table 3.3.

Table 3.3 Comparison of activation function with normalization

No.	Layer 1	Layer 2	Layer 3	Norm	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1	logsig	logsig	logsig	yes	8.916	309.866	157.404	253.208
2	poslin	poslin	poslin	yes	8.504	314.900	159.798	182.084
3	tansig	tansig	tansig	yes	9.431	358.750	172.391	192.778

As show in table 3.3 show that the different activation function slightly affect to forecasting accuracy and the the best performance actibation function for this study with normalization (minmax 0 to 1) is positive linear.

3.5.3 Impact of normalization

Normalization is a process of transforming the input into a more suitable format for machine learning and deep learning models. It can have a significant impact on the performance of the model. Here are some of the impacts of normalization. 1) Normalization can help the model to converge faster during training by preventing it from getting stuck in a local minimum. 2) Normalization can reduce overfitting by preventing the model from becoming too sensitive to the scale of the input features. This can improve the generalization performance of the model on unseen data. 3) Normalization can help the model to extract more meaningful features from the input data by removing correlations between input features. And 4) normalization can increase the efficiency of the model during training by reducing the number of iterations required for convergence.

There are several normalization techniques, such as min-max normalization, z-score normalization, and log normalization, each with its own advantages and disadvantages. The choice of normalization technique depends on the nature of the data and the specific requirements of the model. In summary, normalization is an important technique in machine learning and deep learning models that can help improve convergence, reduce overfitting, improve feature extraction, and increase efficiency during training. Therefore, this thesis aim to compare between commonly used normalization (Faruque et al., 2022) as shown in Table 3.4.

Table 3.4 Comparison of normalization for ML-FNN

No.	Layer 1	Layer 2	Layer 3	Norm	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1	poslin	poslin	poslin	minmax	8.504	314.900	159.798	179.017
2	poslin	poslin	poslin	minmax	8.504	314.900	159.798	182.084
3	poslin	poslin	poslin	zero mean	36.212	314.900	159.798	177.318
4	logsig	logsig	logsig	zero mean	47.613	309.867	157.403	193.467
5	tansig	tansig	tansig	zero mean	47.767	358.770	172.409	486.802

3.5.4 Impact of seasonal selection and test set selection

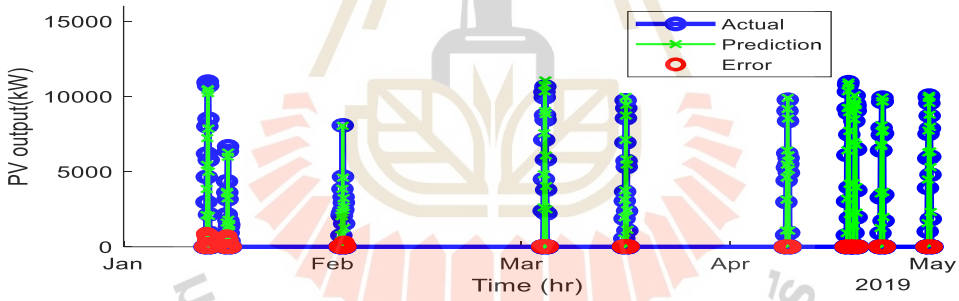
As mentioned, seasonal selection and test set selection can both have significant impacts on the outcomes of experiments and analyses. Seasonal selection refers to the bias that can be introduced when data is collected or analyzed during specific times of the year. For example, if a study is conducted only in the summer months, it may not be representative of the entire year and may lead to incorrect conclusions. To address this issue, it is important to collect data across different seasons or to use statistical methods to account for seasonal effects. Test set selection is the process of choosing a subset of data to evaluate the performance of a machine learning model. The choice of test set can impact the results of the evaluation, and it is important to choose a test set that is representative of the data that the model will encounter in the real world. If the test set is not representative, the evaluation may be overly optimistic or pessimistic, leading to incorrect conclusions about the model's performance.

In summary, both seasonal selection and test set selection can have significant impacts on the outcomes of experiments and analyses. It is important to be

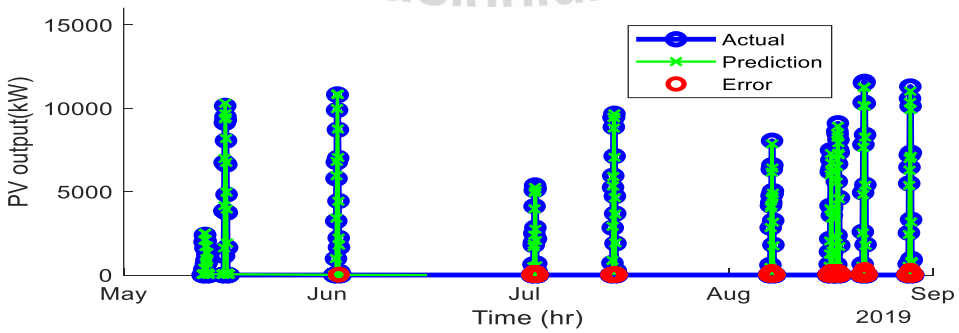
aware of these issues and to take appropriate steps to mitigate their effects. Therefore, the seasonal selection and test set selection need to be investigated. The results are shown in Table 3.5 and Figure 3.4

Table 3.5 Performance of each seasonal model

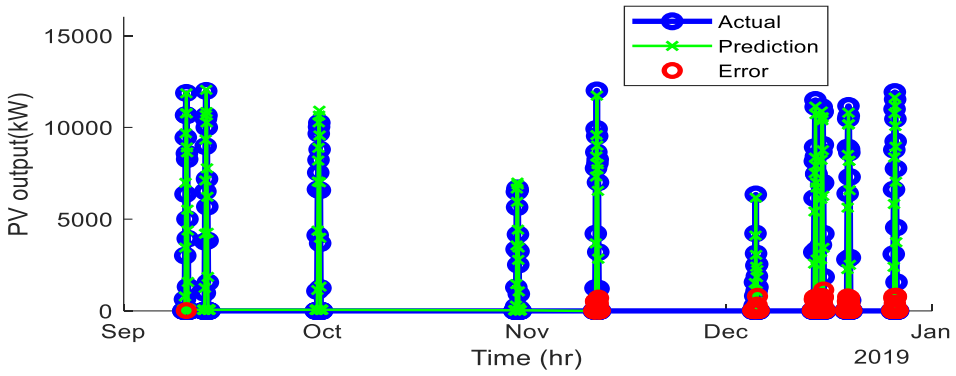
Season	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1 Winter (Jan - April)	6.727	213.355	118.709	30.530
2 Summer (May - August)	9.308	110.297	80.663	5.056
3 Rainy (October - December)	10.486	319.570	209.986	40.054
Average	8.840	214.407	136.453	-



(a)



(b)



(c)

Figure 3.4 forecasting of each model: (a) winter forecasting, (b) summer forecasting, and (c) rainy forecasting

As shown in Figure 3.4 and Table 3.5, the performance of seasonal models depends on various factors, such as the length of the historical data, the selection of appropriate seasonal periods, and the choice of forecasting model. Overall, seasonal models are an effective tool for PV power forecasting, and their accuracy can be further improved by incorporating weather and other relevant data into the forecasting model.

3.5.5 Impact of validation method

To evaluate the performance of forecasting models solar rooftop at Industrial site was used. In order to assess the performance of forecasting models for the solar rooftop site, four case studies were conducted. The first two cases evaluated the performance of all models described in chapter 2. However, for the third and fourth cases, the LSTM and NARX models were excluded due to their significant time requirements. These models would require more time for k-fold cross-validation, and therefore, were not considered for these cases. The best forecasting results of the industrial site are shown in Table 3.6-3.10 and Figure 3.5-3.12, respectively.

1) Case 1: using all training set as training set and validation set

Table 3.6 The forecasting results of case 1

No.	Model	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1	LR: Linear	13.289	383.902	260.725	2.552
2	LR: Interactions linear	13.564	369.345	262.075	0.899
3	LR: Robust linear	9.747	377.425	202.106	1.072
4	LR: Stepwise linear regression	13.564	369.345	262.075	1.127
5	Optimized Ensemble of trees	11.687	395.768	194.175	147.850
6	Optimized SVR	10.995	345.295	203.743	372.390
7	Optimized GPR	16.085	364.564	203.489	2,163.200
8	ML-FNN	8.504	314.900	159.797	181.924
9	NARX	55.437	1,509.200	808.321	36,200.000
10	LSTM	12.167	369.067	204.658	3,873.700
11	Benchmark model: DP	61.310	2,114.300	1,073.100	0.013

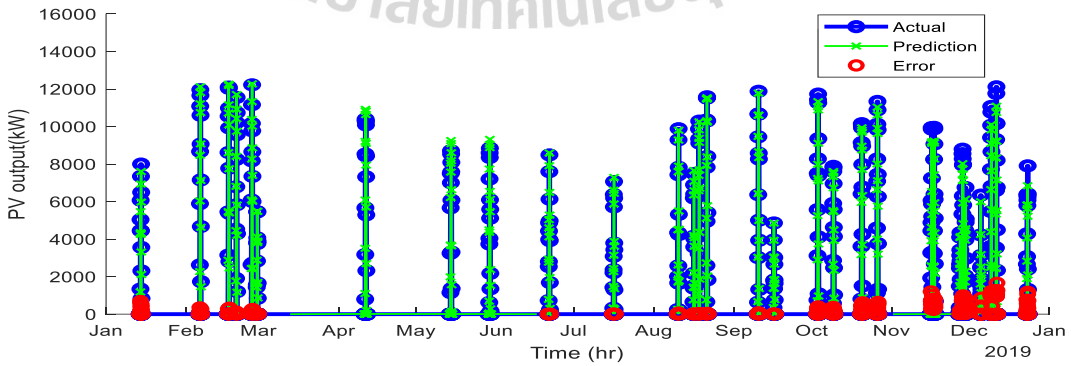


Figure 3.5 The best forecasting results of case 1 (ML-FNN)

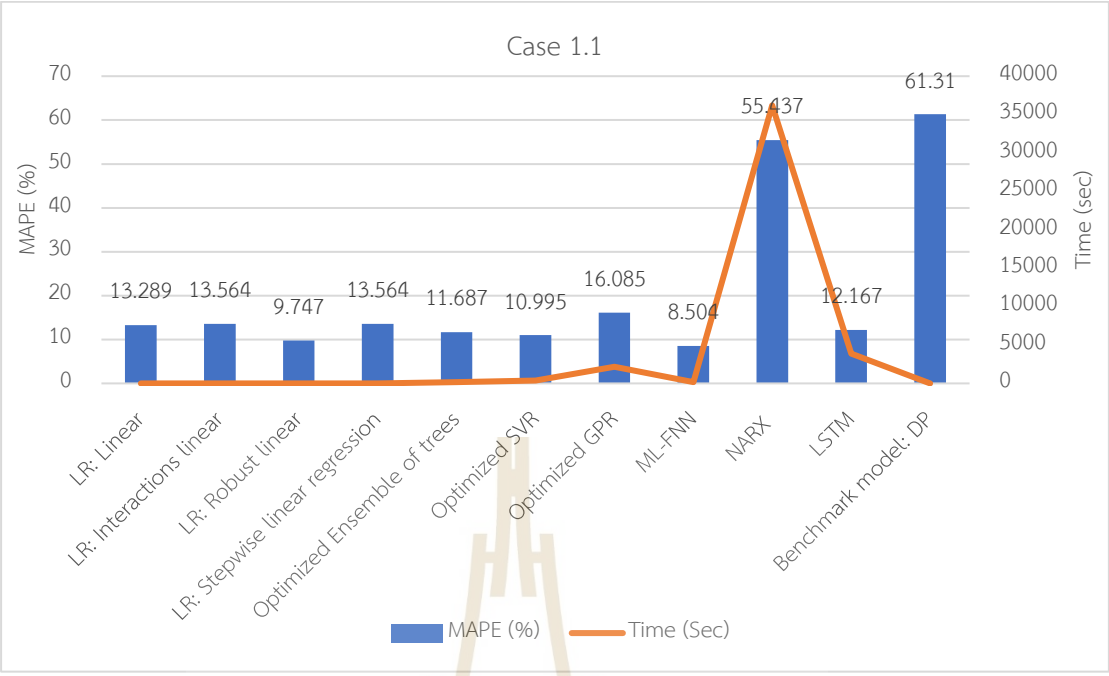


Figure 3.6 summary of case 1

Figure 3.6 reveals that the most accurate model for photovoltaic (PV) forecasting, concerning the use of all training sets as validation sets, is the multi-layer feedforward neural network (ML-FNN). This model exhibits a mean absolute percentage error (MAPE) of 8.504% and a training time of 181.924 seconds. These findings demonstrate that the ML-FNN model is highly effective in predicting short-term hourly power output from PV systems, thus contributing to the development of efficient and effective forecasting schemes for distribution grids that incorporate PV systems.

2) Case 2: using training set and validation set as 70:30

Table 3.7 The forecasting results of case 2

No.	Model	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1	LR: Linear	13.289	383.902	260.725	2.112
2	LR: Interactions linear	13.564	369.345	262.075	0.899
3	LR: Robust linear	9.747	377.425	202.106	1.172
4	LR: Stepwise linear regression	13.564	369.345	262.075	1.027
5	Optimized Ensemble of trees	10.986	333.148	174.674	147.850
6	Optimized SVR	9.781	368.052	201.550	372.390
7	Optimized GPR	9.187	306.598	157.714	2,163.20
8	ML-FNN	8.504	314.900	159.798	494.448
9	NARX	41.521	1,022.400	654.549	480.212
10	LSTM	21.905	634.677	342.914	43,783.0
11	Benchmark model: DP	61.310	2,114.300	1,073.100	0.013

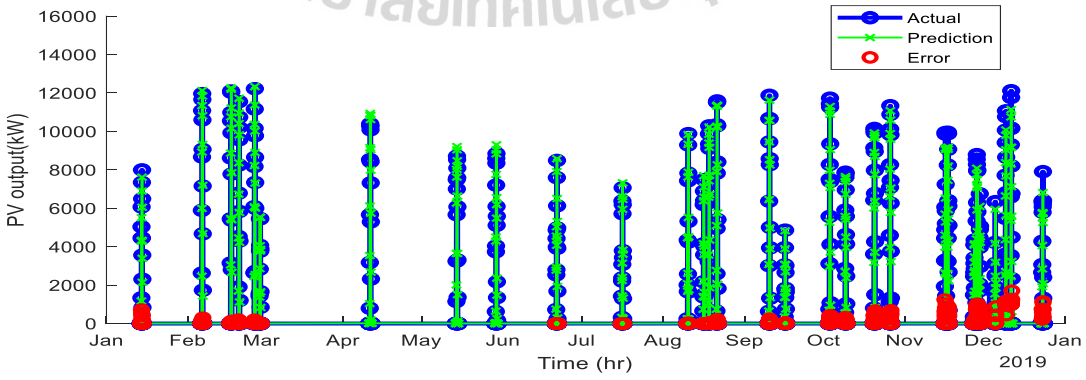


Figure 3.7 The forecasting results of case 2

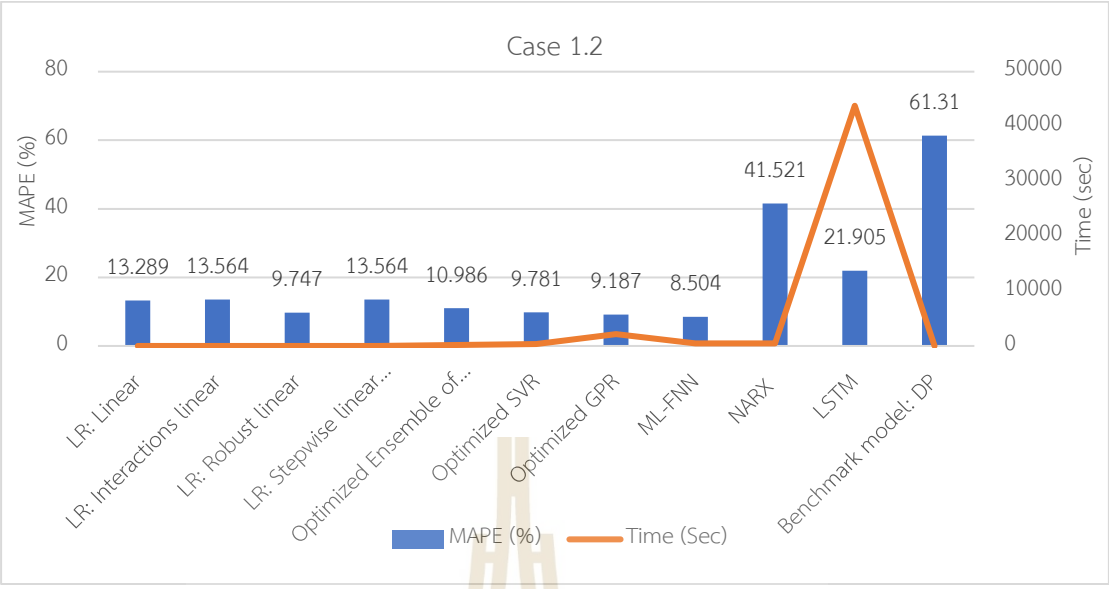


Figure 3.8 Summary of case 2

Figure 3.8 reveals that the most accurate model for photovoltaic (PV) forecasting, concerning the use of holdout (70:30), is the multi-layer feedforward neural network (ML-FNN). This model exhibits a mean absolute percentage error (MAPE) of 8.504% and a training time of 494.448 seconds.

3) Case 3: 5-fold cross-validation

Table 3.8 The forecasting results of case 3

No.	Model	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1	LR: Linear	13.289	383.902	260.725	1.685
2	LR: Interactions linear	13.564	369.345	262.075	1.301
3	LR: Robust linear	9.747	377.425	202.106	1.218
4	LR: Stepwise linear regression	13.564	369.345	262.075	0.907
5	Optimized Ensemble of trees	11.220	337.051	176.242	44.994
6	Optimized SVR	16.699	448.622	316.743	10,322.000
7	Optimized GPR	8.793	304.422	153.081	6,819.400
8	ML-FNN	9.218	317.448	171.287	76.235
9	Benchmark model: DP	61.310	2,114.30	1,073.10	0.013

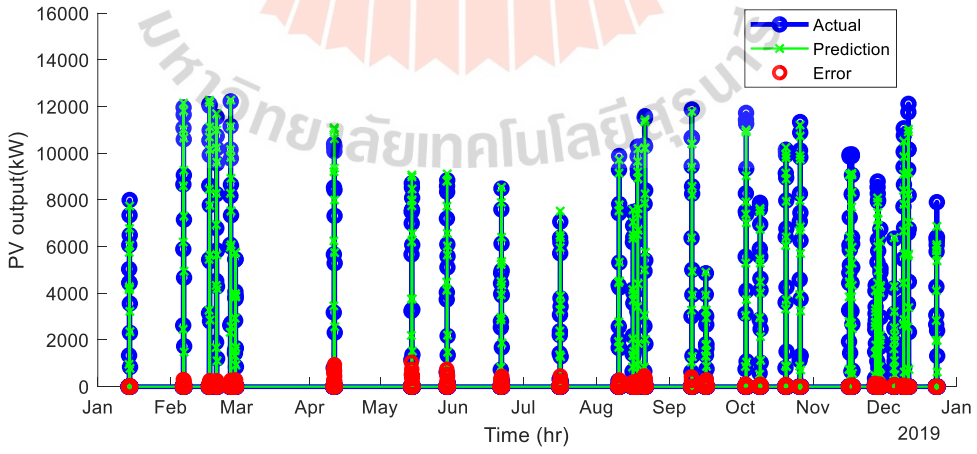


Figure 3.9 The forecasting results of case3 (Optimized GPR)

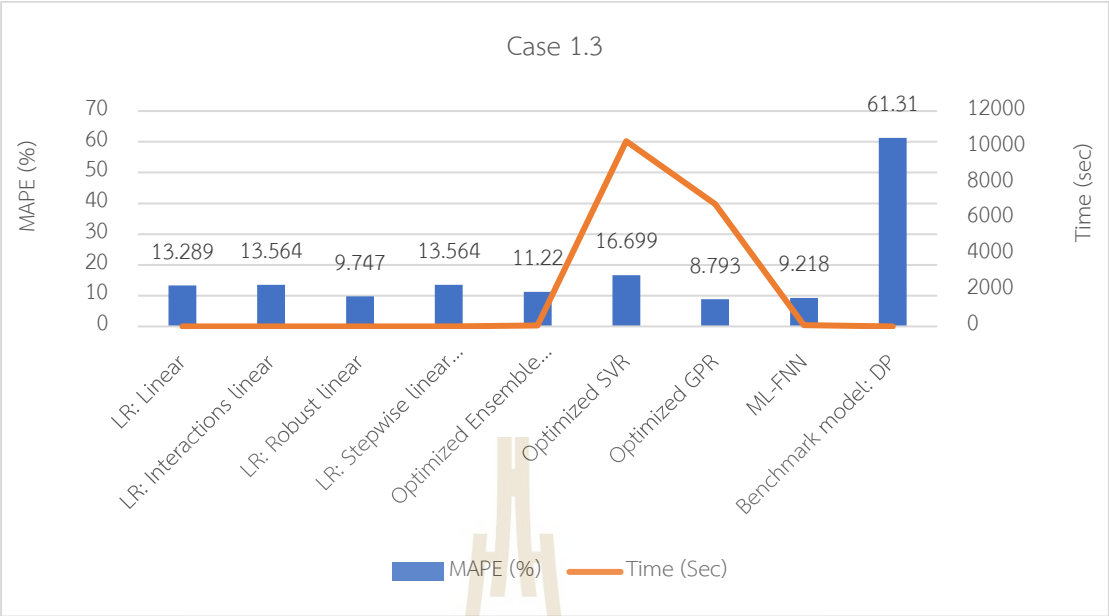


Figure 3.10 Summary of case 3

The results presented in Figure 3.10 demonstrate that the optimized Gaussian Process Regression (GPR) model is the most accurate for photovoltaic (PV) forecasting when using 5-fold cross validation. This model achieves a mean absolute percentage error (MAPE) of 8.793% and a training time of 6,819.400 seconds. It is noteworthy that ML-FNN also demonstrates good performance, with a MAPE of 9.218% and train time of 76.235 seconds. These findings indicate the effectiveness of the optimized GPR model for PV forecasting, while also highlighting the potential utility of ML-FNN for this purpose.

4) Case 4: 10-fold cross-validation

Table 3.9 The forecasting results of case 4

No.	Model	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1	LR: Linear	13.289	383.902	206.725	1.180
2	LR: Interactions linear	13.564	369.345	262.075	1.029
3	LR: Robust linear	9.747	377.425	202.106	1.873
4	LR: Stepwise linear regression	13.564	369.345	262.075	1.619
5	Optimized Ensemble of trees	10.923	330.336	172.168	72.138
6	Optimized SVR	12.290	384.386	229.845	1324.900
7	Optimized GPR	9.178	331.978	161.851	19,617.0
8	ML-FNN	9.219	317.445	171.287	263.890
9	Benchmark model: DP	61.310	2,114.300	1,073.10	0.013

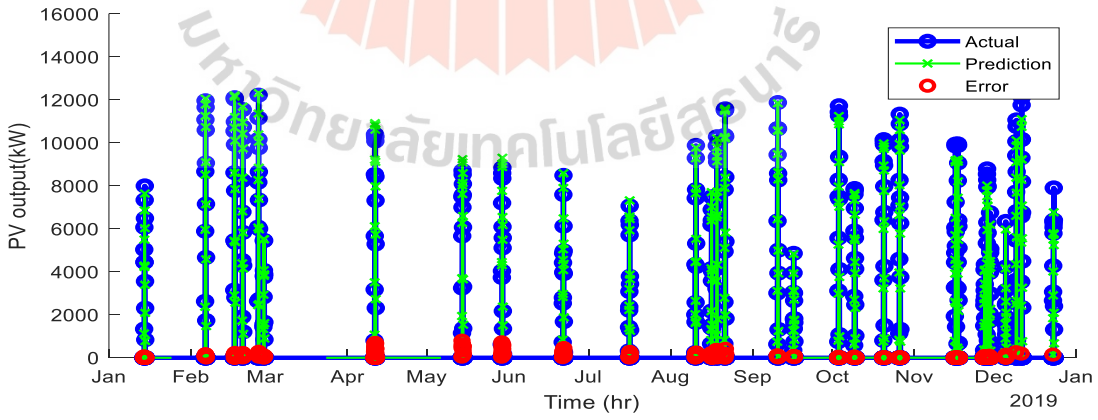


Figure 3.11 The forecasting results of case 4 (Optimized GPR)

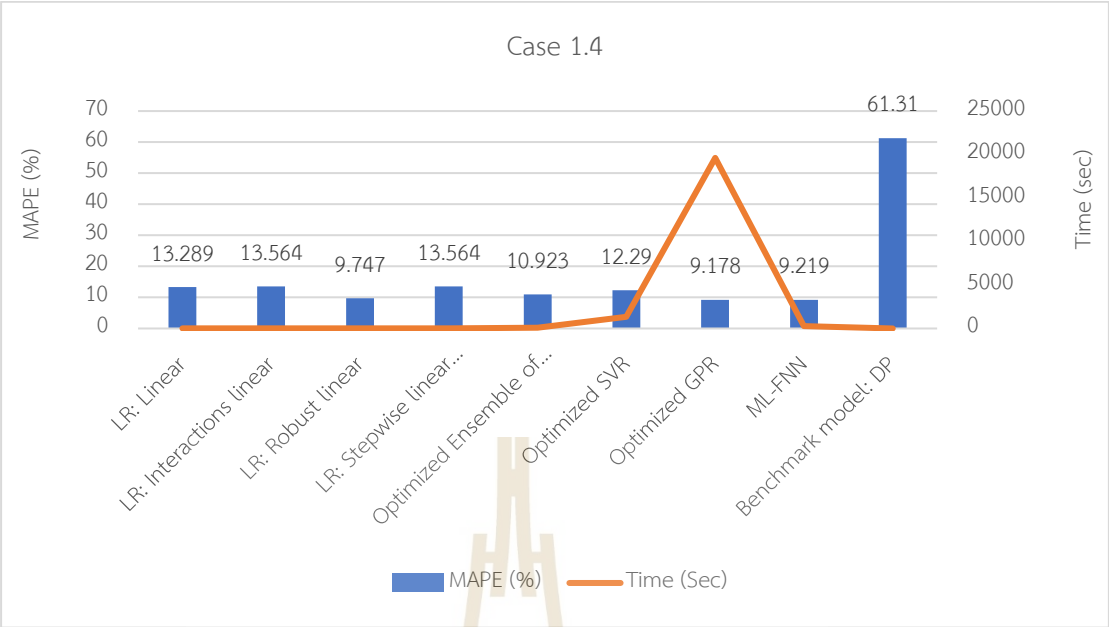


Figure 3.12 summary of case 4

The results presented in Figure 3.12 demonstrate that the optimized Gaussian Process Regression (GPR) model is the most accurate for photovoltaic (PV) forecasting when using 5-fold cross validation. This model achieves a mean absolute percentage error (MAPE) of 9.178% and a training time of 19,617.000 seconds. It is noteworthy that ML-FNN also demonstrates good performance, with a MAPE of 9.219% and train time of 263.890 seconds. These findings indicate the effectiveness of the optimized GPR model for PV forecasting, while also highlighting the potential utility of ML-FNN for this purpose.

From Figure 3.5-3.12, the simulation results show that the MAPE of LR models is between 9-13 %, and the accuracy of each model is close to the same. Moreover, LR models are the fastest training model. Robust LR is the highest forecasting accuracy among LR models because reweight process in Robust LR make reduce the sensitivity of the outlier to the model. Among ML models (Ensemble of trees, SVR, GPR), GPR is the highest forecasting accuracy model with a k-fold cross-validation method. Nerveless, this also model takes the highest time to train because many parameters and matrices have to be calculated in this model. For, SVR is well-known as good for

classification tasks and can be used to forecast regression tasks. However, the performance of this model is quite low when compared to another method. SVR worked well with the holdout validation method. The performance of an ensemble of trees is not outperformed other methods for this PV forecasting task but it is also not bad because the performance of this model is close to the best method for each case. Among DL models (ML-FNN, NARX, LSTM) with this dataset (hourly), the simulation results can be concluded that ML-FNN outperforms other DL methods, in terms of accuracy and training time. ML-FNN can work-well with both holdout and k-fold cross-validation methods. For NARX and LSTM, these models are time series models that can work-well with related time series problems. Therefore, this work aims to forecast hourly PV power generation that has a big gap in changing per step causing the curve fitting deep learning (ML-FNN) to outperform both. However, if both time series models will be used with higher resolution systems such as 5 mins/step or 15 min/step that have a small gap in changing per step, both models may outperform ML-FNN.

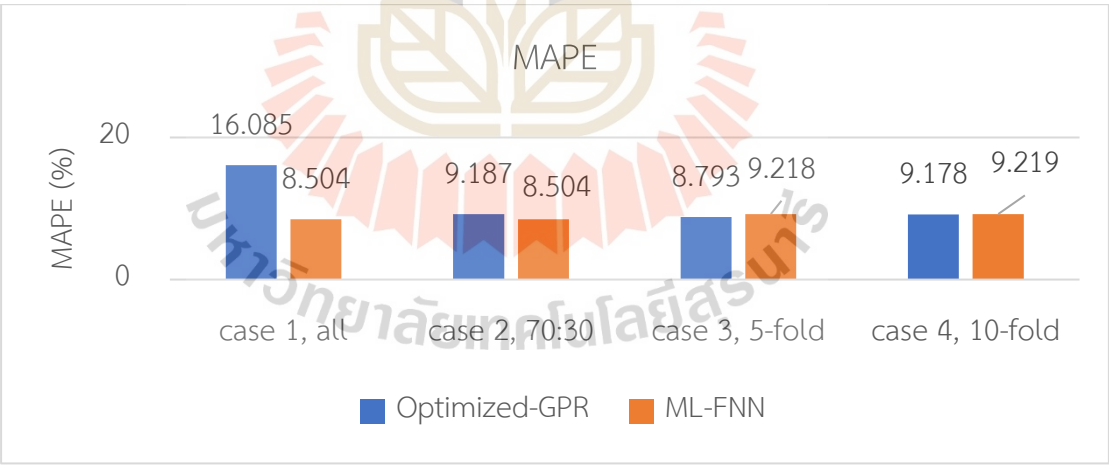


Figure 3.13 Summary of case 1-4 in term of percent

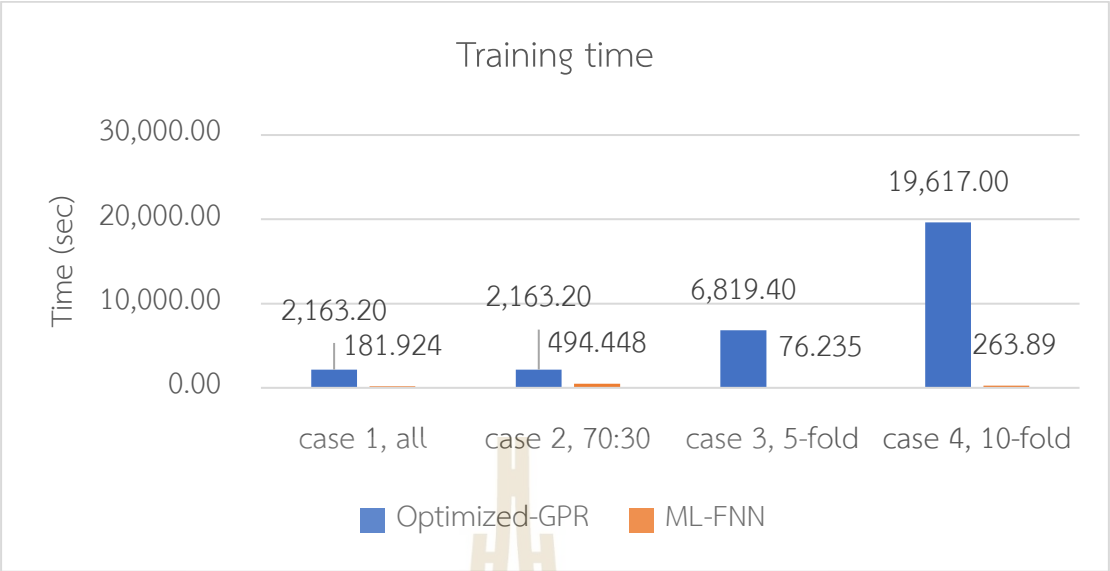


Figure 3.14 Summary of case 1-4 in term of training time

From all validation methods, the simulation results show that ML-FNN and optimized GPR outperform other methods. Figure 3.13 and Figure 3.14 show that ML-FNN is the highest accuracy at MAPE of 8.504% with holdout validation method. Moreover, ML-FNN is more stable than optimized GPR when the validation method was changed and also better in term of training time.

3.5.6 Impact of uncomplete dataset

For the solar floating site, the forecasting models were used to test with 4 cases as solar rooftop sites. All models that were mentioned in chapter 2 will be used to illustrate the performance besides both time series models because there are only 45 days in the dataset that can be used for the forecasting model. Then, there is a big gap between the selected test sets cause of the inappropriate use of time series models. The best forecasting results of the industrial site are shown in Table 3.7-3.10 and Figure 3.8-3.11, respectively.

1) Case 1: using all training set as training set and validation set

Table 3.10 The forecasting results of case 1 of SUT dataset

No.	Model	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1	LR: Linear	25.079	52.945	29.786	3.192
2	LR: Interactions linear	25.876	55.271	30.463	0.878
3	LR: Robust linear	25.385	59.730	27.005	1.545
4	LR: Stepwise linear regression	26.579	54.980	30.978	4.039
5	Optimized Ensemble of trees	20.893	78.901	34.702	101.560
6	Optimized SVR	25.172	58.008	26.912	157.860
7	Optimized GPR	19.680	58.788	27.695	381.040
8	ML-FNN	22.076	47.910	26.364	312.825

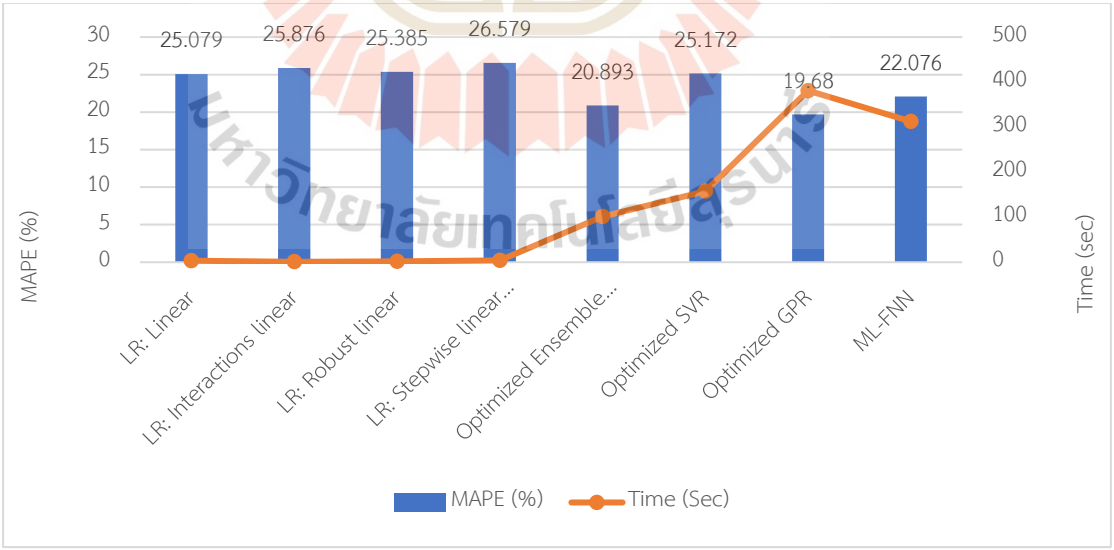


Figure 3.15 Sumary of case 1

2) Case 2: using training set and validation set as 70:30

Table 3.11 The forecasting results of case 2

No.	Model	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1	LR: Linear	25.079	52.945	29.786	2.654
2	LR: Interactions linear	25.876	55.271	30.462	1.279
3	LR: Robust linear	25.385	59.730	27.005	0.826
4	LR: Stepwise linear regression	26.579	54.980	30.978	1.457
5	Optimized Ensemble of trees	21.150	53.626	27.459	169.230
6	Optimized SVR	18.010	52.489	26.108	196.070
7	Optimized GPR	25.079	52.945	29.786	242.910
8	ML-FNN	19.052	47.833	26.552	8.690

Note: for SVR, sometimes the optimizer cannot find the optimum hyperparameter

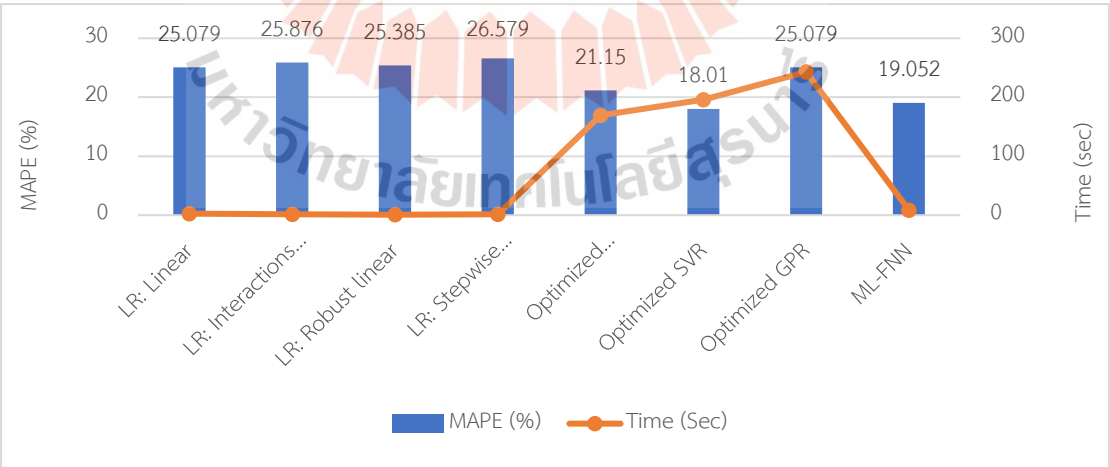


Figure 3.16 The forecasting results of case 2

3) Case 3: 5-fold cross-validation

Table 3.12 The forecasting results of case 3

No.	Model	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1	LR: Linear	25.079	52.945	29.786	1.645
2	LR: Interactions linear	25.876	55.271	30.463	1.213
3	LR: Robust linear	25.385	59.730	27.005	1.327
4	LR: Stepwise linear regression	26.579	54.980	30.978	371.200
5	Optimized Ensemble of trees	24.098	58.189	30.186	563.980
6	Optimized SVR	25.039	57.880	26.977	563.980
7	Optimized GPR	26.189	60.830	33.318	16124.000
8	ML-FNN	19.449	47.508	25.625	27.052

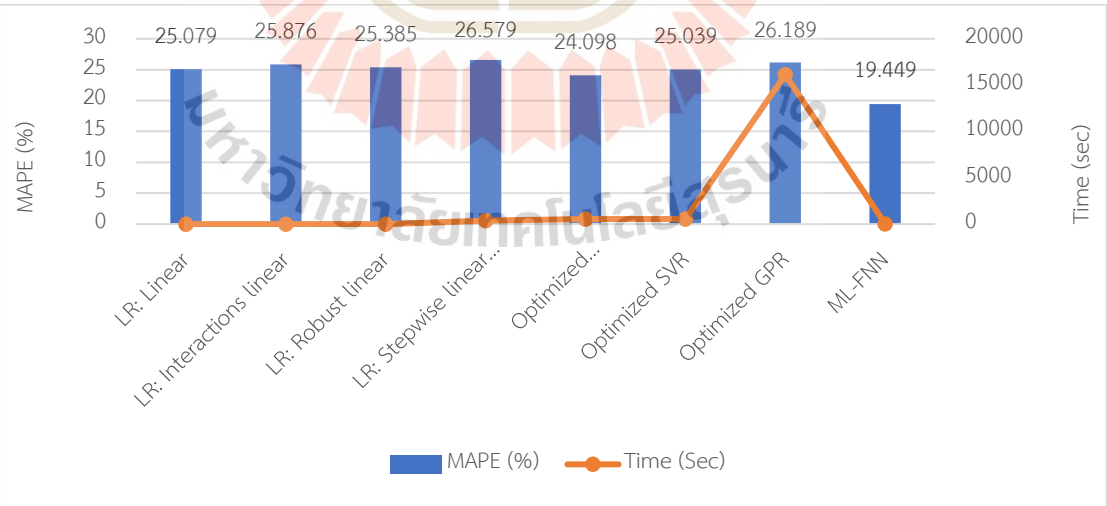


Figure 3.17 The forecasting results of case 3 (ML-FNN):

4) Case 4: 10-fold cross-validation

Table 3.13 The forecasting results of case 4

No.	Model	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1	LR: Linear	25.079	52.945	29.786	1.323
2	LR: Interactions linear	25.876	55.271	30.463	1.055
3	LR: Robust linear	25.385	59.729	27.005	1.158
4	LR: Stepwise linear regression	18.630	51.354	25.151	286.510
5	Optimized Ensemble of trees	152.594	335.350	290.64	687.230
6	Optimized SVR	182.594	335.350	29.648	691.122
7	Optimized GPR	21.703	56.978	29.278	17724.0
8	ML-FNN	19.905	50.774	27.227	33.101

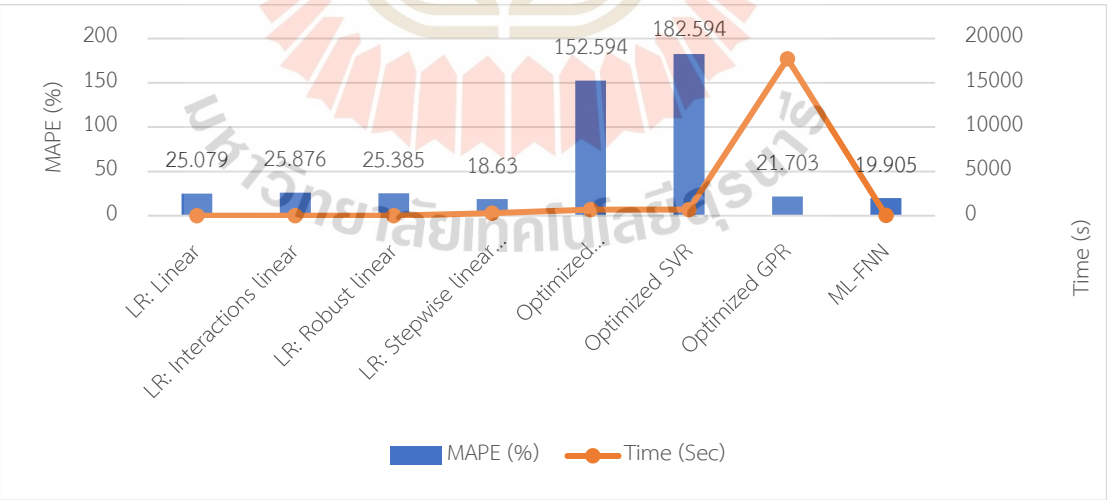


Figure 3.18 Summary of case 4

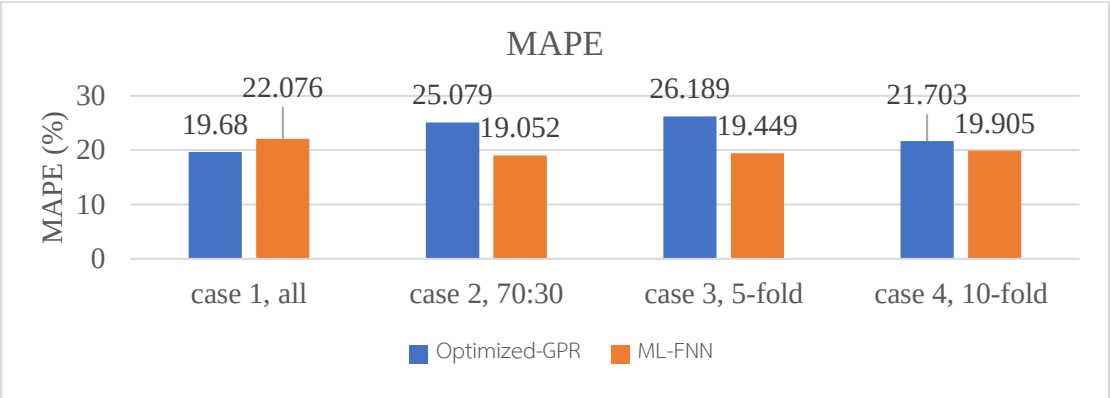


Figure 3.19 Summary of incomplete dataset in term of accuracy

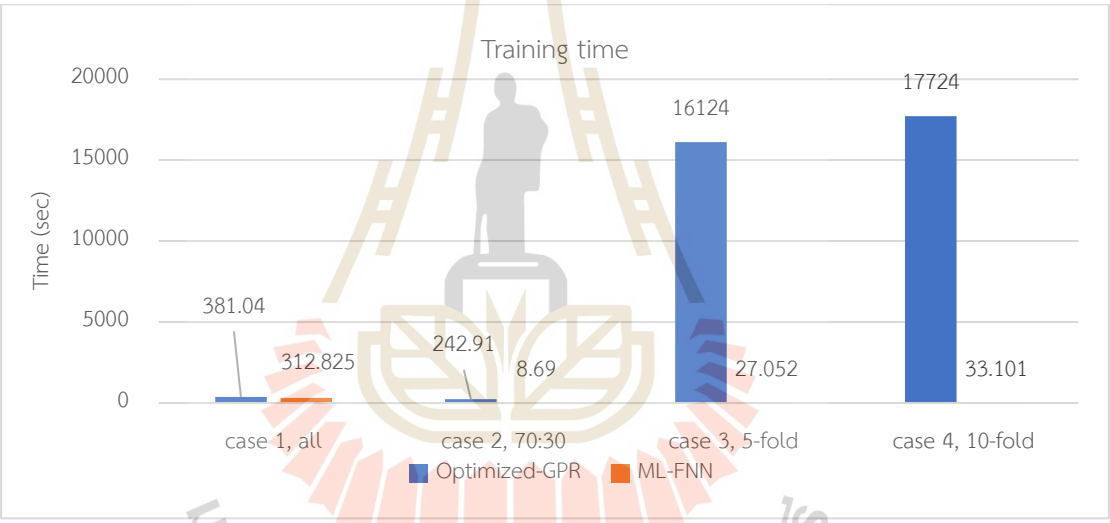


Figure 3.20 Summary of incomplete dataset in term of training time

Based on the simulation results presented in Figure 3.15-3.20, it can be observed that the trend of accuracy for each forecasting model applied to the solar floating site is similar to that of the solar rooftop site. However, the overall forecasting performance for this dataset is lower compared to the solar rooftop site due to the presence of a large amount of missing data. Additionally, it is important to note that the test set used in this study was randomly selected, which introduces variance from the training set.

3.6 Conclusion

In this chapter, a comparative study of various short-term photovoltaic (PV) forecasting methods was conducted to evaluate the appropriate hyperparameter adjustment for each model. The chapter begins by describing the workflow of the comparative study, followed by a discussion on the imported datasets and data preprocessing techniques. The study was divided into six cases, including 1) the impact of hyperparameter tuning, 2) the impact of activation function selection, 3) the impact of normalization techniques, 4) the impact of seasonal and test set selection, 5) the impact of validation methods, and 6) the impact of incomplete datasets. The simulation results demonstrate that these case studies have an impact on the accuracy or training time of data-driven-based forecasting models. The results also show that ML-FNN with holdout validation method outperforms other forecasting methods.

3.7 References

Faruque, M. O., Rabby, M. A. J., Hossain, M. A., Islam, M. R., Rashid, M. M. U., & Muyeen, S. M. (2022). A comparative analysis to forecast carbon dioxide emissions. *Energy Reports, 8*, 8046-8060. doi:https://doi.org/10.1016/j.egyr.2022.06.025

Junhuathon, N., & Chayakulkheeree, K. (2021, 20-22 Oct. 2021). *Comparative Study of Short-Term Photovoltaic Power Generation Forecasting Methods.* Paper presented at the 2021 International Conference on Power, Energy and Innovations (ICPEI).

Mongkol Treekijjanon, N. J., Uthen Leeton and Thanatchai Kulworawanichpong. (2020). Suitable Energy Management Strategy for the Large Factory in Thailand Using Practical Load Profile. *GREATER MEKONG SUBREGION ACADEMIC AND RESEARCH NETWORK (GMSARN), Volume 14, Number 2, June 2020* 7.

CHAPTER 4

Deep-Learning-Based Short-Term Photovoltaic Power Generation Forecasting Using Improved Self-Organization Map Neural Network

4.1 Background

Accurate forecasting of PV power generation is essential for an energy management system for distributed energy resources, efficient operation in distribution systems, and minimizing potential negative consequences of PV systems. This chapter describes an alternate method for improving the accuracy of short-term PV power-generation forecasting models based on deep learning by clustering the input data using a self-organization map (SOM). To verify the proposed model, LSTM, ML-FNN, ML-FNN-SOM, and LSTM-SOM were evaluated and compared with hourly datasets spanning one year (8,760 samples). RMSE, MSE, and MAPE were used as parameters evaluated. The results demonstrate that the suggested method provides a more precise forecast of solar power generation than alternative methods. Moreover, the proposed method can operate well with a minimal number of inputs.

4.2 Introduction

According to the literature review, LSTM and ANN are the most commonly used forecasting models for solar power generation in current research. These models can be improved by using data preprocessing or statistical techniques and by including additional climate variables in the analysis. However, many small to medium-sized PV systems lack accurate measurements and historical data on variables such as cell temperature and irradiation angle, which are important for predicting energy generation. To address this issue, a DL-based forecasting model that approximates the

relationship between these factors was proposed. This technique uses SOM to enhance clustering efficiency and depict the relation between two or more parameters as numerous states. By estimating unmeasured and related factors as inputs, this method provides an alternate technique for improving the performance of a DL-based forecasting model with few inputs. The proposed method offers several novel aspects that can enhance the accuracy of the predictive model as follows: 1) The proposed method is suitable for photovoltaic power facilities that monitor only the ambient temperature and irradiance. The sensor data is more reliable than the available weather website, which considerably increases the accuracy of forecasts. 2) SOM was used to Classify the level of correlation between latent variables and PV power generation. And 3) the proposed method can be used as an alternate strategy for improving the precision of photovoltaic power generation forecasts.

4.3 Methodology

The current forecasting models for PV power generation rely on historical data and weather forecasts, which are often available through public weather websites. However, some crucial factors such as cell temperature and sky classification are not always available, which can lead to inaccurate forecasting. To improve the precision of PV forecasting, a new model proposes using a relative state factor from SOM as an input to the forecasting model. This approach estimates unmeasured relative components by clustering measured input and can effectively handle time series regression and classification problems. The model includes dataset preparation, estimation of unmeasured factors, and statistical analysis using ANN clustering-based preprocessing. This section will provide a detailed explanation of this approach.

In this section, we will describe the ANN clustering-based preprocessing method, which includes dataset preparation, estimation of unmeasured factors, and statistical analysis. To ensure the accuracy of the data and eliminate outliers, historical data was processed for the PV forecasting framework. The processed data was then

categorized using SOM, as shown in Figure 4.1, and both the processed and clustered data were used to train the DL model for higher efficiency.

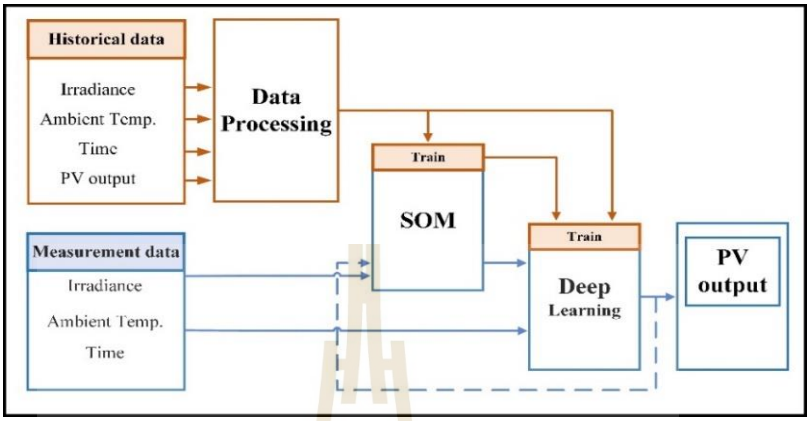


Figure 4.1 Conceptual framework for forecasting PV power generation

4.3.1 APPROACH OVERVIEW

The first step in developing a model for predicting PV power generation is to gather and preprocess historical data related to the plant's power output and meteorological data, including metrics such as solar irradiance and temperature. However, some important parameters may not have been measured. To address this, the existing dataset can be clustered using a clustering method to provide input for a DL-based forecasting model. Past data for each hour of the day can be clustered into various "states" of unobserved parameters to identify elements related to irradiance, time of day, weather, and past PV power generation. The state is determined using a SOM algorithm and used as one of the inputs for a DL-based forecasting model. The proposed framework will be compared to a recently developed method to evaluate its performance.

4.3.2 DATA PREPROCESSING

To prepare the data for the DL model, two fundamental processes are necessary: data cleansing and preprocessing. The goal of data preparation is to make

the data ready for use with the DL model. In contrast, data cleaning is focused on improving the quality of the data, which involves handling null or outlier data. Data completeness or data cleaning is the process of verifying and correcting inaccurate or incomplete data entries from a dataset or database. Since inaccurate data can impact the reliability of the database, it is essential to correct, update, or delete these errors to ensure data accuracy. For this study, data with zero solar irradiance were removed as they may affect the DL model parameters during the training process.

The procedure for removing these data is illustrated in Table 4.1, where *x* represents the input matrix, *y* represents the output matrix, *k* corresponds to the number of inputs, and $\hat{\boldsymbol{x}}$ is the input matrix post-preprocessing.

Table 4.1 Data preprocessing process

Input: $x \in R^{k \times j}$ contain time(j), temperature(j) and irradiance(j), PV output at step (j-1)
Output: $y \in R^{j}$ PV output at the previous step (j)
1: j =1, ..., 8760
2: k = 4
3: $\hat{x} = \{\text{Remove } x(all, j) \text{ and } y(j) \text{ if } x(irr(j), j) = 0\}$
$\hat{x} \in R^{k \times j_{new}}$ is an $(k \times j_{new})$ matrix

Once the dataset has undergone the data preprocessing stage, it proceeds to the clustering phase. Clustering involves grouping similar items together into input clusters of the same type. Two commonly used clustering techniques are the K-means algorithm and Self-Organizing Maps (SOMs). SOMs work in a similar way to K-means but with the added challenge of determining centroids using input vector categories, which can be tricky in this study.

In this stage, the dataset was grouped using a SOM. A SOM trains an ANN to provide clustering based on pattern similarities and related topology by exposing it to patterns. This is beneficial for data analysis and simplification before further processing. ANNs have demonstrated their effectiveness as classifiers and are ideally

suited for handling nonlinear issues. Given the occurrence of nonlinear behaviors in the real world, such as sorting, ANNs are unquestionably a great contender for tackling the clustering problem. For a SOM to cluster a dataset, the input data for clustering issues are prepared. Each row of the input matrix would have the same number of elements as the component being calculated. For this study, the four assessed variables (i.e., time, temperature, irradiance, and PV output) will be supplied into a SOM network, which will transfer the data j_{new} sample to a two-dimensional layer of neurons. Input data for clustering problems are formatted as a matrix. Each row j^{th} of The input matrix will have *k* members that correspond to a vector derived from the PV plant data. Then, there are k rows in each j^{th} sample as an input set. Defining the number of neurons in each layer dimension enables SOM to categorize samples to acquire the dataset's state parameter. In a hexagonal grid, a layer of two-dimensional neurons was utilized. Using more neurons produces higher resolution, while the addition of dimensions enables the modeling of the topology of more intricate function spaces. For the SOM process, given input x^j, the i^{th} unit is found with the closest weight vector W_j^i by competition and $W_j^i x^j$ will be the maximum for each unit j^{th} in the neighborhood *N(i)* of winning neuron *i* to update the weights of *j* (W_j), and the weights outside of *N(i)* are not updated (Table 4.2). The SOM has three stages: 1) competition, 2) collaboration, and 3) weight update. For the competition stage, the most similar unit *i(x)* is found with Equation 1:

$$i(x) = \arg\max_j \| x - W_j \|_2 \quad , \tag{4.1}$$

Where *j*=1, 2, ..., m, and m=samples. For the collaboration state, the lateral distance d_{ij} between the winner unit *i* and unit *j* is used in Equations 4.2 and 4.3:

$$h_{i,j}(d_{ij}) = \exp(\frac{-d_{ij}^2}{2\delta^2}) \, , \tag{4.2}$$

$$\delta(n) = \delta_0 \exp(-\frac{n}{T}) \, , \tag{4.3}$$

Where *h* is the neighborhood function, *n* is the number of iterations, and *T* is constant. Weights-updated states are shown in equations 4.4 and 4.5:

$$W_j(n+1) = W_j(n) + \Delta W_j \,, \tag{4.4}$$

$$\Delta W_j = \eta y_j x - g(y_j) W_j \,, \tag{4.5}$$

where η is the learning rate, y_j is the output, and g(y_j) can be found with equation 6:

$$g(y_j) = \eta y_i = \eta h_{ij}(x) \,, \tag{4.6}$$

Table 4.2 Self-organizing map process

Input: $\hat{x} \in R^{k \times j_{new}}$ is an $(k \times j_{new})\, matrix$
Output: $i(\hat{x})$ is neighborhood *i* as equation 1
1: **for** j=1:1:8760
2: $N(\hat{x}_j) = \{i, if\ i(\hat{x}_j)\, closest\, to\, N(i)\}$ as shown in Figure 1.

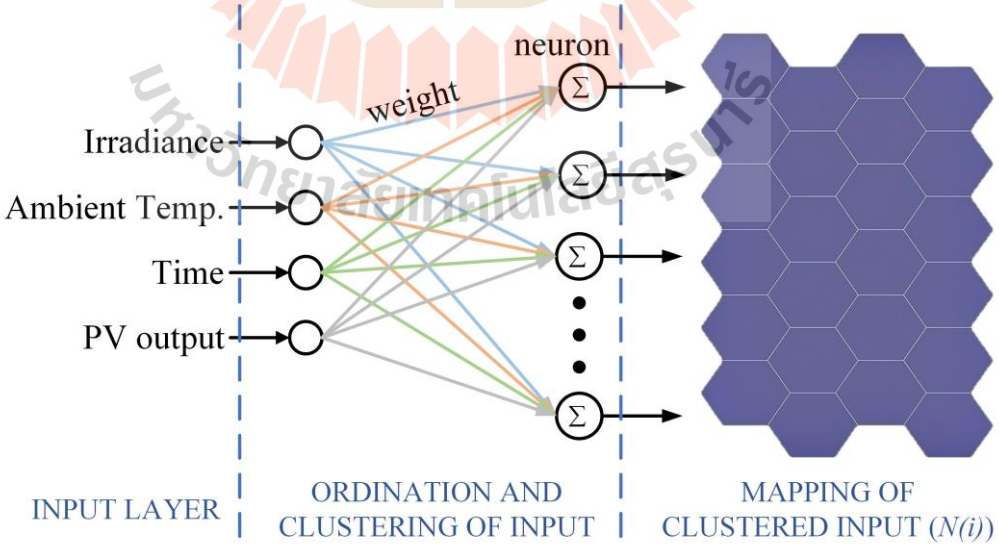


Figure 4.2 SOM structure

Figure 4.2 depicts the neuron-to-neuron connections. Typically, nearby samples are categorized using neighborhood. The SOMs' topology consists of *i* neurons organized in a hexagonal grid. Each neuron has acquired the capability to represent a unique state class, with neighboring neurons typically expressing the same class.

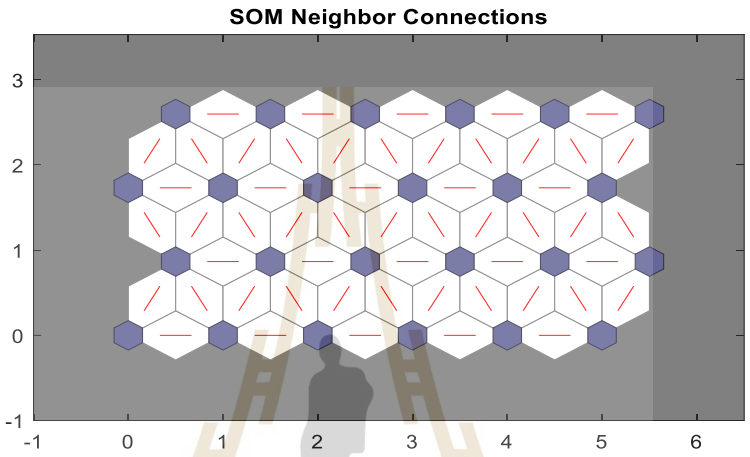


Figure 4.3 SOM Neighbor Connections

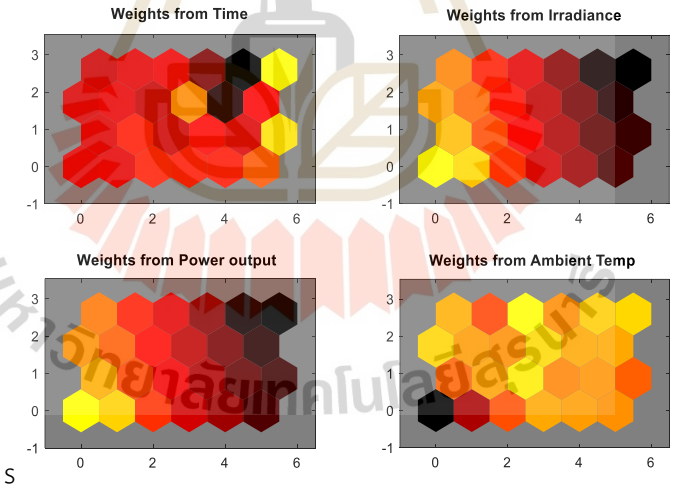
4.4 Results and Discussion

The results of the simulation will be presented in two parts: (1) the categorization using SOM and (2) the forecasting results of the proposed framework. The weight planes of the SOM classes are shown in Figure 4.4 (a), which represents a visual representation of the weights that link each input to one of the 24 neurons in the 6x4 hexagonal grid. Darker hues indicate heavier weights, while lighter hues indicate lighter weights. When the weight planes of four variables are similar, there is a strong correlation between them. Additionally, it was observed that power output and solar irradiance have equal weight, which is contrary to the pattern observed with temperature.

During the peak of light intensity and power output, there is a substantial impact over time. Figure 4.4 illustrates the Euclidean distance between the class of

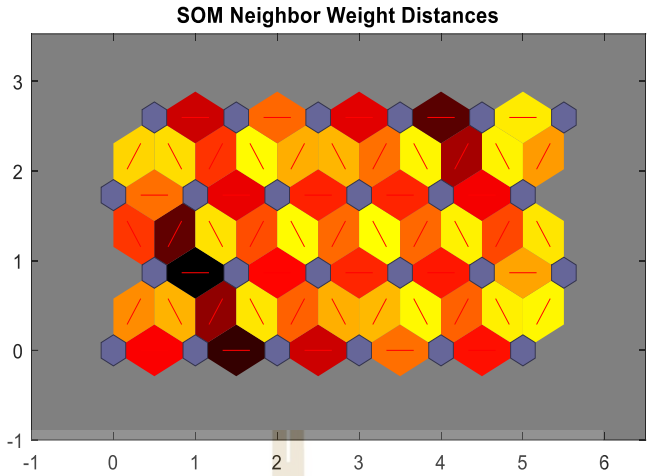
each neuron and its nearby regions. The bright links represent regions in the input space that have a strong connection. On the other hand, the black links indicate groups that represent parts of the feature space that have few or no members. Large barriers with dark connections that divide significant areas of the input space indicate that the groups on either side of the boundary have notably different characteristics.

Figure 4.5 displays the number of members within each class, along with the classes associated with each neighborhood. Areas of high neural activity correspond to groups that are similar in densely populated areas of the feature space. In contrast, areas with low activity indicate sparsely populated parts of the feature space. Through the analysis of the weights in Figure 4.4 and the clustering results in Figure 4.5, it was discovered that the classes were evenly distributed on the 4x6 plane but had a greater proportion than the other classes, indicating that the solar cells located in the upper-right plane were more likely to generate energy.



S

(a)



(b)

Figure 4.4 (a) Weights from inputs; (b) SOM neighbor weight distances

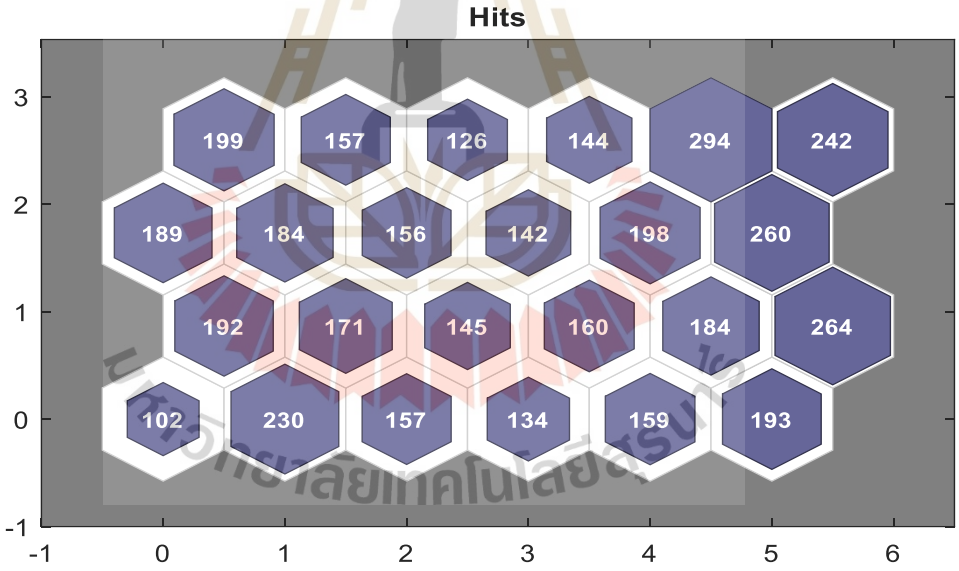
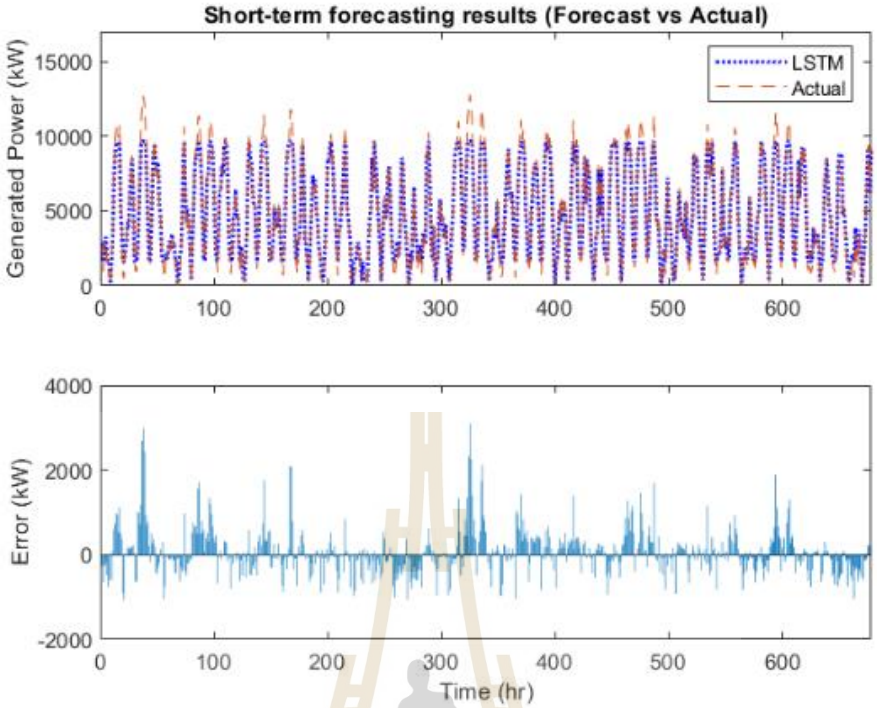
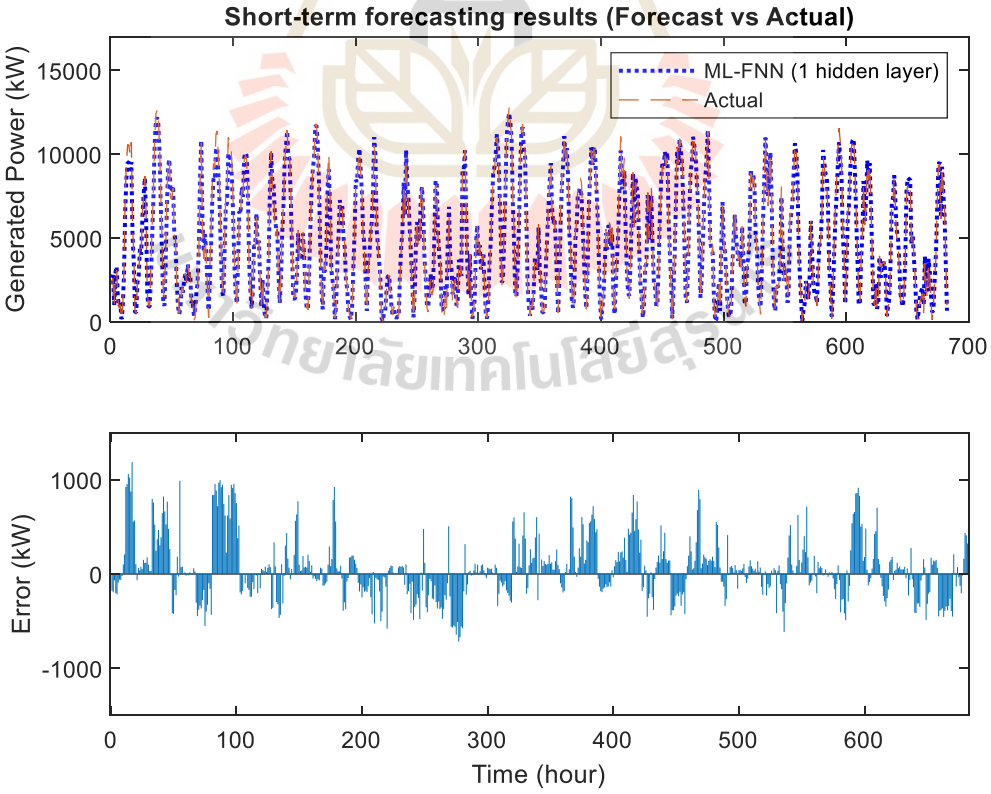


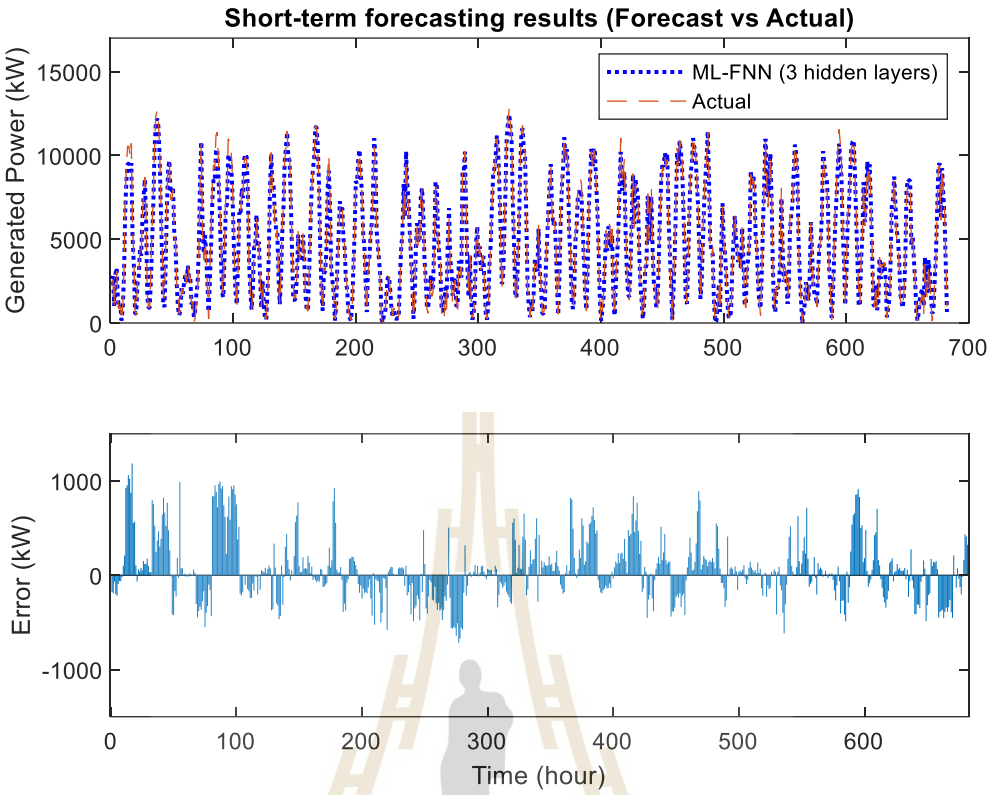
Figure 4.5 Sample hits of SOM


Short-term forecasting results (Forecast vs Actual)
Generated Power (kW)
LSTM
Actual
Error (kW)
Time (hr)

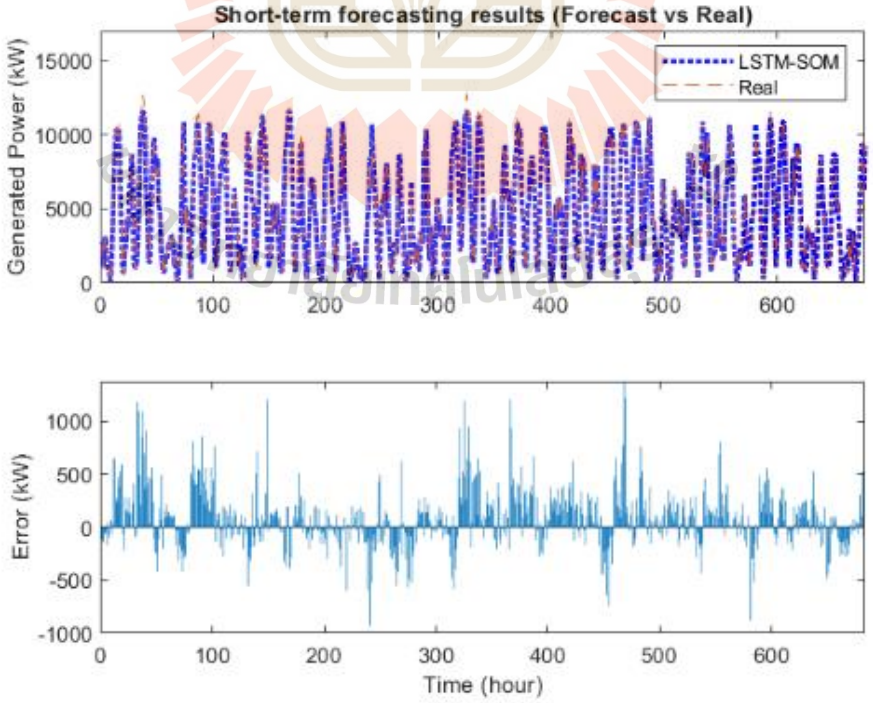
(a)


Short-term forecasting results (Forecast vs Actual)
Generated Power (kW)
ML-FNN (1 hidden layer)
Actual
Error (kW)
Time (hour)

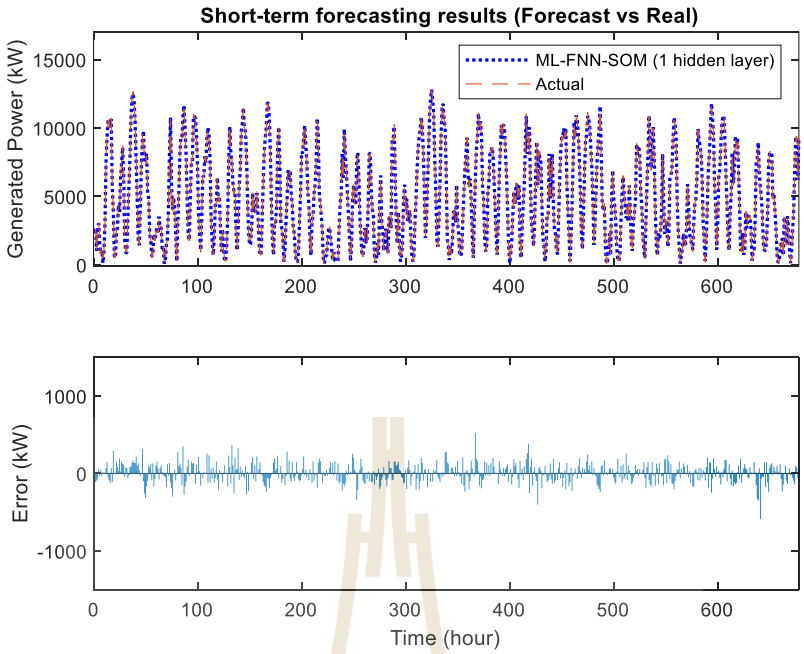
(b)


Short-term forecasting results (Forecast vs Actual)
ML-FNN (3 hidden layers)
Actual
Generated Power (kW)
15000
10000
5000
0
0
100
200
300
400
500
600
700
Error (kW)
1000
0
-1000
0
100
200
300
400
500
600
Time (hour)

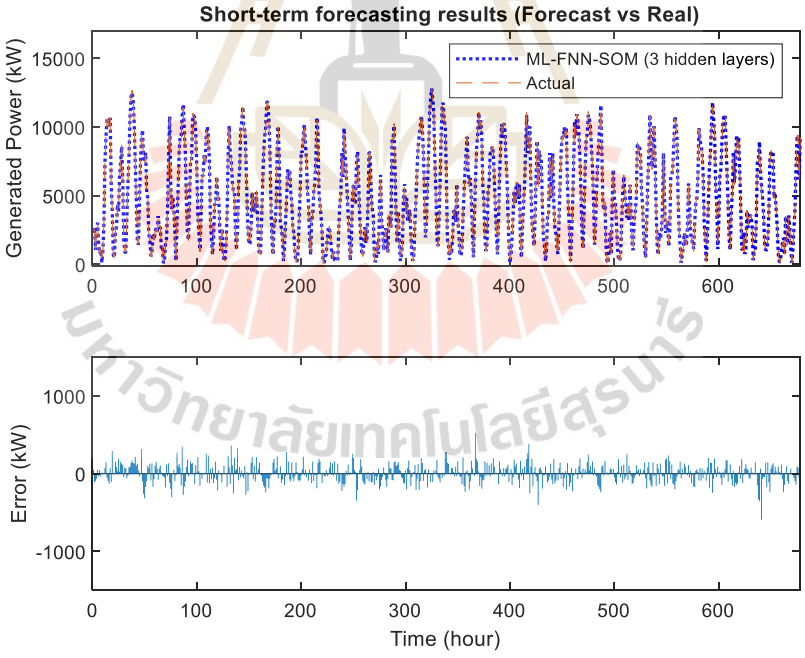
(c)


Short-term forecasting results (Forecast vs Real)
LSTM-SOM
Real
Generated Power (kW)
15000
10000
5000
0
0
100
200
300
400
500
600
Error (kW)
1000
500
0
-500
-1000
0
100
200
300
400
500
600
Time (hour)

(d)



(e)



(f)

Figure 4.6 The results of (a) ML-FNN (1 hidden layer); (b) LSTM, (c) ML-FNN (3 hidden layers); (d) LSTM-SOM; (e) ML-FNN-SOM (1 hidden layer) (f) ML-FNN SOM (3 hidden layers)

To verify that the proposed model can be forecast at the same performance, ML-FNN-SOM (3 hidden layers) was trained and tested 30 times, and the simulation are results shown in Figure 4.7. We can conclude that the proposed model can achieve almost forecasting accuracy when the model was retrained and test at 30 times.

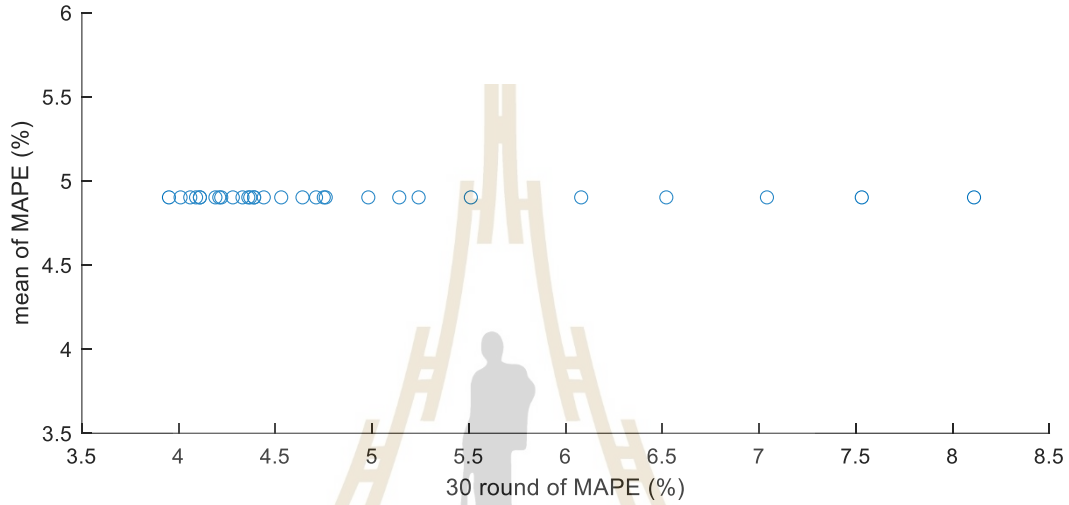


Figure 4.7 The retrain results of ML-FNN-SOM (3 hidden layer)

After the classification of the dataset, the resulting data from the classes were used as one of the DL features in the FNN and LSTM models. The proposed model's prediction results were then compared to those of the unclassified model, but only during the electricity generation period of 678 hours. Figure 4.6 (a) shows the LSTM forecasting results with error values and forecasting results, where the maximum error was 3,104.5 kW and the mean error was 371.77 kW. Figure 4.6 (b) shows the FNN forecasting results with resultant and error values, where the maximum inaccuracy was 853.05 kW and the mean error was 225.28 kW. Figure 4.6 (c) shows the ML-FNN (3 HIDDEN LAYER) forecasting results with resultant and error values, where the maximum inaccuracy was 1,105.70 kW and the mean error was 232.49 kW. Figure 4.6 (d) shows the LSTM-SOM forecasting results with resultant error values, where the maximum error was 1,371 kW and the mean error was 216.95 kW. Figure 4.6 (e) shows the FNN-

SOM forecasting results with resultant error values, where the maximum error was 458.77 kW, and the mean error was 106.69 kW. Figure 4.6 (f) shows the FNN forecasting results with resultant and error values, where the maximum inaccuracy was 1445.70 kW and the mean error was 105.03 kW. The forecast results during the peak period had greater errors than other periods in all cases.

Table 4.3 Comparing the simulation results of ML-FNN/ LSTM/LSTM-SOM/ ML-FNN -SOM

Methods	MAPE (%)	MAE (kW)	RMSE (kW)
ML-FNN (1 hidden layer)	12.92	205.01	266.72
ML-FNN (3 hidden layers)	11.46	240.36	334.285
LSTM	17.38	371.77	551.05
ML-FNN-SOM (1 hidden layer)	4.56	106.69	131.32
ML-FNN-SOM (3 hidden layers)	4.08	95.84	122.84
LSTM -SOM	7.55	216.95	301.18

Table 4.3 presents the results of the simulation, comparing the performance of several models for short-term power forecasting. ML-FNN (1 hidden layer) model achieved a MAPE of 12.92%, a MAE of 205.01 kW, and a RMSE of 266.72 kW. ML-FNN (3 hidden layers) model had a lower MAPE of 11.46%, but a higher MAE of 240.36 kW, and a higher RMSE of 334.285 kW. LSTM model had the highest MAPE of 17.38%, a MAE of 371.77 kW, and an RMSE of 551.05 kW. On the other hand, the ML-FNN-SOM (1 hidden layer) model showed superior performance, achieving a MAPE of 4.56%, a MAE of 106.69 kW, and an RMSE of 131.32 kW. Similarly, the ML-FNN (3 hidden layer)-SOM model performed well, with a MAPE of 4.08%, a MAE of 95.84 kW, and an RMSE of 122.84 kW. Finally, the LSTM-SOM model achieved a MAPE of 7.55%, a MAE of 216.95 kW, and an RMSE of 301.18 kW. These results indicate that the models that

used a clustering method to group the training dataset were more accurate than the conventional methods, particularly during peak power production times.

4.5 Conclusion

This chapter presents an approach aimed at enhancing the precision of deep learning (DL)-based short-term forecasting models for photovoltaic (PV) power generation. The proposed method utilizes self-organizing maps (SOM) and data processing techniques to cluster input data, which were thoroughly tested and compared against commonly employed forecasting techniques. By employing SOM, a set of numerical inputs can be grouped into a single feature during the training process, thereby addressing the issue of curve fitting in the ML-FNN (3 hidden layer) approach. The results obtained from extensive simulations demonstrate the superiority of SOM-based clustering in enhancing the accuracy of PV power generation forecasting, making it a viable alternative to DL-based forecasting methods such as ML-FNN and LSTM. This alternative approach not only outperforms existing techniques but also exhibits exceptional adaptability to situations where the number of available inputs is limited, making it a practical and efficient solution for PV power generation forecasting. The utilization of SOM as a clustering mechanism plays a pivotal role in improving the accuracy of the forecasting models. By organizing input data into cohesive groups, SOM enables the identification of underlying patterns and relationships within the dataset. This clustering approach ensures that the forecasting model captures the intricate dynamics of PV power generation, resulting in more precise predictions. Moreover, the SOM-based clustering technique overcomes the limitations associated with traditional DL-based methods. ML-FNN and LSTM often encounter difficulties in accurately capturing the complexities of PV power generation due to curve fitting issues and the potential loss of important information in the data. However, the integration of SOM mitigates these problems, leading to significantly improved forecasting accuracy.

One notable advantage of the proposed method is its ability to perform effectively with a limited number of inputs. This feature is particularly valuable in practical applications where acquiring a vast amount of data may be challenging or costly. By making accurate predictions using a smaller set of inputs, the SOM-based clustering approach offers a cost-effective and efficient solution for PV power generation forecasting. In conclusion, the introduction of SOM-based clustering combined with data processing presents a pioneering approach for enhancing the accuracy of DL-based short-term forecasting models in PV power generation. The extensive simulations conducted demonstrate its superiority over traditional methods such as ML-FNN and LSTM, making it an appealing alternative for accurate and practical PV power generation forecasting, even with a limited number of inputs.

CHAPTER 5
PROBABILISTIC FORECASTING OF SHORT-TERM PV POWER GENERATION

5.1 Background

In recent years, solar energy has expanded rapidly. Numerous nations have invested in solar energy technology, particularly photovoltaic (PV) energy generation. Increasing solar energy penetration makes solar power forecasting more difficult. Probabilistic forecasting provides more information than traditional point forecasting to account for solar power's inherent uncertainty. In addition, multiple PV sites with spatial-temporal correlations should be considered. This thesis proposed a method to minimize the probabilistic range of PV power generation forecasting. The simulation results will be verified using both Kf-cv and the holdout method.

Forecasting photovoltaic power is difficult since PV power is impacted by several variables, such as irradiance, temperature, etc. (VanDeventer et al., 2019). Consequently, PV power forecasting is now primarily separated into two groups based on the distinct prediction results: certain point/deterministic prediction (Oneto, Laureri, Robba, Delfino, & Anguita, 2018) and uncertainty interval prediction (El-Baz, Tzscheutschler, & Wagner, 2018). In recent years, several researchers have concentrated on the study of deterministic prediction, and artificial intelligence has become a popular technique. To anticipate photovoltaic power, the approach unearths the link between the input factors inherent in the historical output data of solar power plants and the expected outcomes via machine learning. Common artificial intelligence algorithms include mostly of BP neural networks, support vector machines, regression tree techniques, etc (Gao, Li, Hong, & Long, 2019). Recognizing that photovoltaic power generation is significantly influenced by meteorological parameters when the meteorological conditions within the forecast period fluctuate

significantly, the photovoltaic output curve will no longer be smooth and there will be a large peak-to-valley difference, resulting in a significant reduction in the accuracy of the deterministic prediction results (Ahmed, Sreeram, Mishra, & Arif, 2020). Interval predictions may thus compensate for the absence of deterministic forecasts and provide more detailed information. This not only enables decision-makers to comprehend the possible output of the prediction point, but it also enables decision-makers to comprehend the future change trend of the output of the prediction point, vastly enhancing the prediction accuracy and promoting grid planning, risk analysis, and reliability evaluation (Sayed, Elgeldawi, Zaki, & Galal, 2020). Therefore, interval prediction is a useful technique for enhancing the precision of solar power forecasts. In the field of deterministic prediction, Reference (VanDeventer et al., 2019) categorizes meteorological circumstances as either ideal or non-ideal. For ideal weather, the LSTM prediction technique is used; for non-ideal weather, the time-series correlation and features of non-ideal weather types are considered to obtain the final point prediction value. Regarding the constraints and inadequacies of historical PV output data and weather information, Reference (Wang et al., 2020) introduced a day-ahead forecasting approach related to cloud space synthesis to accomplish point prediction. Furthermore, Reference (Li et al., 2018)created independent day-ahead PV power forecasting models based on LSTM and proposed a method to modify the forecasting results of the LSTM model based on the principle of time correlation, which improved the model's prediction accuracy. In the field of probabilistic forecasting, Reference (Ni, Zhuang, Sheng, Kang, & Xiao, 2017) developed an adaptive method of short-term PV power forecasting based on extreme learning machine (ELM) and lower and upper bound estimation (LUBE), as well as an improved differential evolution algorithm to determine the best-generating prediction intervals. Reference (Zhang, Wang, Liao, Zhang, & Zha, 2015)established a novel two-stage model to quantify the forecast interval value of solar power production, combined several neural network models to provide point forecasting values and generated the prediction interval using the kernel density estimation technique. Under the principle of assuring interval coverage, Reference (Raza, Mithulananthan, & Summerfield, 2018) provided a modified Bootstrap

approach to enhance the classic theoretical method, eliminate the issue of incorrect prediction error hypothesis, and minimize the interval width. Reference Furthermore, (Bouzerdoum, Mellit, & Massi Pavan, 2013) suggested a forecasting model based on PSO and boundary theory; the interval prediction of photovoltaic output was achieved by utilizing a PSO to optimize the output weight of boundary estimation theory. ML and DL algorithms dynamically alter their internal settings depending on inputs. These parameters are referred to as "model parameters". Other factors, however, are not modified throughout the learning process, but rather must be preconfigured before the learning process is initiated. These parameters are often known as "hyperparameters." The model parameters specify how the input data are transformed into the intended output, while the hyperparameters describe the model's architecture. The efficiency of an ML and DL model is very sensitive to the selection and settings of its hyperparameters. A variety of techniques could be used to establish hyperparameters for a particular dataset. The first is to manually configure them and determine their accuracy appropriately. Then, various hyperparameter values may be evaluated, and the associated accuracy can be determined for each modification. Configuring the hyperparameter settings manually in this trial-and-error manner is a laborious and time-consuming operation. The default values of hyperparameters that are suggested by the software packages used in the implementation, which are in turn based on recommendations from the literature and experience, may also be utilized to determine an acceptable hyperparameter configuration. Sometimes the default values work well for a particular dataset, but this does not necessarily imply that they provide the highest degree of precision. hyperparameter optimization techniques may be used to achieve the problem. These techniques are data-dependent optimization algorithms that aim to minimize the predicted training error of a machine learning model throughout the search space of possible hyperparameter configurations. The ML algorithms were initially assessed using the default hyperparameter values, followed by a comparison with the outcomes of hyperparameter tweaking methods. When attempting to solve a particular classification issue, the vast majority of published publications examine the impact of one hyperparameter tuning strategy on

the precision of one or more machine language algorithms. Both the classification accuracy of the machine learning approach and the hyperparameters combination that provides the highest classification accuracy is heavily influenced by the nature of the challenge. Prior studies focused mostly on deterministic or point forecasting. Hong and Fan (Rahab, Zitouni, & Djoudi, 2018) theorized that this might be because probabilistic predictions were evaluated using the same performance measures as deterministic forecasts and performed worse than their deterministic equivalent. Chapter 2, which describes the most used performance measures, can be deduced that evaluating probabilistic predictions using metrics designed for point forecasts may result in erroneous findings. Supplying a utility with a PDF or prediction interval, i.e., an interval in which the random variable is projected to be measured with a specified probability of future production and demand is arguably more beneficial than providing a single value since it permits risk management. It should be emphasized that a prediction interval and a confidence interval are not the same, but they are regrettably used interchangeably at times. A prediction interval relates to a random variable, while a confidence interval is related to an unknown parameter and is generated using the data. In probabilistic forecasting, there are often two ways to generate a PDF. First, a density function may be assumed, which is the parametric method. Second is the nonparametric method, which makes no such assumption. Nevertheless, assuming a distribution is seldom indicative of data and is often inaccurate or suboptimal (Rauf et al., 2020).

Probabilistic photovoltaic (PV) forecasting with truncation and Monte Carlo simulation is a technique used to predict the output of a solar PV system with a high degree of accuracy. This approach combines the use of truncated probability distributions with the Monte Carlo simulation method, which involves generating multiple iterations of the forecast to account for the variability in PV output due to changing weather conditions.

Probabilistic PV forecasting with truncation and Monte Carlo simulation relies on statistical models and algorithms to generate truncated probability distributions

that describe the possible outcomes of a PV system's output. These models take into account a range of factors, including historical data, weather patterns, and solar irradiance to predict future outcomes. The use of truncation improves the accuracy of the forecast by capping the upper and lower limits of the output range, while Monte Carlo simulation generates multiple iterations of the forecast to account for the uncertainty in the input parameters. The output of a probabilistic PV forecast with truncation and Monte Carlo simulation typically includes a probability distribution, such as a histogram or density plot, that shows the likelihood of different outcomes occurring within the truncated range. This information can be useful for grid operators, energy traders, and PV system operators, who can use the forecast to better manage their resources and make more informed decisions. Overall, probabilistic PV forecasting with truncation and Monte Carlo simulation is a powerful tool for anyone seeking to maximize the efficiency and profitability of a solar PV system. By combining the benefits of truncated probability distributions with the Monte Carlo simulation method, it provides decision-makers with a more nuanced understanding of the probabilities surrounding the system's output and allows them to optimize their use of this renewable energy source.

5.2 Methodology

The present study involves a stepwise approach to probabilistic forecasting. Firstly, a point forecasting method, namely the ML-FNN (3 hidden layers), is employed to generate a point forecast. Subsequently, the obtained point forecast is used in conjunction with probabilistic computation techniques, based on a training set, to derive the prediction intervals that represent the range of confident predictions for each time step. The ensuing discussion delineates the methodology employed to achieve probabilistic forecasting at each time step and the resulting findings are presented in Figure 5.1, which also includes a comparative analysis.

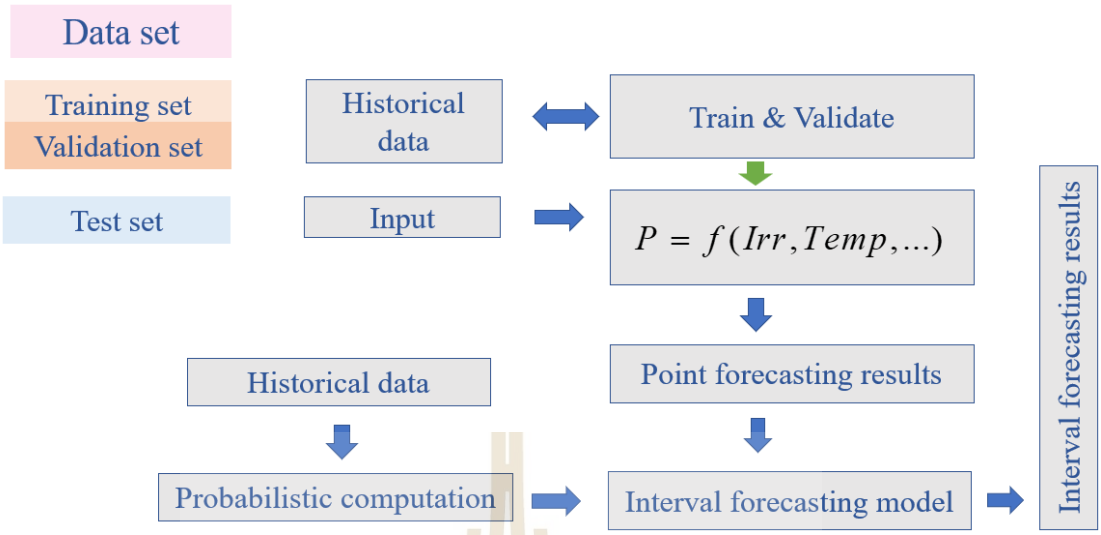


Figure 5.1 Probabilistic computation workflow

5.2.1 Probabilistic computation of the dataset

The first step of probabilistic analysis is to use the point forecasting model that was proposed in chapter 3 (3 hidden layers). Then, the probabilistic of the training set was computed as all, seasons, months, 2 weeks, and 7 days before the test set. After getting the appropriate case, the truncated was used to select the appropriate range and select the lower bound and upper bound of interval forecasting. The process of probabilistic analysis of the dataset is shown in Figure 5.2

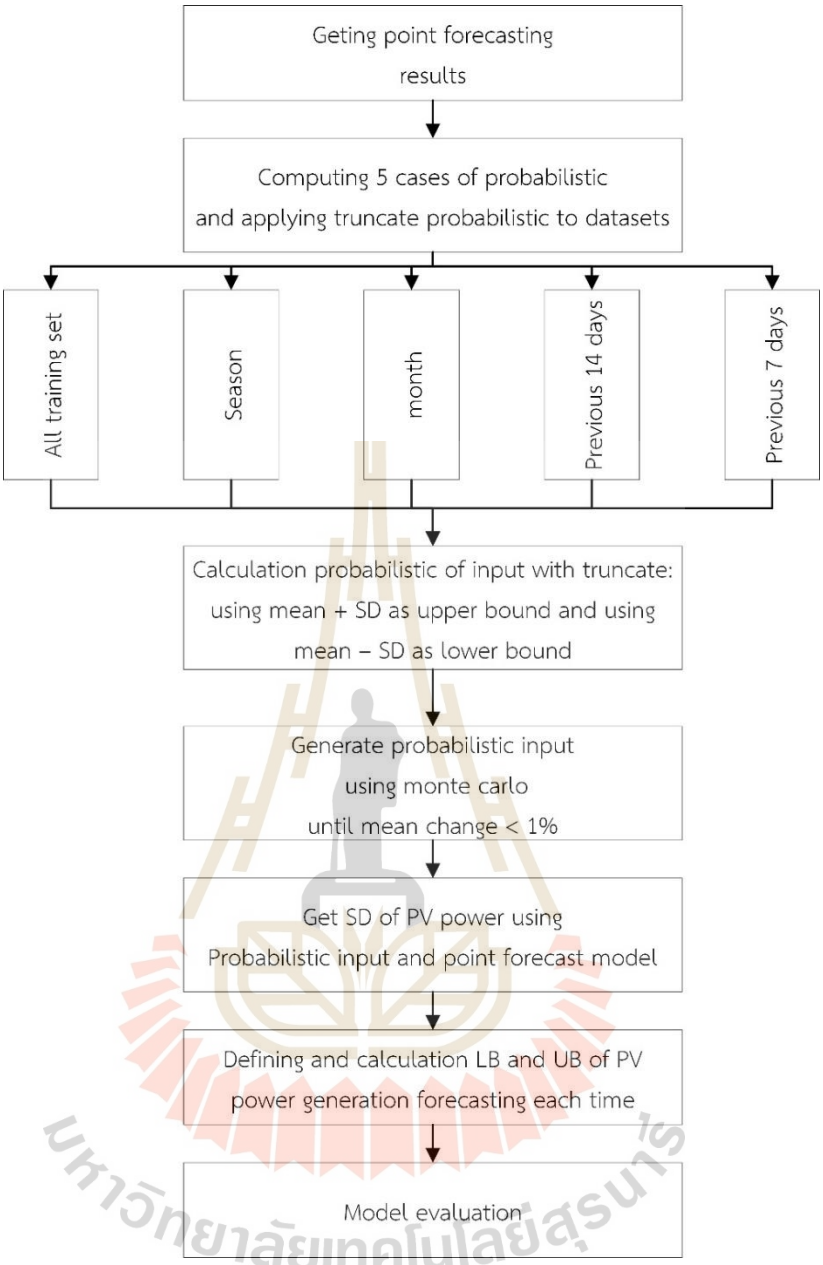


Figure 5.2 Probabilistic analysis of dataset process

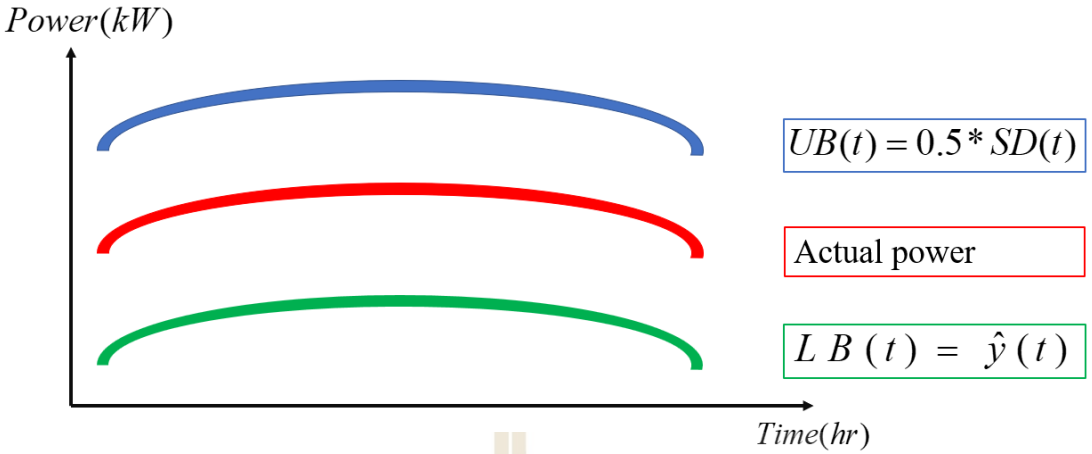


Figure 5.3 Lower bound and upper bound selection of PV power

5.2.2 Truncate probabilistic

Truncation is a concept used in probabilistic forecasting to improve the accuracy of the forecast by capping the upper and lower limits of the probability distribution. In this approach, the probability distribution of possible outcomes is generated using statistical techniques such as Kernel Density Estimation (KDE) or Gaussian Mixture Models (GMM). The distribution is then truncated to a specified range, removing any extreme values outside of the range. Truncation is particularly useful in situations where there is a high degree of uncertainty or where the outcomes can vary significantly. By truncating the distribution to a specific range, the probabilistic forecast becomes more accurate, as it focuses on the most probable outcomes and removes the unlikely or extreme values. This approach also allows decision-makers to better understand the range of possible outcomes and make more informed decisions based on that information. The process of truncation involves setting upper and lower limits on the probability distribution. This can be done using a fixed range or a dynamic range that changes depending on the input data. For example, in solar PV forecasting, the upper limit of the distribution might be set to the maximum expected output of the PV system, while the lower limit might be set to zero or a small negative value to account for measurement error.

The truncated probabilistic workflow is a method used to forecast the output of a system while taking into account the uncertainty associated with the prediction. This approach involves generating a probability distribution of possible outcomes, and then truncating the distribution to a specified range to improve the accuracy of the forecast. The following steps outline the truncated probabilistic workflow: 1) Data Collection: Collect relevant data such as historical performance, external variables, weather patterns, etc. 2) Point Forecasting: Use a point forecasting model such as Artificial Neural Networks (ANN), Support Vector Regression (SVR), or Gradient Boosted Regression Trees (GBRT) to generate a point forecast. 3) Probability Distribution Generation: Using the point forecast, generate a probability distribution of possible outcomes using statistical techniques such as Kernel Density Estimation (KDE) or Gaussian Mixture Models (GMM). 4) Truncation: Truncate the probability distribution to a specified range to improve the accuracy of the forecast. This can be achieved by capping the upper and lower limits of the range to remove any extreme values. 5) Prediction Intervals: Use the truncated distribution to determine the prediction intervals, which represent the range of confident predictions for each time step. 6) Model Validation: Validate the model by comparing the predicted intervals against the actual outcomes. This step ensures that the model is accurate and can be used with confidence.

Overall, the truncated probabilistic workflow is a powerful method that allows decision-makers to make informed decisions in the face of uncertainty. By combining point forecasting with probability distribution generation and truncation, this approach provides a more nuanced understanding of the range of possible outcomes and enables decision-makers to optimize their use of resources and plan for a range of contingencies.

5.2.3 Monte carlo

Monte Carlo is a computational method that involves using random sampling techniques to generate a large number of possible outcomes or scenarios. This method is often used in statistical analysis and mathematical modeling to

calculate the probability of certain events or outcomes. The technique is named after the Monte Carlo Casino in Monaco, where games of chance involve a high degree of randomness and unpredictability. In the Monte Carlo method, a large number of simulations are performed, each using different random inputs or variables. The outputs from these simulations are then aggregated and analyzed to determine the probability of certain events or outcomes. This approach can be used to generate a probability distribution for a particular variable, allowing decision-makers to better understand the likelihood of different scenarios and make more informed decisions. The Monte Carlo method is particularly useful when dealing with complex systems or situations where there is a high degree of uncertainty. For example, it can be used to model the potential outcomes of a financial investment, to simulate the spread of a disease, or to forecast the output of a solar PV system under changing weather conditions. By generating a large number of possible outcomes, the Monte Carlo method can provide decision-makers with a more nuanced understanding of the range of possible outcomes and enable them to plan for a variety of contingencies.

The Monte Carlo workflow is a method used to generate probabilistic forecasts using random sampling techniques. The following steps outline the basic Monte Carlo workflow: 1) Define the Problem: Identify the problem or system to be modeled and determine the input variables, output variables, and assumptions to be made. 2) Define Probability Distributions: Define the probability distributions for each of the input variables, based on historical data, expert knowledge, or assumptions. 3) Sample Inputs: Generate a large number of random samples from the probability distributions for each of the input variables. 4) Simulate System: Run the model or simulation using the sampled inputs to generate a large number of outputs. 5) Aggregate Outputs: Aggregate the outputs from the simulations to generate a probability distribution for the output variable(s). 6) Analyze Results: Analyze the probability distribution to determine the likelihood of different outcomes or scenarios. 7) Sensitivity Analysis: Conduct sensitivity analysis to determine which input variables have the greatest impact on the output variable(s) and identify potential sources of

uncertainty. 8) Validation: Validate the model or simulation by comparing the output against historical data or known outcomes, and adjust the model as necessary.

Overall, the Monte Carlo workflow is a powerful method for generating probabilistic forecasts and analyzing complex systems. By using random sampling techniques to simulate a large number of possible outcomes, this approach allows decision-makers to better understand the range of possible outcomes and make more informed decisions in the face of uncertainty.

5.3 Simulation Results and Discussion

The simulation results presented in this study can be divided into two parts. The first part involves the generation of probabilistic PV power using Monte Carlo simulation techniques, which is subsequently used to establish the lower and upper bounds of the probability distribution, as depicted in Figures 5.3, 5.4, and 5.5. The second part entails the probabilistic forecasting results obtained from six case studies, as outlined in Table 5.1, with the best-case scenario presented in Figure 5.6. The generative probabilistic PV power results from the Monte Carlo simulation serve as a crucial input in defining the lower and upper bounds of the probability distribution. This step is instrumental in establishing the range of potential outcomes and enables decision-makers to better understand the likelihood of different scenarios. The probabilistic forecasting results provide further insights into the range of possible outcomes and allow decision-makers to optimize their use of RES

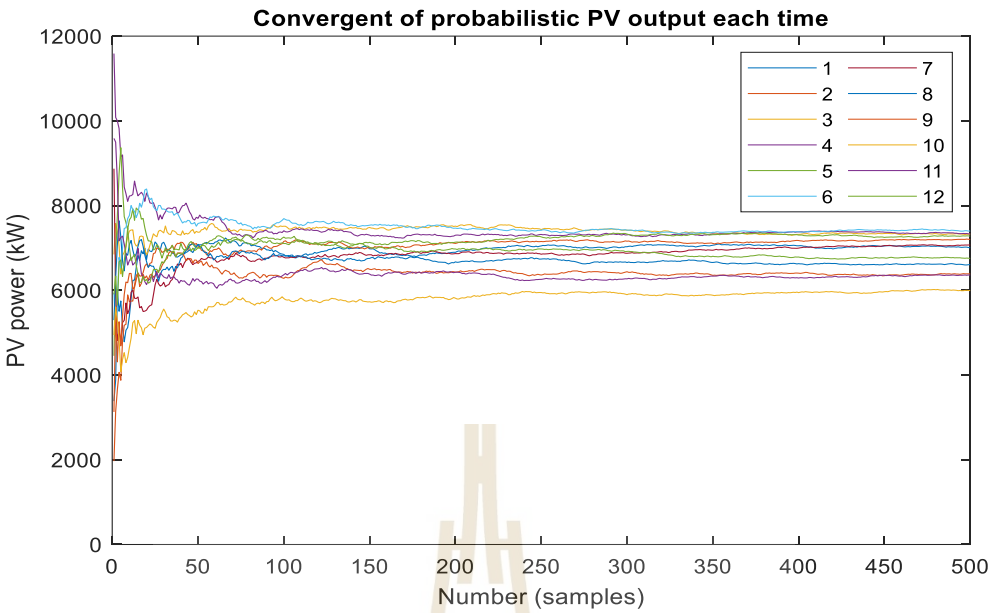


Figure 5.4 Convergent of mean of probabilistic PV output each hr from monte carlo simulation (1 to 12 refer to 7 A.M. to 18 P.M.)

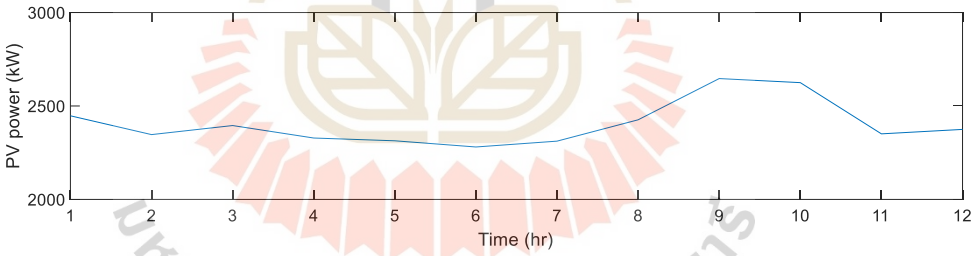


Figure 5.5 SD of probabilistic PV output each hr from monte carlo simulation

Upon analyzing the dataset, the researchers determined the appropriate lower and upper bounds for the probabilistic PV power forecast. The lower bound was selected as the point forecast value, since the proposed point forecasting was found to be not over the actual value. The upper bound was determined to be half of the standard deviation of the power output in the training set, based on its comprehensiveness to the test set. The simulation results, which validate the chosen

lower and upper bounds, are presented in Figure 5.6, while the corresponding probabilistic forecasting results for the six case studies are provided in Table 5.1.

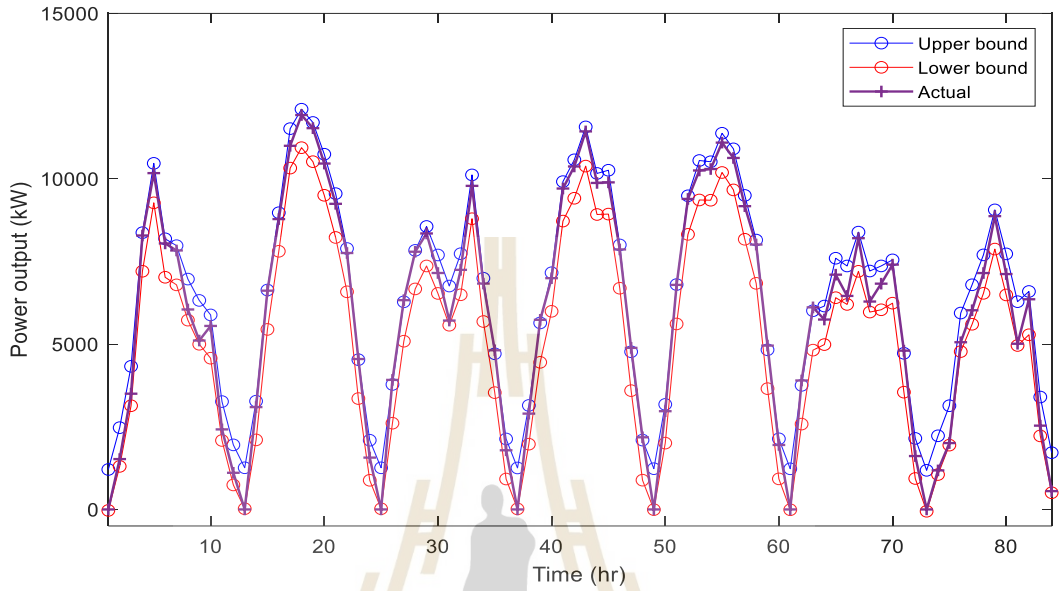


Figure 5.6 Probabilistic forecasting results of Monthly case (best PINAW)

Table 5.1 Probabilistic forecasting results

Case	PICP	PINAW
1 Using all training set	1	0.1010
2 Seasonal (Oct - Dec)	1	0.1008
3 Monthly (Dec)	1	0.1002
5 2 weeks	1	0.1004
6 Previous 7 days	1	0.1018

Note: the performance of each model is not the same when monte carlo was re-calculated

Simulation results show that the selected lower bound and upper bound can be comprehensively used to forecast the interval of the study. As shown in Table 5.1, PICP means how this forecasting model comprehensive the actual value is at 1 (max value). PINAW illustrates the proportion of the width of the average prediction range

per actual value range (max-min) that mean if PINAW is higher, the prediction is wider. PINAW I this case is 10.02 %. As mentioned, we can conclude that this probabilistic model is cover al of the actual value and the prediction range is not over size

5.4 Conclusion

The present chapter introduced a alternative method for enhancing the reliability of forecasting results through the utilization of probabilistic computation in conjunction with a proposed point forecasting technique, namely ML-FNN (3 HIDDEN LAYER). The simulation results demonstrated the effectiveness of this approach in providing a reliable and accurate forecast range for the dataset under consideration, thereby boosting decision-makers' confidence in the reliability of the forecast.

5.5 References

Ahmed, R., Sreeram, V., Mishra, Y., & Arif, M. D. (2020). A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization. *Renewable and Sustainable Energy Reviews,* 124, 109792. doi:https://doi.org/10.1016/j.rser.2020.109792

Bouzerdoum, M., Mellit, A., & Massi Pavan, A. (2013). A hybrid model (SARIMA–SVM) for short-term power forecasting of a small-scale grid-connected photovoltaic plant. *Solar Energy,* 98, 226-235. doi:https://doi.org/10.1016/j.solener.2013.10.002

El-Baz, W., Tzscheutschler, P., & Wagner, U. (2018). Day-ahead probabilistic PV generation forecast for buildings energy management systems. *Solar Energy,* 171, 478-490. doi:https://doi.org/10.1016/j.solener.2018.06.100

Gao, M., Li, J., Hong, F., & Long, D. (2019). Day-ahead power forecasting in a large-scale photovoltaic plant based on weather classification using LSTM. *Energy,* 187, 115838. doi:https://doi.org/10.1016/j.energy.2019.07.168

Li, K., Wang, R., Lei, H., Zhang, T., Liu, Y., & Zheng, X. (2018). Interval prediction of solar power using an Improved Bootstrap method. *Solar Energy,* 159, 97-112. doi:https://doi.org/10.1016/j.solener.2017.10.051

Ni, Q., Zhuang, S., Sheng, H., Kang, G., & Xiao, J. (2017). An ensemble prediction intervals approach for short-term PV power forecasting. *Solar Energy,* 155, 1072-1083. doi:https://doi.org/10.1016/j.solener.2017.07.052

Oneto, L., Laureri, F., Robba, M., Delfino, F., & Anguita, D. (2018). Data-Driven Photovoltaic Power Production Nowcasting and Forecasting for Polygeneration Microgrids. *IEEE Systems Journal,* 12(3), 2842-2853. doi:10.1109/JSYST.2017.2688359

Rahab, H., Zitouni, A., & Djoudi, M. (2018, 2018//). *SIAAC: Sentiment Polarity Identification on Arabic Algerian Newspaper Comments.* Paper presented at the Applied Computational Intelligence and Mathematical Methods, Cham.

Rauf, H. T., Shoaib, U., Lali, M. I., Alhaisoni, M., Irfan, M. N., & Khan, M. A. (2020). Particle Swarm Optimization With Probability Sequence for Global Optimization. *IEEE Access,* 8, 110535-110549. doi:10.1109/ACCESS.2020.3002725

Raza, M. Q., Mithulananthan, N., & Summerfield, A. (2018). Solar output power forecast using an ensemble framework with neural predictors and Bayesian adaptive combination. *Solar Energy,* 166, 226-241. doi:https://doi.org/10.1016/j.solener.2018.03.066

Sayed, A. A., Elgeldawi, E., Zaki, A. M., & Galal, A. R. (2020, 8-9 Feb. 2020). *Sentiment Analysis for Arabic Reviews using Machine Learning Classification Algorithms.* Paper presented at the 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE).

VanDeventer, W., Jamei, E., Thirunavukkarasu, G. S., Seyedmahmoudian, M., Soon, T. K., Horan, B., . . . Stojcevski, A. (2019). Short-term PV power forecasting using hybrid GASVM technique. *Renewable Energy,* 140, 367-379. doi:https://doi.org/10.1016/j.renene.2019.02.087

Wang, F., Xuan, Z., Zhen, Z., Li, K., Wang, T., & Shi, M. (2020). A day-ahead PV power forecasting method based on LSTM-RNN model and time correlation modification under partial daily pattern prediction framework. *Energy Conversion and Management,* 212, 112766. doi:https://doi.org/10.1016/j.enconman.2020.112766

Zhang, X., Wang, R., Liao, T., Zhang, T., & Zha, Y. (2015, 7-10 Dec. 2015). *Short-Term Forecasting of Wind Power Generation Based on the Similar Day and Elman*

Neural Network. Paper presented at the 2015 IEEE Symposium Series on Computational Intelligence.

CHAPTER 6
Conclusion

6.1 Conclusion

The thesis aims to develop the performance and reliability of forecasting results using SOM and probabilistic forecasting. This thesis divides into 3 parts: 1) comparative study to find the appropriate forecasting model and the way to tune the hyperparameter, the simulation results show that ML-FNN (3 hidden layers) is the best model when considering accuracy, training time, and reaching the global minimum error. 2) Improving the performance of the forecasting model using SOM, the simulation results show that SOM can be used to improve the performance of data-driven-based forecasting methods including ML-FNN-SOM (3 hidden layers) and LSTM. And 3) using the probabilistic computation to firm the forecasting range of the point forecasting model, the simulation results show that this proposed method can cover 100 percent of the prediction interval of this dataset. The limitation of this study is considered only hourly data

6.2 Suggestions

There is a suggestion for hyperparameter tuning, using SOM, and probabilistic computation.

6.2.1 Suggestions for tuning hyperparameter of ML-FNN

Tuning hyperparameters of a neural network is an essential step in achieving optimal performance. Here are some suggestions to help you tune your hyperparameters effectively:

Start with a baseline model: Before beginning any hyperparameter tuning, start by building a baseline model. This model should be relatively simple and

use default hyperparameters. It provides a starting point for comparison, and you can use it to establish a baseline performance metric.

Identify the Key Hyperparameters: Hyperparameters are model parameters that are not learned during training. These include learning rate, batch size, activation function, number of hidden layers, and number of neurons in each layer. Identify which hyperparameters are most critical to the performance of your model.

Choose a search strategy: There are several strategies for hyperparameter tuning, including grid search, random search, and Bayesian optimization. Grid search involves trying all possible combinations of hyperparameters, whereas random search randomly selects a subset of hyperparameters to try. Bayesian optimization uses past results to determine the most promising hyperparameters to try next. Choose a search strategy that suits your problem asnd computational resources.

Define the search space: The search space is the range of hyperparameters you want to explore. Define the search space for each hyperparameter you want to tune. For example, you might define a learning rate search space of [0.001, 0.01, 0.1].

Train and evaluate models: train and evaluate the model using the search space you defined. For each combination of hyperparameters, train the model and evaluate its performance. Repeat this process for all combinations of hyperparameters in your search space.

Analyze results: analyze the results of your hyperparameter search. Identify the hyperparameters that result in the best performance. Visualize the results to identify any trends or relationships between hyperparameters and performance.

Refine the search space: refine the search space based on the results of your analysis. If a particular hyperparameter is not affecting the performance of the

model, remove it from the search space. If a particular hyperparameter has a significant impact on performance, expand the search space to include more values.

Repeat the process: repeat the hyperparameter tuning process until you have found the optimal set of hyperparameters for your model. It may take several rounds of tuning to achieve the best performance.

Test on unseen data: finally, evaluate the performance of your model on unseen data to ensure that it generalizes well.

By following these suggestions, you should be able to tune the hyperparameters of your neural network effectively and achieve optimal performance.

6.2.2 Suggestions for using self-organizing maps (SOMs)

Data Visualization: SOMs can be used for visualizing high-dimensional data. By mapping the data onto a 2D or 3D space, patterns and relationships between data points can be easily observed.

Clustering: SOMs can be used for clustering data. Data points that are mapped onto the same neuron on the SOM are considered to be similar and can be grouped.

Anomaly Detection: SOMs can be used for anomaly detection. Data points that are mapped onto neurons that are far away from the others may be considered outliers.

Feature Selection: SOMs can be used for feature selection. By analyzing which features contribute the most to the formation of clusters on the SOM, less important features can be removed, reducing the dimensionality of the data.

Prediction: SOMs can be used for prediction. Once the SOM has been trained on a set of data, it can be used to predict the mapping of new, unseen data points onto the SOM, allowing for classification or regression.

Optimization: SOMs can be used for optimization. By creating a SOM of a set of design parameters and their corresponding outcomes, it is possible to find the optimal set of parameters that lead to the desired outcome.

Image Processing: SOMs can be used for image processing. By mapping the pixels of an image onto the SOM, the SOM can learn to recognize patterns in the image, allowing for image segmentation, object recognition, and image compression.

These are just a few examples of how SOMs can be used. The applications of SOMs are wide-ranging, and they can be used in any field where high-dimensional data needs to be analyzed.

6.2.3 Suggestion for PV power generation probabilistic forecasting

Probabilistic forecasting is a critical component of PV power generation forecasting, as it provides information about the uncertainty and risk associated with the predicted output. Here are some suggestions to help you develop an effective probabilistic forecast for PV power generation:

Data preprocessing: ensure that the input data is preprocessed and cleaned before feeding it into the forecasting model. This includes outlier removal, normalization, and feature engineering to extract useful information.

Select appropriate Models: Several models can be used for probabilistic forecasting, including Gaussian Process regression, quantile regression, bayesian neural networks, and ensemble methods. choose the model that is most suitable for your problem and data.

Train the model: train the model using historical PV power generation data, and use cross-validation techniques to evaluate the model's performance. Use appropriate loss functions such as mean absolute error or quantile loss function to train the model to predict the desired quantile.

Use ensemble models: ensemble models such as Bayesian Model Averaging, Weighted Average, or Stacking can help combine the strength of multiple models to produce more accurate probabilistic forecasts.

Incorporate weather forecast: weather plays a critical role in the generation of PV power. Incorporate weather forecast data, such as solar irradiance and temperature, into the model to improve the accuracy of the forecast.

Model evaluation: evaluate the model's performance using appropriate metrics such as Continuous Ranked Probability Score (CRPS) or Pinball loss to assess the quality of the probabilistic forecast.

Calibration: ensure that the probabilistic forecast is well calibrated by comparing the predicted probabilities to the actual outcomes. Calibration can be achieved using post-processing techniques like Platt scaling or Beta Regression.

Update the model: update the model regularly as new data becomes available to ensure the forecast remains accurate and up to date.

By following these suggestions, you can develop an effective probabilistic forecast for PV power generation that provides valuable information about the uncertainty and risk associated with the predicted output.

APPENDIX A

TEST CASE FOR HYPERPARAMETER TUNING

A.1 ML-FNN

Table A.1. Hyperparameter for case 1 (at 1000 iteration or maximum performance)

No.	Layer/ node	Activation function	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1.1	1 layer/layer 1: 10 nodes	tansig	8.827	308.454	164.410	5.413
1.2	1 layer/layer 1: 20 nodes	tansig	9.006	307.988	158.655	11.583
1.3	1 layer/layer 1: 50 nodes	tansig	8.998	306.941	161.314	25.132
1.4	1 layer/layer 1: 10 nodes	logsig	9.116	311.396	171.566	5.291
1.5	1 layer/layer 1: 20 nodes	logsig	8.663	301.181	156.489	11.331
1.6	1 layer/layer 1: 50 nodes	logsig	9.114	300.041	156.585	25.926
1.7	1 layer/layer 1: 10 nodes	poslin	8.604	315.388	158.569	0.317
1.8	1 layer/layer 1: 20 nodes	poslin	8.839	316.673	167.373	1.307
1.9	1 layer/layer 1: 50 nodes	poslin	8.638	314.238	160.388	21.441
2.1	2 layer/layer 1: 10 nodes layer 2: 10 nodes	tansig	9.349	303.037	158.554	18.177
2.2	2 layer/layer 1: 20 nodes	tansig	8.738	295.470	152.800	72.047

No.	Layer/ node	Activation function	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
	layer 2: 20 nodes					
2.3	2 layer/layer 1: 50 nodes layer 2: 50 nodes	tansig	8.915	300.648	156.385	2,570.800
2.4	2 layer/layer 1: 10 nodes layer 2: 20 nodes	tansig	8.568	298.384	156.850	35.547
2.5	2 layer/layer 1: 10 nodes layer 2: 50 nodes	tansig	8.568	298.384	156.850	33.68
2.6	2 layer/layer 1: 20 nodes layer 2: 10 nodes	tansig	8.792	299.770	159.217	127.344
2.7	2 layer/layer 1: 20 nodes layer 2: 50 nodes	tansig	8.891	299.485	160.940	369.004
2.8	2 layer/layer 1: 50 nodes layer 2: 10 nodes	tansig	8.763	293.94	153.19	115.820
2.9	2 layer/layer 1: 50 nodes layer 2: 20 nodes	tansig	9.018	302.211	156.94	305.480
2.10	2 layer/layer 1: 10 nodes layer 2: 10 nodes	logsig	8.764	297.340	156.32	17.911

No.	Layer/ node	Activation function	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
2.11	2 layer/layer 1: 20 nodes layer 2: 20 nodes	logsig	8.664	295.596	155.695	71.771
2.12	2 layer/layer 1: 50 nodes layer 2: 50 nodes	logsig	9.343	304.309	161.270	2506.400
2.13	2 layer/layer 1: 10 nodes layer 2: 20 nodes	logsig	8.598	294.741	151.720	29.302
2.14	2 layer/layer 1: 10 nodes layer 2: 50 nodes	logsig	9.391	300.848	155.331	107.715
2.15	2 layer/layer 1: 20 nodes layer 2: 10 nodes	logsig	8.987	298.534	156.071	32.412
2.16	2 layer/layer 1: 20 nodes layer 2: 50 nodes	logsig	8.533	297.839	155.567	294.962
2.17	2 layer/layer 1: 50 nodes layer 2: 10 nodes	logsig	9.194	305.095	157.108	129.257
2.18	2 layer/layer 1: 50 nodes layer 2: 20 nodes	logsig	9.018	302.211	156.941	305.480
2.19	2 layer/layer 1: 10 nodes	poslin	9.005	313.663	164.073	2.072

No.	Layer/ node	Activation function	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
	layer 2: 10 nodes					
2.20	2 layer/layer 1: 20 nodes layer 2: 20 nodes	poslin	8.728	320.388	167.217	2.465
2.21	2 layer/layer 1: 50 nodes layer 2: 50 nodes	poslin	8.717	321.62	167.521	774.970
2.22	2 layer/layer 1: 10 nodes layer 2: 20 nodes	poslin	8.731	320.799	164.458	2.965
2.23	2 layer/layer 1: 10 nodes layer 2: 50 nodes	poslin	8.782	324.250	165.510	23.570
2.24	2 layer/layer 1: 20 nodes layer 2: 10 nodes	poslin	8.992	322.971	170.001	1.353
2.25	2 layer/layer 1: 20 nodes layer 2: 50 nodes	poslin	8.697	319.608	163.157	136.343
2.26	2 layer/layer 1: 50 nodes layer 2: 10 nodes	poslin	8.560	311.590	157.260	7.937
2.27	2 layer/layer 1: 50 nodes layer 2: 20 nodes	poslin	9.011	317.9190	168.044	25.967

No.	Layer/ node	Activation function	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
3.1	3 layer/layer 1: 10 nodes layer 2: 10 nodes layer 3: 10 nodes	tansig	9.363	300.937	153.231	31.939
3.2	3 layer/layer 1: 20 nodes layer 2: 20 nodes layer 3: 20 nodes	tansig	9.431	358.750	172.391	193.958
3.3	3 layer/layer 1: 50 nodes layer 2: 50 nodes layer 3: 50 nodes	tansig	9.374	312.871	157.963	10,198.000
3.4	3 layer/layer 1: 10 nodes layer 2: 10 nodes layer 3: 20 nodes	tansig	8.620	309.195	158.454	49.862
3.5	3 layer/layer 1: 10 nodes layer 2: 20 nodes layer 3: 10 nodes	tansig	9.383	332.639	164.741	66.486
3.6	3 layer/layer 1: 10 nodes layer 2: 20 nodes layer 3: 20 nodes	tansig	9.032	300.547	154.746	120.150
3.7	3 layer/layer 1: 20 nodes	tansig	9.406	302.888	158.611	54.985

No.	Layer/ node	Activation function	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
	layer 2: 10 nodes layer 3: 10 nodes					
3.8	3 layer/layer 1: 20 nodes layer 2: 20 nodes layer 3: 10 nodes	tansig	9.176	325.266	160.932	124.592
3.9	3 layer/layer 1: 20 nodes layer 2: 10 nodes layer 3: 20 nodes	tansig	9.196	306.239	163.160	84.842
3.10	3 layer/layer 1: 10 nodes layer 2: 10 nodes layer 3: 10 nodes	logsig	8.857	306.998	161.593	31.153
3.11	3 layer/layer 1: 20 nodes layer 2: 20 nodes layer 3: 20 nodes	logsig	8.916	309.866	157.404	193.980
3.12	3 layer/layer 1: 50 nodes layer 2: 50 nodes layer 3: 50 nodes	logsig	8.962	303.772	158.284	9,751.600
3.13	3 layer/layer 1: 10 nodes layer 2: 10 nodes layer 3: 20 nodes	logsig	9.264	304.938	161.328	48.439

No.	Layer/ node	Activation function	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
3.14	3 layer/layer 1: 10 nodes layer 2: 20 nodes layer 3: 10 nodes	logsig	9.555	311.327	166.509	63.207
3.15	3 layer/layer 1: 10 nodes layer 2: 20 nodes layer 3: 20 nodes	logsig	8.716	310.796	162.895	118.522
3.16	3 layer/layer 1: 20 nodes layer 2: 10 nodes layer 3: 10 nodes	logsig	9.040	305.686	160.838	53.528
3.17	3 layer/layer 1: 20 nodes layer 2: 20 nodes layer 3: 10 nodes	logsig	9.272	302.928	163.158	126.519
3.18	3 layer/layer 1: 20 nodes layer 2: 10 nodes layer 3: 20 nodes	logsig	9.244	306.197	163.340	85.083
3.19	3 layer/layer 1: 10 nodes layer 2: 10 nodes layer 3: 10 nodes	poslin	8.572	319.565	162.838	9.378
3.20	3 layer/layer 1: 20 nodes	poslin	8.504	314.900	159.797	181.924

No.	Layer/ node	Activation function	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
	layer 2: 20 nodes layer 3: 20 nodes					
3.21	3 layer/layer 1: 50 nodes layer 2: 50 nodes layer 3: 50 nodes	poslin	10.115	326.178	186.254	717.26
3.22	3 layer/layer 1: 10 nodes layer 2: 10 nodes layer 3: 20 nodes	poslin	8.681	319.550	164.365	4.564
3.23	3 layer/layer 1: 10 nodes layer 2: 20 nodes layer 3: 10 nodes	poslin	8.711	326.173	166.322	4.617
3.24	3 layer/layer 1: 10 nodes layer 2: 20 nodes layer 3: 20 nodes	poslin	8.745	320.129	167.589	6.069
3.25	3 layer/layer 1: 20 nodes layer 2: 10 nodes layer 3: 10 nodes	poslin	8.565	312.657	158.320	27.654
3.26	3 layer/layer 1: 20 nodes layer 2: 20 nodes layer 3: 10 nodes	poslin	8.509	317.002	162.633	7.634

No.	Layer/ node	Activation function	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
3.27	3 layer/layer 1: 20 nodes layer 2: 10 nodes layer 3: 20 nodes	poslin	8.713	321.253	164.837	3.557

Table A.2 Comparison of activation function with normalization

No.	Layer 1	Layer 2	Layer 3	Norm	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1	**logsig**	**logsig**	**logsig**	yes	8.916	309.866	157.404	253.208
2	logsig	logsig	tansig	yes	8.916	309.866	157.404	253.208
3	logsig	tansig	logsig	yes	9.411	315.340	159.152	217.273
4	logsig	logsig	poslin	yes	9.333	317.794	170.151	45.891
5	logsig	poslin	logsig	yes	8.947	323.675	170.460	116.523
6	logsig	tansig	tansig	yes	8.812	297.740	155.471	259.814
7	logsig	poslin	poslin	yes	8.402	311.954	159.126	14.705
8	logsig	tansig	poslin	yes	10.590	235.430	174.892	12.594
9	logsig	poslin	tansig	yes	8.881	321.409	166.080	14.292
10	**poslin**	**poslin**	**poslin**	yes	8.504	314.900	159.798	182.084
11	poslin	poslin	logsig	yes	8.771	318.652	165.939	20.258
12	poslin	logsig	poslin	yes	8.740	309.095	159.658	18.651
13	poslin	poslin	tansig	yes	9.986	324.266	176.096	18.204
14	poslin	tansig	poslin	yes	8.690	314.263	160.972	33.714
15	tansig	poslin	tansig	yes	8.687	317.713	164.333	5.575
16	poslin	tansig	tansig	yes	8.977	313.266	163.037	48.419
17	poslin	logsig	tansig	yes	8.491	310.309	158.237	14.470
18	poslin	tansig	logsig	yes	8.969	322.249	172.325	32.454
19	**tansig**	**tansig**	**tansig**	yes	9.431	358.750	172.391	192.778
20	tansig	tansig	logsig	yes	9.999	387.018	186.085	197.925
21	tansig	logsig	tansig	yes	8.736	300.372	154.916	192.239

22	tansig	tansig	poslin	yes	9.754	319.155	178.164	33.591
23	tansig	poslin	tansig	yes	8.675	310.752	158.637	37.944
24	tansig	logsig	logsig	yes	9.402	306.910	156.977	190.803
25	tansig	poslin	poslin	yes	9.346	322.221	173.886	176.386
26	tansig	poslin	logsig	yes	8.774	315.054	163.050	186.226
27	tansig	logsig	poslin	yes	8.705	313.445	160.146	46.766

Note: Using model 3.20 in table A.1

Table A.3 Comparison of activation function without normalization

No.	Layer 1	Layer 2	Layer 3	Norm	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1	logsig	poslin	poslin	no	8.402	311.954	159.126	10.598
2	poslin	logsig	tansig	no	8.491	310.309	158.237	14.590
3	poslin	poslin	poslin	no	8.504	314.900	159.798	179.017
4	tansig	poslin	tansig	no	8.675	310.752	158.637	37.396
5	tansig	logsig	poslin	no	8.705	313.445	160.146	45.387

Note: Using top 5 models in table A.2

A.2 NARX (Manual, See more details in Appendix C)

Table A.4 hyperparameter tuning for NARX

No.	Layer/ node	Activation function	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1	1 layer/layer 1: 10nodes	poslin	59.190	1,578.200	918.342	35.732
2	1 layer/layer 1: 20nodes	poslin	59.894	1,584.200	920.452	426.591
3	1 layer/layer 1: 50nodes	poslin	56.319	1,554.700	904.376	4,918.500
4	2 layer/layer 1: 10nodes layer 2: 10nodes	poslin	60.038	1,654.900	928.406	20.675
5	2 layer/layer 1: 20nodes layer 2: 20nodes	poslin	64.320	1,637.900	934.445	346.316
6	2 layer/layer 1: 50nodes layer 2: 50nodes	poslin	56.231	1,573.900	898.595	13,753.000
7	3 layer/layer 1: 10nodes layer 2: 10 nodes layer 3: 10 nodes	poslin	58.160	1,647.000	912.574	21.936
8	3 layer/layer 1: 20nodes layer 2: 20 nodes layer 3: 20 nodes	poslin	56.797	1,609.500	904.325	1,900.100
9	3 layer/layer 1: 50nodes layer 2: 50 nodes layer 3: 50 nodes	poslin	55.437	1,509.200	808.321	36,200.000

A.3 LSTM (Manual, See more details in Appendix C)

Table A.5 hyperparameter tuning for LSTM

No.	Layer/ node	MAPE (%)	RMSE (kW)	MAE (kW)	Time (Sec)
1	1 layer/layer 1: 10nodes	14.808	462.120	249.644	3,571.000
2	1 layer/layer 1: 20nodes	14.398	450.952	235.199	3,596.200
3	1 layer/layer 1: 50nodes	12.167	369.067	204.658	3,873.700
4	2 layer/layer 1: 10nodes layer 2: 10nodes	13.464	390.240	241.690	2,672.400
5	2 layer/layer 1: 20nodes layer 2: 20 nodes	15.922	373.357	222.856	2,935.300
6	2 layer/layer 1: 50nodes layer 2: 50 nodes	18.018	430.047	254.411	2,841.200
7	3 layer/layer 1: 10nodes layer 2: 10 nodes layer 3: 10 nodes	16.037	392.5301	229.179	5,270.000
8	3 layer/layer 1: 20nodes layer 2: 20 nodes layer 3: 20 nodes	12.968	339.225	17.3623	5,713.900
9	3 layer/layer 1: 50nodes layer 2: 50 nodes layer 3: 50 nodes	15.972	361.602	223.119	5,493.500

A.4 Others Optimize-based ML

To achieve the best performance of ML based-forecasting model, hyperparameter optimization technique is necessary. The hyperparameter were optimized by Bayesian Optimization that used MSE as minimize function results are shown in figure C.1- C.12

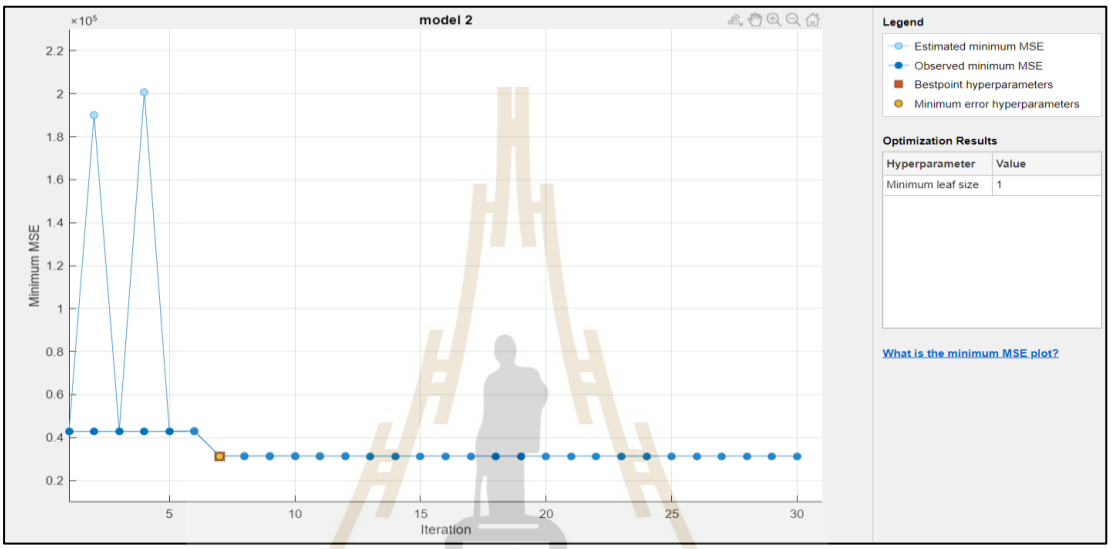


Figure A.1 Optimized Ensemble of trees for case 1

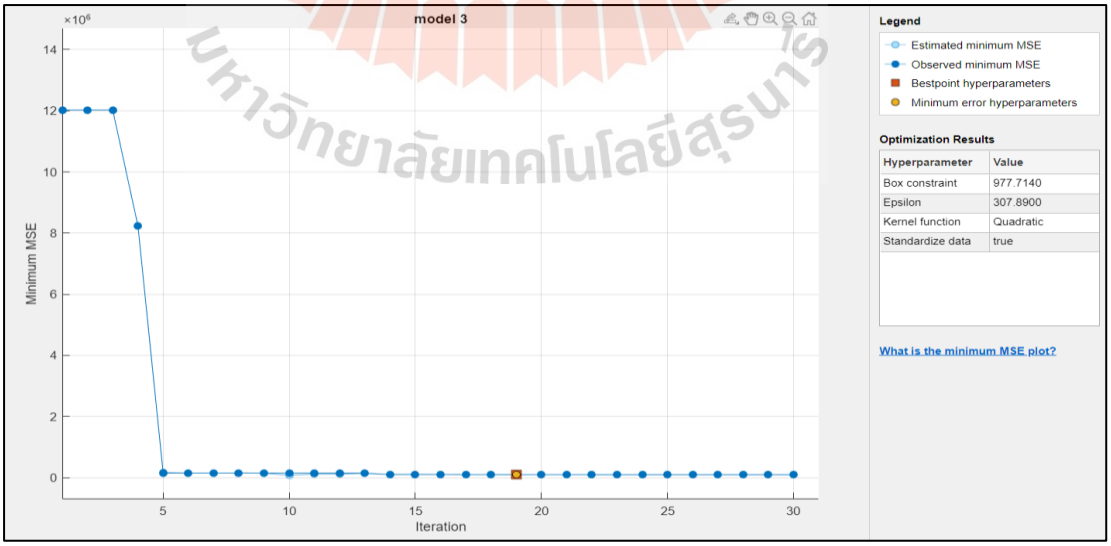


Figure A.2 Optimized SVR for case 1

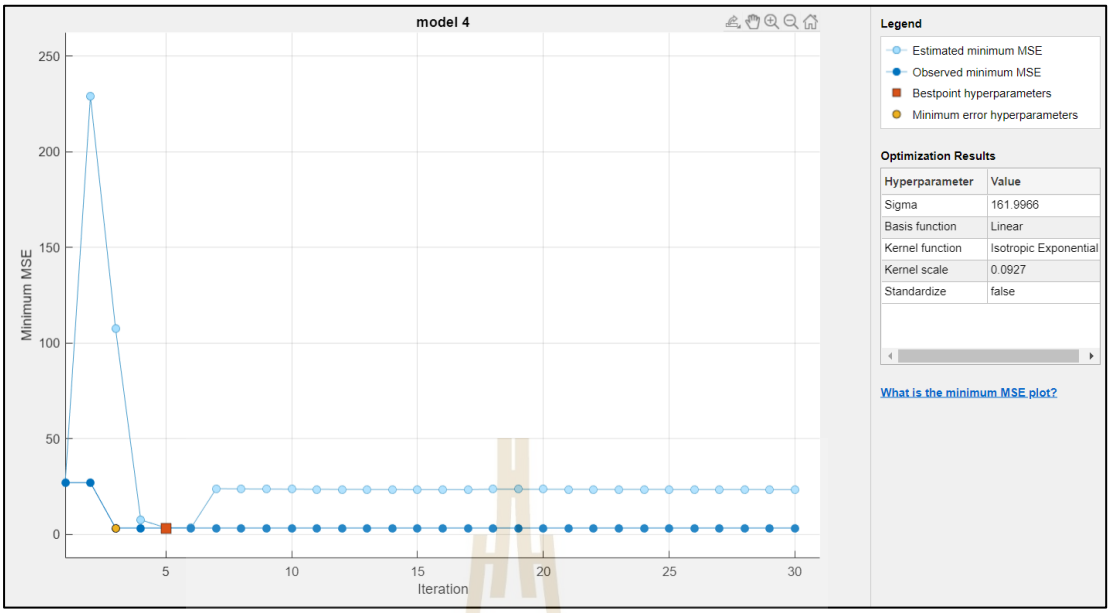


Figure A.3 Optimized GPR for case 1

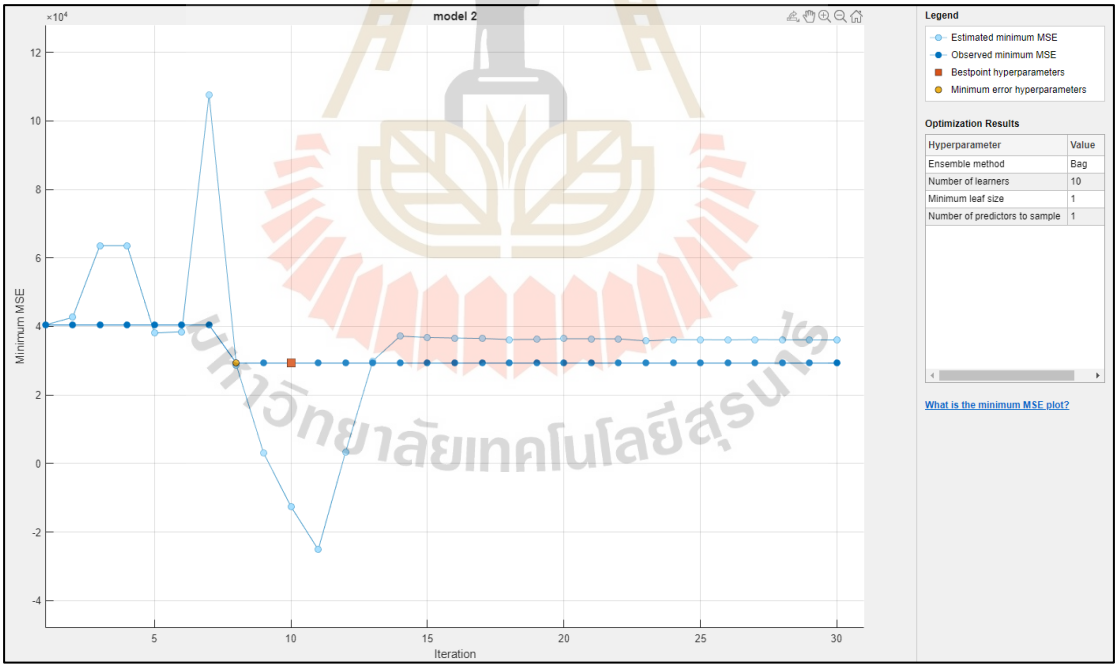


Figure A.4 Optimized Ensemble of trees for case 2

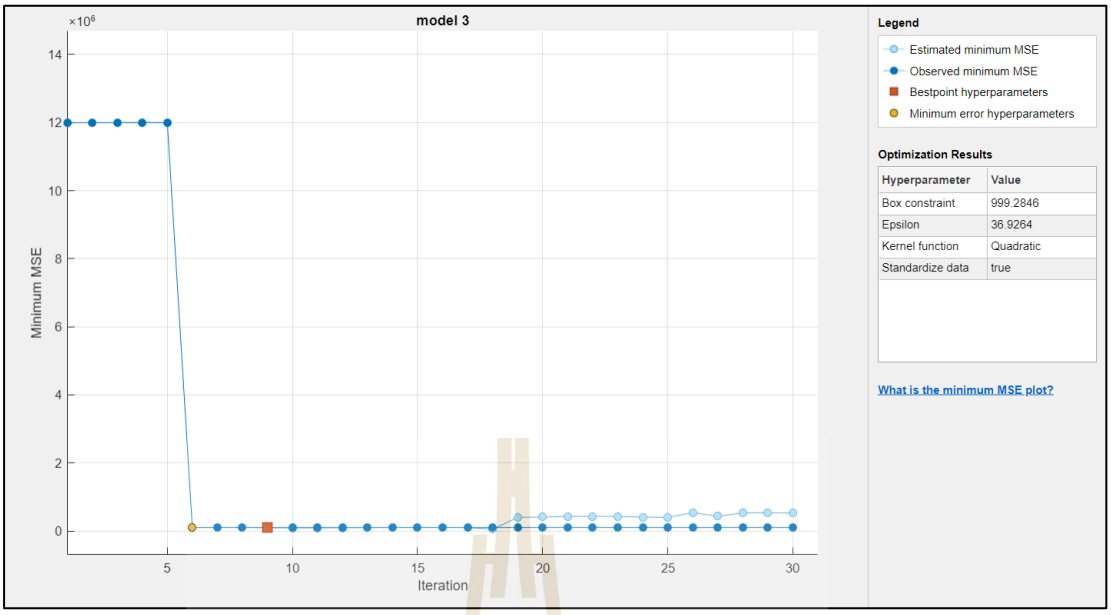


Figure A.5 Optimized SVR for case 2

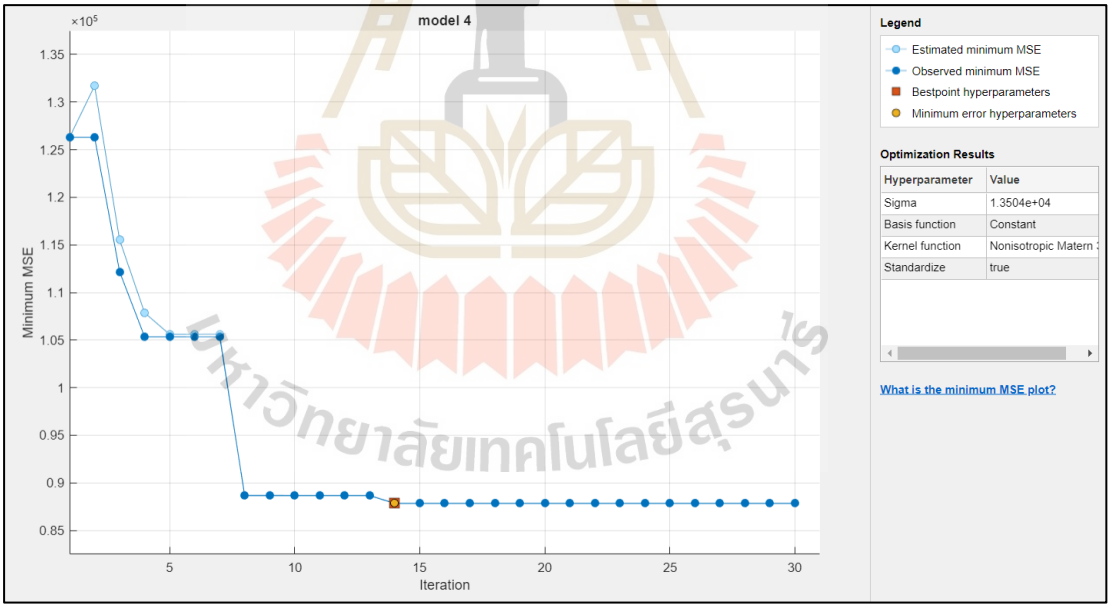


Figure A.6 Optimized GPR for case 2

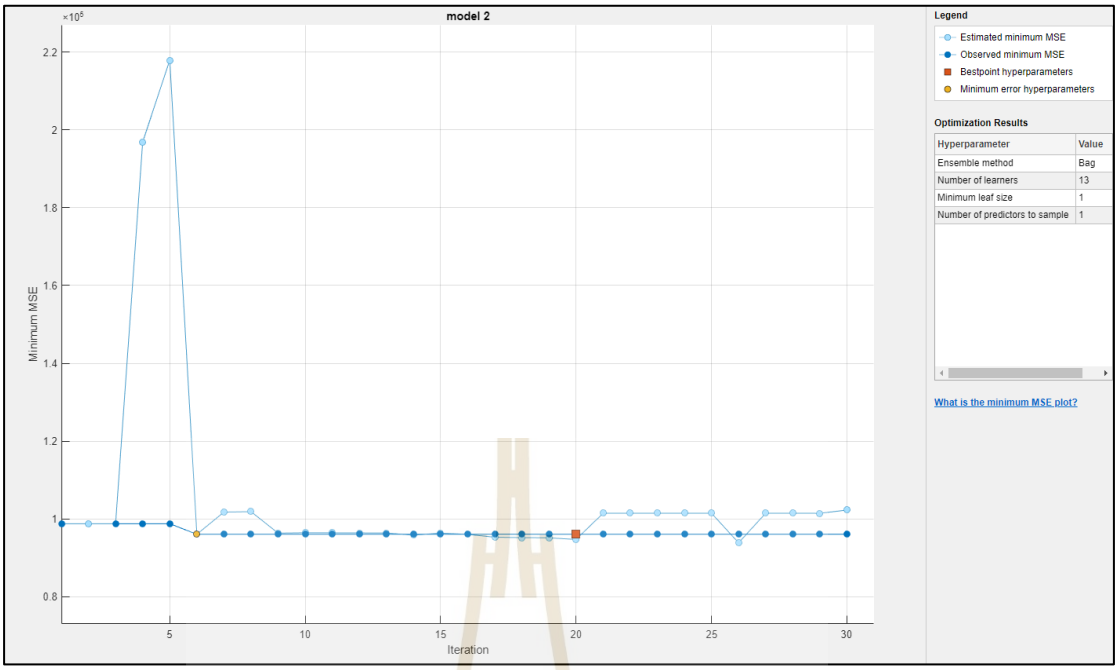


Figure A.7 Optimized Ensemble of trees for case 3

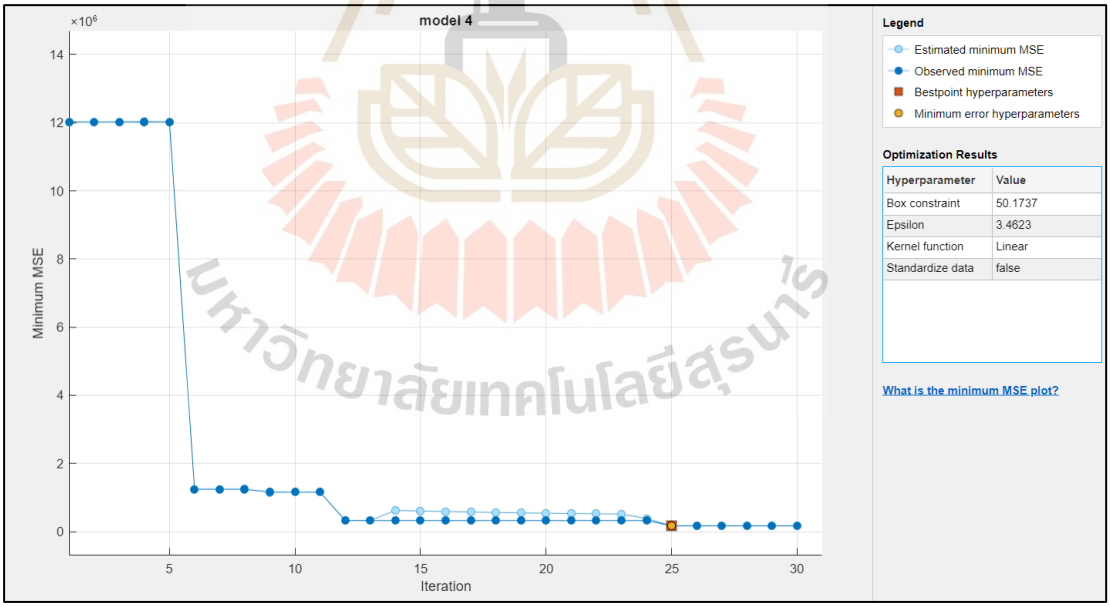


Figure A.8 Optimized SVR for case 3

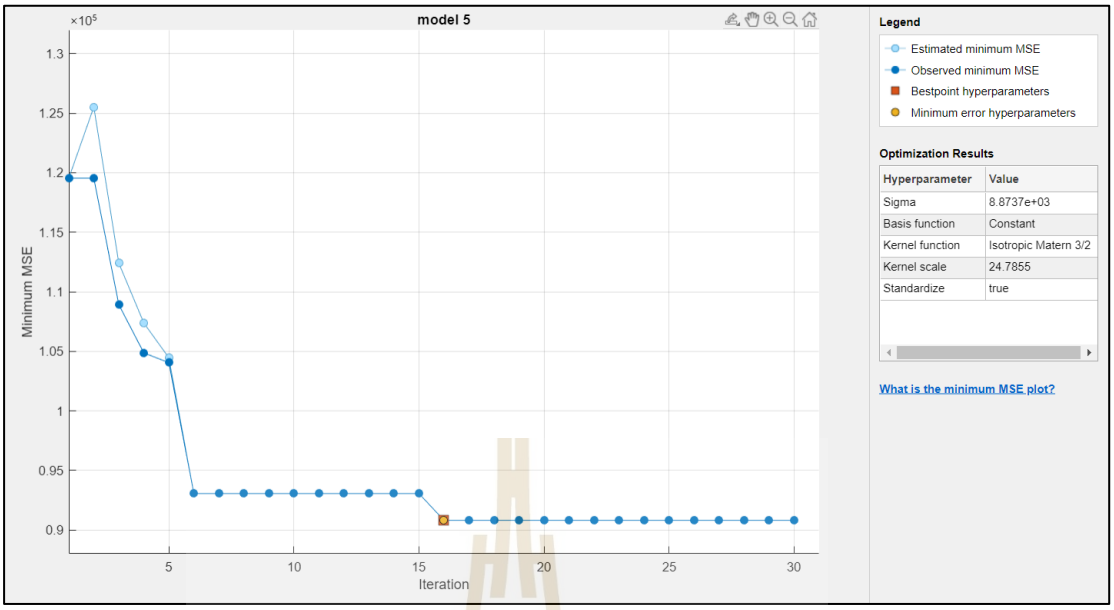


Figure A.9 Optimized GPR for case 3

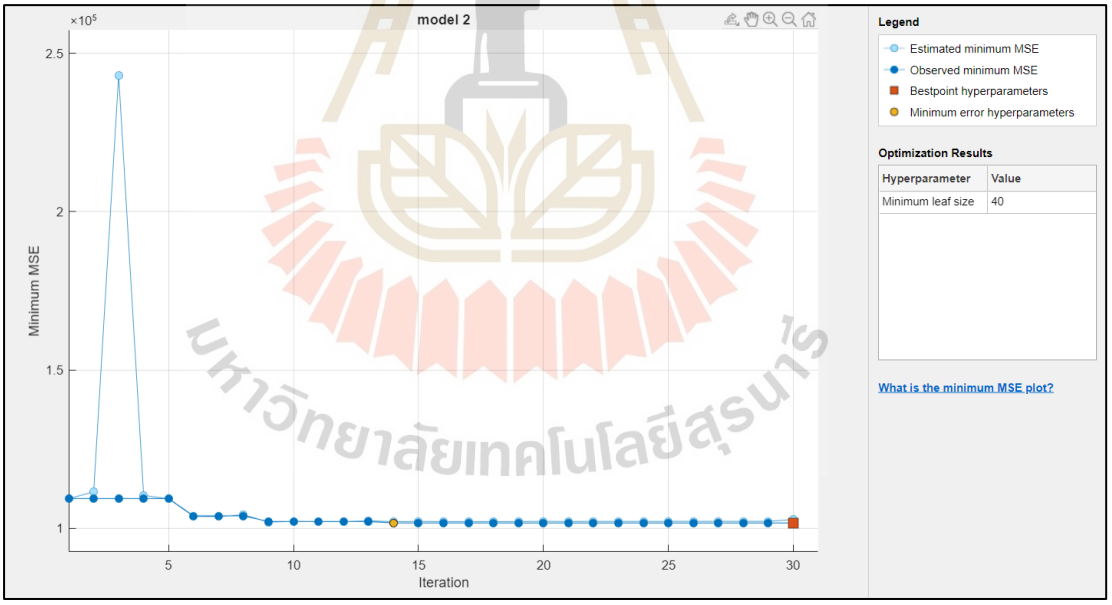


Figure A.10 Optimized Ensemble of trees for case 4

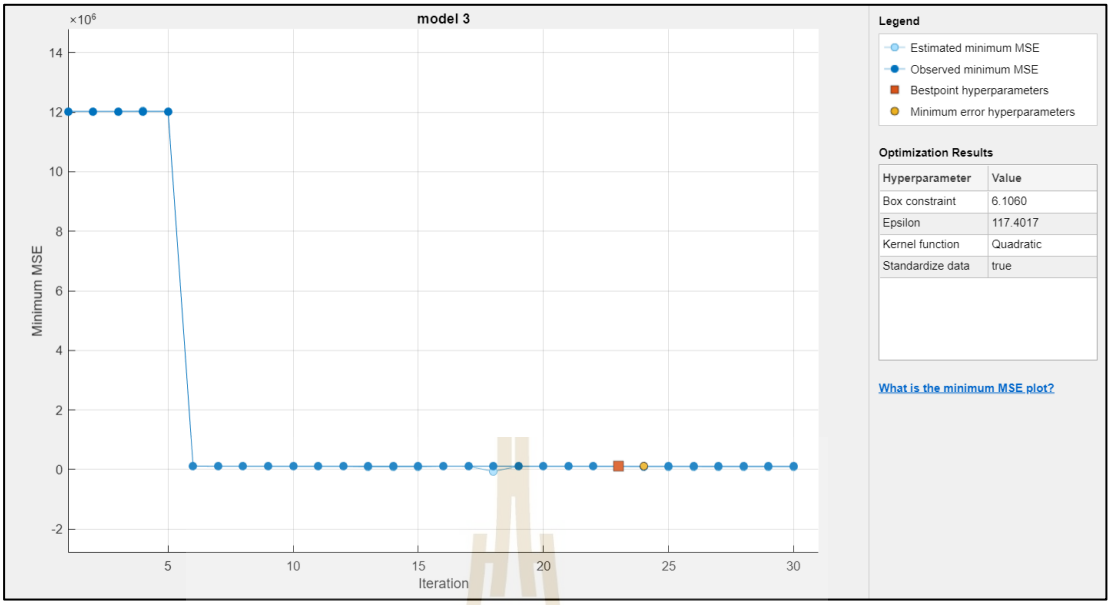


Figure A.11 Optimized SVR for case 4

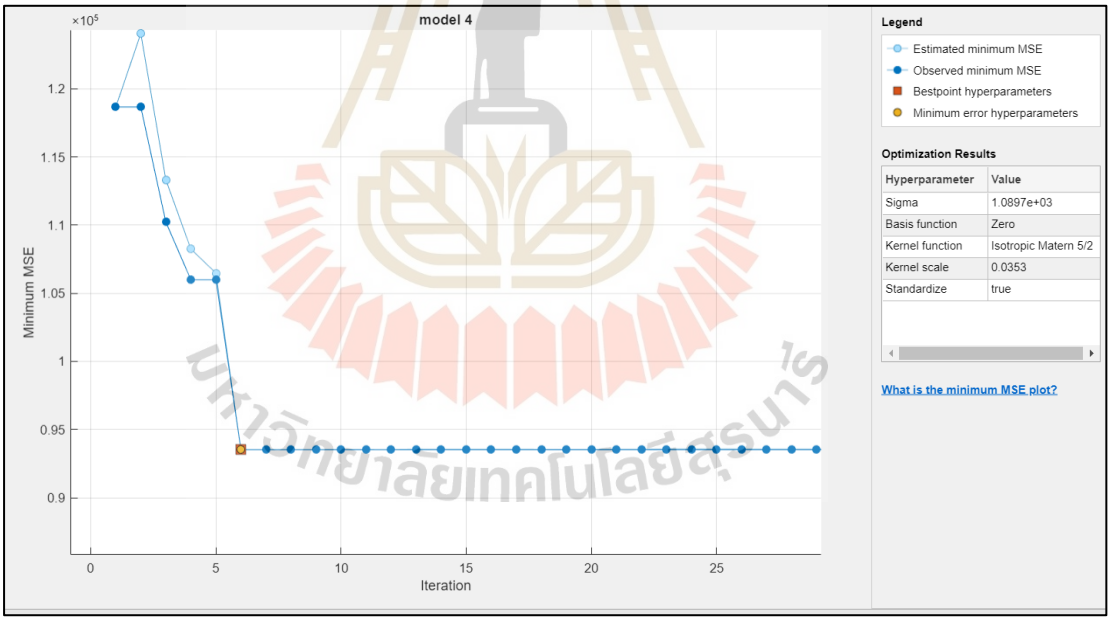
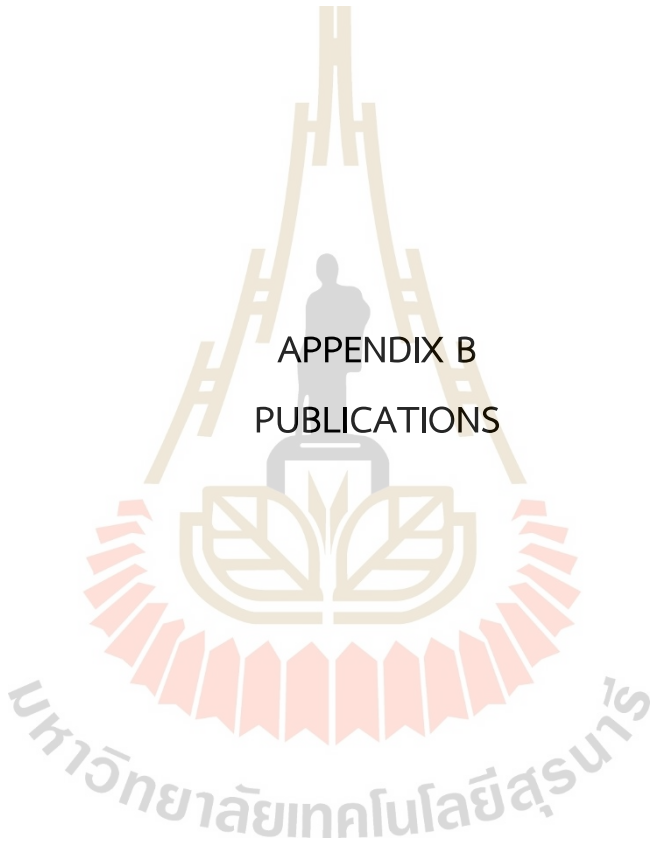


Figure A.12 Optimized GPR for case 4

APPENDIX B

PUBLICATIONS



List of Publication

INTERNATIONAL CONFERENCE PAPER

N. Junhuathon and K. Chayakulkheeree, **"Comparative Study of Short-Term Photovoltaic Power Generation Forecasting Methods,"** 2021 International Conference on Power, Energy and Innovations (ICPEI), Nakhon Ratchasima, Thailand, 2021, pp. 159-162, doi: 10.1109/ICPEI52436.2021.9690695.

INTERNATIONAL JOURNAL PAPER

Nitikorn Junhuathon and Keerati Chayakulkheeree , **"Deep-learning-based short-term photovoltaic power generation forecasting using improved self-organization map neural network"**, Journal of Renewable and Sustainable Energy 14, 043702 (2022) https://doi.org/10.1063/5.0091454

2021 International Conference on Power, Energy and Innovations (ICPEI 2021)
October 20-22, 2021, Nakhon Ratchasima, THAILAND

Comparative Study of Short-Term Photovoltaic Power Generation Forecasting Methods

Nitikorn Junhuathon:
School of electrical engineering
Institute of Engineering
Suranaree University of Technology
Nakhon Ratchasima, Thailand
nitikorn_j@rmutt.ac.th

Keerati Chayakulkheeree
School of electrical engineering
Institute of Engineering
Suranaree University of Technology
Nakhon Ratchasima, Thailand
keerati.ch@sut.ac.th

***Abstract*— An accurate forecasting scheme is essential to optimize efficiency when planning a distribution grid with a photovoltaic system. This article proposed a comparative study of the efficiency and advantages between supervised photovoltaic power forecasting methods that can be appropriately applied to various systems. For comparative study, Feedforward Neural Network (FNN) was compared with Long Short-Term Memory (LSTM) and Nonlinear Autoregressive with External Input (NARX) that use the previous output as input via MATLAB program. The forecasting dataset consists of time, cell temperature, irradiance, and power output from the PV system in 8,760 samples (for each hour). The dataset was divided into three parts following: (1) training set is 70%, (2) validating set 20% and (3) testing 10 %. The simulation results show that, compared to the three methods, LSTM had the highest accuracy, followed by FNN and NARX, respectively. However, NARX is closer to the peak at high temperatures and irradiance. For data set analysis, the difference in power output ranges between the training and test sets affects forecasting accuracy.**

Keywords—PV power generation forecasting, short-term forecasting, LSTM, FNN, NARX

I. Introduction

Providing the resources is essential to balance the increasingly powerful of inherently variable and unpredictable generation from solar and wind that might be prohibitively expensive for the balancing authority responsible for maintaining the balance of load and generation within their region. Solar photovoltaic (PV) generation is one of the most promising renewable energy options for mitigating climate change and enhancing global energy security. Weather factors such as solar irradiance, temperature, humidity, and cloud characteristics influence photovoltaic power generation. These variables contribute to the intermittent and stochastic nature of photovoltaic production. As a result, uncertain power should be accurately forecasted to manage system costs and power balance.

Numerous strategies for forecasting photovoltaic electricity generation have been developed over the years. They can be generally categorized according to their approach to the problem. For instance, indirect approaches anticipate sun irradiance first and then photovoltaic output, but direct methods forecast photovoltaic output directly. The most frequently used classification scheme for direct methods is machine learning and statistical approaches [1]. PV forecasting was divided into two categories as follows: Short-term forecasting, long-term forecasting. The short-term forecasting of photovoltaic generation spans an hour to 24 hours and is crucial for grid security and operation [1, 2]. Long-term forecasting horizons, on the contrary, range from one month to one year and are used for long-term planning [3, 4].

In the field of short-term forecasting, statistical techniques are viral. They use historical data to train various data-driven methodologies, including time series, regression models, machine learning, and deep learning. Statistical models are more adaptable and simpler to operate than physical models. The most often utilized statistical techniques are Feedforward Neural Networks (FNN) [5], Long Short-Term Memories (LSTM) [1, 2], and in some interesting cases, Nonlinear Auto Regressive Networks with Exogenous inputs (NARX) [6, 7]. This popularity originates from their performance, resulting from the high nonlinearity of the connections between solar photovoltaic output and its associated variables.

According to the research discussed previously, a variety of methods and datasets were used to forecast. As a result, this article presents a comparative study between interesting methods and factors for forecasting solar energy generation. The rest of the article is organized as follows: Section 2 discusses forecasting solar cell power using deep learning. Section 3 contains the case study, Data set, Method parameter, and validation factor. Section 4 contains the simulation and its results, as well as a discussion. The last section is conclusions.

II. PV Forecasting Using Deep Learning

This part introduces a commonly used deep learning technique to forecast power output from the PV system. Hourly weather data and power output data from the past are used as inputs to the deep learning forecast algorithm. Multiple rows of historical hourly data comprise the input data. The input data set has m rows. Thus, the input data for FNN and LSTM are m rows by m (m for weather data). NARX is m+1 for (for weather data and PV output power data), and the output of these

2021 International Conference on Power, Energy and Innovations (ICPEI) | 978-1-6654-0216-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPEI52436.2021.9690695

978-1-6654-0216-3/21/$31.00 ©2021 IEEE

2021 International Conference on Power, Energy and Innovations (ICPEI 2021)
October 20-22, 2021, Nakhon Ratchasima, THAILAND

techniques is the power output from the photovoltaic system.

A. FNN

The following sections outline the critical aspects of implementing FNN. The primary aim is to train the feed-forward neural network to get the optimal weights and biases for improving the network's performance. The weights and biases of the training FNN are adjusted using a technique known as Levenberg-Marquardt backpropagation, an enhanced version of the gradient descent approach [5]. The learning error computation function may also be used as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). In this work, MSE is used and can be calculated as following as equation 1.

The workflows diagram of the hidden node and FNN are shown in Fig. 1.

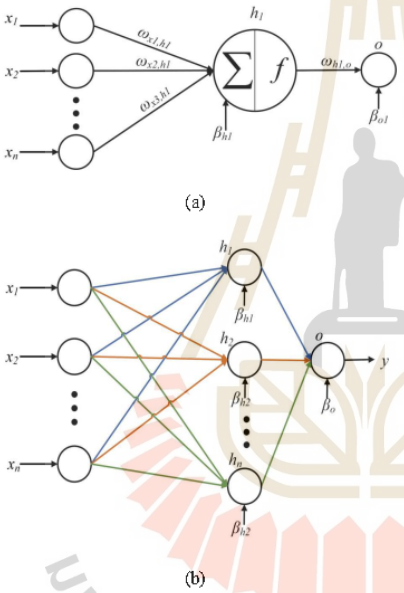


Fig. 1. (a) Structure of hidden node 1
(b) Structure of FNN

$$E_j = \sum_{i=1}^{n} MSE\ (y^j_{forecast\,i}, y^j_{actual\,i}) \quad (1)$$

E_j is training error at training iteration j^{th}, is actual output at sample i^{th}, x_i is input, n is number of training sample, and forecasting output at the training sample j^{th} can be calculated following equation 2.

$$y_j = \sum_{j=1}^{n} \omega_{k,j} f(h_k) + \beta_j \quad (2)$$

Where $\omega_{k,j}$ is weights from hidden node k^{th} to output node at iteration j^{th}, β_j is the bias of output node at j^{th}. $f(h_k)$ is an outcome of hidden h_k. The fitness function is estimated using the training error (E). The fitness function can be calculated as follows.

$$Fitness\,(x) = Minimize\ E(x) \quad (3)$$

B. LSTM

Hochreiter and Schmidhuberin introduced the LSTM in 1997 [5]. A conventional LSTM neural network comprises an input layer, hidden layers, and an output layer. The hidden layer is composed of a collection of memory cells equipped with input and output gates. Then, Gers et al. enhanced the LSTM by introducing a new gate in the memory cell called the forget gate [6]. These gates regulate the flow of information through a memory cell. The central component of a hidden layer is a memory block, which consists of a collection of memory cells that share the same gate units. Figure 2 illustrates the architecture of the LSTM block.

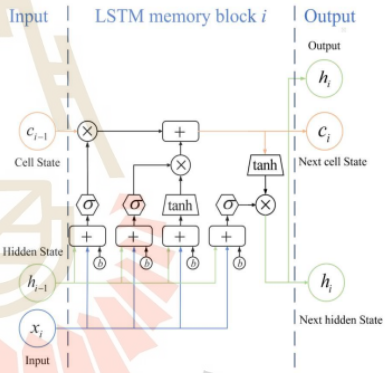


Fig. 2. The structure of the LSTM memory block

Given an input $\{x_i \mid i = 1,2, ..., n\}$ with j^{th} frames, where x_j is the static feature of the i^{th} sample, the standard LSTM is used to learn a sequence of hidden states $\{h_i \mid i = 1, ..., n\}$ to describe the output and dynamic of this input at time step $i^{th}+1$. The standard LSTM mainly consists of an input gate, forget gate, output gate, input modulation gate, and memory cell state. [1] provides additional LSTM equations.

C. NARX

NARX was presented as a more advanced implementation of the Nonlinear Auto-Regressive Neural Network.

2021 International Conference on Power, Energy and Innovations (ICPEI 2021)
October 20-22, 2021, Nakhon Ratchasima, THAILAND

The output regressor of the Nonlinear Auto-Regressive Neural Network was given via a single delayed feedback loop. On the other hand, the NARX Neural Network uses m tapped delay lines in the input and output signals *n* time [7]. The exogenous values are included in NARX's parametric equation as follows.

$$y(t) = f\left[x(t-0),...,x(t-d);y(t-1),...,y(t-d)\right] \quad (4)$$

Where d denote the passed value of output $y(t)$ and another series input $x(t)$ at sample i^{th}. The structure of NARX is shown in Fig. 3.

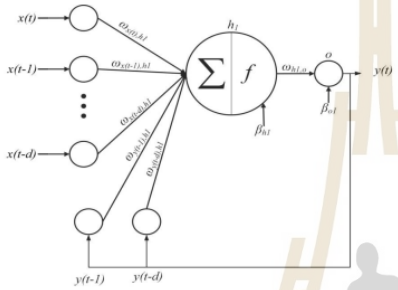


Fig. 3. NARX structure

III. Data Set and Parameter

A. Data set

To compare the forecasting methods, the datasets were used to test each method. The data set includes time, irradiance, and cell temperature as input and power output from the PV system. This research uses a dataset from the northeastern region of Thailand with the installation of a 14 MW solar cell system. The data set with 8,760 samples for each hour was divided into three sets: 70% of the training set, 20% of the validation set, and 10% of the test set. The detail of the datasets is shown in Fig. 4.

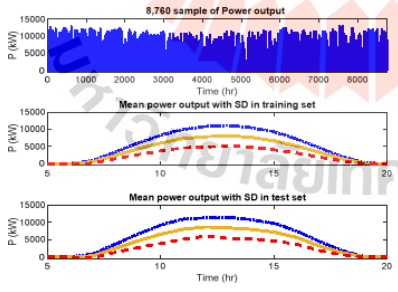


Fig. 4. Dataset for comparative study

B. Method parameter

The data set was trained and tested by FNN, LSTM, and NARX. Before training, the structure parameter of these forecast methods must be set for each scenario following as table I. The hidden node and epoch were defined by several tests to determine the minimal parameter that can reach the optimal network to forecast with the same condition.

TABLE I. PARAMETER FOR EACH NETWORK

Methods	input	Hidden node	epoch
LSTM	3	40	300
FNN	3	40	300
NARX	3+1	40	300

C. Validation Factor

To compare and validate each method, MAE, RMSE, and MAPE, respectively, were used as validation factors. These can be calculated as follow equation 5 to 7.

$$MAE = \frac{\sum_{i=1}^{n} | y_{actual,i} - y_{forecast,i} |}{n} \quad (5)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(y_{actual,i} - y_{forecast,i} \right)^2}{n}} \quad (6)$$

$$MAPE = \sum_{i=1}^{n} \frac{\left| y_{actual,i} - y_{forecast,i} \right|}{y_{real,i}} \times 100\% \quad (7)$$

Where $y_{real,i}$ is the actual power output at sample i^{th}, and n is sample number.

IV. Simulation Results and Discussion

The simulation results in Fig. 5 show the comparison between actual power output and forecast power output 875 samples in the test set. The simulation results show that FNN and LSTM are quite accurate at power output lower than 8,000 kW, and NARX is lower accurate at this range. For over 8,000 kW of power output, NARX output is higher accurate than other methods. The Fig. 6. Show the different forecasting results in high and low power output clearly.

After calculating the validation factors by used forecast and actual power output, the results are shown in Table 2. Table 2 shows LSTM, which is 680 kW of RMSE, 429 kW of MAE, and 30.06 % MAPE, is the best forecasting method among these three methods by considered 3 of validation factors. However, the forecasting results of LSTM quite low accurately on the day that high power output, but the output shape is the same (different in size). For FNN, that is 798 kW of RMSE, 647 kW of MAE, and 35.56 % MAPE, the shape results of this method closely to LSTM, but there is more error than LSTM. For

2021 International Conference on Power, Energy and Innovations (ICPEI 2021)
October 20-22, 2021, Nakhon Ratchasima, THAILAND

NARX that is 680 kW of RMSE, 429 kW of MAE, and 30.06 % MAPE, the shape results are quite different from LSTM and FNN but, at the peak of high-power output day, the size of power output is closest than LSTM and FNN.

Moreover, the cause of FNN and LSTM error is the difference in the Power output of the test set and training set. The cause can be occurred by the difference of weather such as temperature, wind speed, or humidity that cause PV panel is lower efficiency.

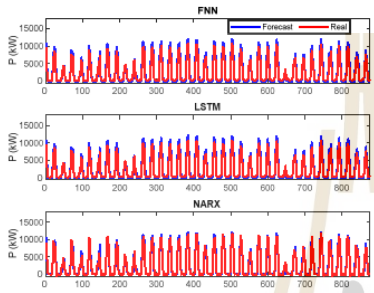


Fig. 5. Forecasting results for 875 of the test set

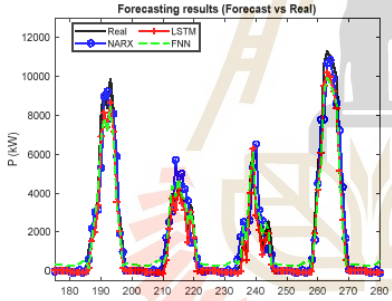


Fig. 6. The most different power output period

TABLE II. FORECASTING RESULTS

Network	RMSE (kW)	MAE (kW)	MAPE (%)
LSTM	680	429	30.06
FNN	798	647	35.56
NARX	761	429	43.93

V. CONCLUSION

This paper proposed the comparative study of short-term PV forecasting methods including FNN, LSTM, and NARX with 8,760 samples of time, solar irradiance, cell temperature, and power output for each hour. The simulation results show LSTM is the most accurately followed by FNN and NARX. Besides, at the peak of high-power output day, NARX is the most accurate in this range. The cause that makes a high error at this range is the difference of the weather data caused efficiency of PV panel is decreased. For the impact from dataset, the data in the test set should cover the season for which the forecasting model is being evaluated in order to achieve high performance.

ACKNOWLEDGMENT

This work was supported by the Suranaree University of Technology.

REFERENCES

[1] M. S. Hossain and H. Mahmood, "Short-Term Photovoltaic Power Forecasting Using an LSTM Neural Network and Synthetic Weather Forecast," *IEEE Access*, vol. 8, pp. 172524-172533, 2020, doi: 10.1109/ACCESS.2020.3024901.

[2] H. Zhou, Y. Zhang, L. Yang, Q. Liu, K. Yan, and Y. Du, "Short-Term Photovoltaic Power Forecasting Based on Long Short Term Memory Neural Network and Attention Mechanism," *IEEE Access*, vol. 7, pp. 78063-78074, 2019, doi: 10.1109/ACCESS.2019.2923006.

[3] H. Eom, Y. Son, and S. Choi, "Feature-Selective Ensemble Learning-Based Long-Term Regional PV Generation Forecasting," *IEEE Access*, vol. 8, pp. 54620-54630, 2020, doi: 10.1109/ACCESS.2020.2981819.

[4] S. Makhloufi, M. Debbache, and S. Boulahchiche, "Long-term Forecasting of Intermittent Wind and Photovoltaic Resources by using Adaptive Neuro-Fuzzy Inference System (ANFIS)," in *2018 International Conference on Wind Energy and Applications in Algeria (ICWEAA)*, 6-7 Nov. 2018 2018, pp. 1-4, doi: 10.1109/ICWEAA.2018.8605102.

[5] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural computation*, vol. 9, pp. 1735-80, 12/01 1997, doi: 10.1162/neco.1997.9.8.1735.

[6] F. Gers, J. Schmidhuber, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," *Neural computation*, vol. 12, pp. 2451-71, 10/01 2000, doi: 10.1162/089976600300015015.

[7] A. Di Piazza, M. C. Di Piazza, and G. Vitale, "Solar and wind forecasting by NARX neural networks," *Renewable Energy and Environmental Sustainability*, vol. 1, p. 39, 01/01 2016, doi: 10.1051/rees/2016047.

Journal of Renewable and Sustainable Energy

Deep-learning-based short-term photovoltaic power generation forecasting using improved self-organization map neural network

Cite as: J. Renewable Sustainable Energy 14, 043702 (2022); https://doi.org/10.1063/5.0091454
Submitted: 15 March 2022 • Accepted: 14 June 2022 • Published Online: 05 July 2022

Ntikorn Junhuathon and Keerati Chayakulkheeree

Journal of Renewable and Sustainable Energy ARTICLE scitation.org/journal/rse

Deep-learning-based short-term photovoltaic power generation forecasting using improved self-organization map neural network

Cite as: J. Renewable Sustainable Energy 14, 043702 (2022); doi: 10.1063/5.0091454
Submitted: 15 March 2022 · Accepted: 14 June 2022 ·
Published Online: 5 July 2022

Nitikorn Junhuathon[a] and Keerati Chayakulkheeree[b]

AFFILIATIONS

School of Electrical Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, Thailand

[a]E-mail: nitikorn_j@rmutt.ac.th
[b]Author to whom correspondence should be addressed: keerati.ch@sut.ac.th

ABSTRACT

As a vital function of an energy management system for distributed energy resources, optimal operation in distribution systems, and mitigating potentially adverse effects of photovoltaic (PV) systems, accurate forecasting of PV power generation is required. This article presents an alternative technique to improve the accuracy of deep-learning-based short-term PV power-generation forecasting models by clustering the input data using a self-organization map (SOM). To validate the proposed model, long short-term memory (LSTM), feed-forward neural network (FNN), FNN with the proposed SOM clustering method (FNN-SOM), and LSTM with the proposed SOM clustering method (LSTM-SOM) were tested and compared with one-year hourly datasets (8760 samples). Root mean square error, mean absolute error, and mean absolute percentage error were used as validation factors in this work. The results show that the proposed method provides a more accurate solar power generation forecast than other methods. Moreover, the proposed method can work effectively even with a few inputs system.

I. INTRODUCTION

Renewable energy penetration is greater than 20% at the national level in Denmark, Germany, and Ireland.[1] A regional grid would provide a more significant proportion of clean energy. For instance, China may likely meet all new electricity demands with renewable energy by 2035.[2] These projects ensure that in the extreme, instantaneous electric power supply could be entirely supplied by photovoltaics (PVs). This penetration can be considerably higher at the local level.[3,4] Due to the high variability of PV-generated electricity, power-system management and control become more challenging as its penetration increases by either renewable energy plants or separate renewable energy resource (RER) operations. According to this argument, forecasting solar energy production is essential for operation and management,[5,6] especially for short-term planning for distribution systems or low-marginal-risk power systems.[7] These requirements demonstrate why a significant amount of recent research has been dedicated to forecasting PV power generation.

Short-term PV power generation forecasting is essential for the efficient functioning of current power systems.[8,9] Due to the decentralization of the power grid, PV forecasting technologies have become much more prominent in today's power delivery networks. Real-time tracking of distribution networks, neighborhood battery usage, peak shaving techniques, and some benefits of demand response technology are only a few examples of how reliable one-step-ahead PV power-generation predictions are vital. Over the years, the issue of PV power-generation forecasting development has been well-established. Presently employed methods for predicting short-term PV production are primarily classified as either physical or mathematical and statistical methods.[10] The physical approach is based on the equations for solar irradiance conversion, PV module function, or other physical equations. This category creates a model using comprehensive and specific geographic-position details and weather, solar irradiance, and other data from PV plants. This solution, which is a significant challenge, relies on comprehensive and accurate geographic position knowledge as well as weather and solar irradiance data from the PV plants to establish the model and the physical formula that contains defects, indicating that the physical method's anti-interference capability and robustness are inadequate.[10,11] However, the disadvantage of this model is that it requires plenty of computing resources and is less effective than statistical approaches for short-term forecasting.

Journal of Renewable and Sustainable Energy ARTICLE scitation.org/journal/rse

The statistical approach is focused on correlations between the PV forecasting model's input and output variables. However, it does not include complex spatial knowledge about PV power plants,[9] which simplifies forecasting using mathematical regression methods such as the support vector machine (SVM)[12] and artificial neural networks (ANNs).[13,14] Artificial intelligence (AI)-based models capture the stochastic structures of PV power time series by using ANNs, and other deep-learning (DL) techniques were also proposed in Refs. 15 and 16. Both physical and statistical models have recently been merged and proposed hybrid models by Zhou *et al.*[17] Akhter *et al.* studied the success of DL, statistical, and hybrid models in forecasting. According to this extensive review, hybrid models outperform physical and statistical models that only focus on DL or mathematical techniques.[18] Wang proposed a hybrid short-term wind-speed prediction that utilized particle swarm optimization to determine the optimal regularization and kernel parameters regularization for an SVM.[12] In Ref. 13, Asrari *et al.* proposed a single-step-forward forecasting algorithm that combined metaheuristic optimization with an ANN. Meanwhile, a probabilistic forecasting method for single-step-ahead PV power forecasting that incorporates quantile regression and an extreme learning machine was proposed by Wan *et al.*[19] The prediction models in Refs. 13 and 19 demonstrate high precision, but their use is limited in real-time grid operations due to a PV generation sampling resolution of only 5 min. When the sampling resolution is greater than 15 min, long short-term memory neural network (LSTM NN)-based models outperform traditional NN-based models as shown in Ref. 17. In addition, the model suggested in Ref. 17 employed the attention mechanism and LSTM for the single-step forecasting of PV generation at sampling resolutions ranging from 7.5 min to 1 h.

Aside from the hybrid forecasting system, data processing with datasets before training improves forecasting performance. Hossain and Mahmood utilized the K-means algorithm to classify historical irradiance data into complex sky categories that differed from hour to hour within a single season to mitigate short-term PV power forecasting using LSTM.[16] To improve the method's performance in Ref. 20, Massaoudi *et al.* permitted one parameter and estimated the forecasting error increase in each case by calculating the probability value (P-value).

The literature survey illustrates that the most popular PV power generation forecasting models in the present research are LSTM and ANNs. These two models can use data-preprocessing or other statistical techniques to create features for improving the efficiency of the forecasting model. Moreover, numerous studies frequently incorporate additional variables, including various climate variables. However, small- to medium-scale PV systems frequently lack detailed measurements and some historical parameters. The historical data for these systems are usually limited to ambient temperature, solar irradiance, and power output. Other variables, such as cell temperature and angle of irradiance, which significantly affect the amount of energy generated by a PV system, usually have limited access over long periods in many areas.[21] If the relation of these variables can be estimated and used as inputs for a DL algorithm, the accuracy of the forecasting model will increase.

Therefore, a self-organization map (SOM) was proposed. SOMs were used to improve the clustering efficiency of a nonlinear problem, and the results demonstrated that this technique was effective for this type of problem. Additionally, this technique can illustrate the relationship between two or more parameters as multiple states. This paper presents an alternative method to improve the efficiency of a DL-based forecasting model with few inputs, which employs a SOM to estimate an unmeasured and related factor as one of the inputs.

The novelties of the proposed method are as follows:

(1) The proposed method is appropriate for photovoltaic power plants that typically measure only ambient temperature and irradiance. The sensor data are more precise than the weather web and, thus, significantly improve forecasting accuracy. In addition, some systems may make it challenging to find accurate information on other topic areas.
(2) Classifying the level of correlation between latent variables and PV power generation using SOM.
(3) Provide an alternative method for enhancing the accuracy of photovoltaic power generation forecasting.

The remainder of the article is organized as follows: Sec. II discusses the approach, data preparation, the clustering method, the DL architecture, and the proposed PV forecasting framework. Section III describes the dataset and system configuration used in a few case studies. Section IV contains results demonstrating the efficacy of the proposed method for increasing accuracy. Finally, Sec. V provides concluding remarks for the paper.

II. METHODOLOGY

As discussed previously, recent forecasting models for PV power generation consider the historical data of the PV plant and take advantage of available weather forecasting utilities or sensors at the PV plant. The weather forecast for a city area is available on public weather websites at hourly and regular resolution. Moreover, climate variables, such as temperature and solar irradiance, are nearly identical in city environments. However, certain critical factors cannot be obtained from public weather websites. Many PV plants do not collect historical data such as the cell temperature or accurate classifications of the sky in the region. As a result, the available data from a PV plant may be insufficient for PV forecasting. This can result in an erroneous forecast of PV generation at a specific plant location. Therefore, utilizing a relative state factor from SOM as an input to a forecasting model that can be correlated with the output level can be a beneficial strategy for increasing forecasting accuracy. The proposed model's data processing includes estimating unmeasured relative factors through clustering measured input. Cluster analysis is typically used to solve time series regression and classification.[22,23] This section will describe the ANN clustering-based preprocessing approach, including a dataset preparation, the estimation of relative unmeasured factors, and statistical analysis.

A. Approach overview

The first step to create a PV forecasting model is gathering and preprocessing the historical data, including PV power generation and weather data. Solar irradiance, temperature, and other related parameters are processed in this step. As mentioned, other related factors were unmeasured. This process can be accomplished by clustering the historical dataset into related groups using the clustering method for input to a DL framework based on pre-forecasting. It is proposed that the historical data were clustered into complex states of unmeasured factors for each hour of the day. The objective of defining unmeasured

Journal of Renewable and Sustainable Energy ARTICLE scitation.org/journal/rse

factors is to determine the factors for each hour related to the degree of irradiance, the time of day, the weather, and previous PV power production. Consequently, each hour of the day is divided into distinct clusters based on the factors mentioned above. In this work, distinct clusters will be called "states." A state is accomplished with the aid of a SOM algorithm. After obtaining the state using SOM, it will be used as one of the inputs to a DL-based PV forecasting model. The proposed forecasting framework will be compared to a recently developed method to ensure that the proposed method performs as expected.

B. Data preprocessing

Before going to the DL model, data have to go through two primary processes: clean data and preprocessing data. This paragraph will serve as a summary of preprocessing that can be used in its entirety. Data preprocessing aims to prepare our data for use with the DL model by selecting features or converting text to numeric values. In comparison to clean data, which focus on improving the accuracy of the data, such as dealing with undefinition (NAN) or no value (Null) data or dealing with outlier data, clean data focus on data analysis. Clean data steps are also included in this task. Data cleaning or data completion is the process of verifying and correcting (or removing) invalid data entries from a dataset, table, or database that have been designated as a "managed object." Because invalid data refer to incompleteness, inaccuracies, and correlation with other data, a database's cornerstone, these inaccuracies must be replaced, updated, or removed to ensure the quality of the data.

This work removes the dataset in the zero solar irradiance period, because those data will affect the training process's DL model parameters. The process to remove these data is shown in Table I, where x is the input matrix, y is the output matrix, k is the number of inputs, and $\hat{x}$ is the input matrix after preparation.

After obtaining the dataset from the data preprocessing step, it is processed through the clustering step. Clustering is the capability of grouping similar objects into a similar class of input clusters. The well-known methods to cluster include the K-means algorithm and SOMs. SOMs perform similarly to a constrained K-mean, but it is challenging to use the input vector category in this study to set the centroids. Therefore, a SOM was used to cluster the dataset in this step.

A SOM trains an ANN on patterns to generate classifications based on pattern similarities and relative topology. This is advantageous for delving into data or simplifying it before further processing. ANNs have shown their ability to perform well as classifiers and are especially well-suited to solve nonlinear problems. Given the nonlinear existence of real-world phenomena such as sorting, ANNs are undoubtedly an excellent candidate for solving the clustering problem. The input data for clustering problems are prepared for a SOM to cluster the dataset. Each row of the input matrix would contain the same number of elements as the calculated component. For this work, the four evaluated variables (i.e., time, temperature, irradiance, and PV output) will be fed into the SOM network, mapping the data j_{new} sample to a two-dimensional layer of neurons. The input data for clustering problems are prepared for a SOM as an input matrix. Each row jth of the input matrix will contain k elements, corresponding to a calculated vector from the PV plant data. Note that the jth sample contains k rows as an input set in this research. Specifying the number of neurons in each dimension of the layer provides SOM for classifying samples to obtain the state parameter of the dataset. A two-dimensional layer of neurons was used in a hexagonal grid. Using more neurons provides more detail, and adding dimensions enables modeling the topology of more complex function spaces. For the SOM process, given input x^j, the ith unit is found with the closest weight vector W_j by competition and $W_j^T x^j$ will be the maximum for each unit jth in the neighborhood $N(i)$ of winning neuron i to update the weights of j (W_j), and the weights outside of $N(i)$ is not updated (Table II). The SOM has three stages: (1) competition, (2) collaboration, and (3) weight update. For the competition stage, the most similar unit $i(x)$ is found with the following equation:

$$i(x) = \arg\max_j ||x - W_j||_2, \tag{1}$$

where $j = 1, 2, \ldots, m$ and m = samples. For the collaboration state, the lateral distance d_{ij} between the winner unit i and unit j is used in the following equations:

$$h_{i,j}(d_{ij}) = \exp\left(\frac{-d_j^2}{2\delta^2}\right), \tag{2}$$

$$\delta(n) = \delta_0 \exp\left(-\frac{n}{T}\right), \tag{3}$$

where h is the neighborhood function, n is the number of iterations, and T is constant. Weights-updated states are shown in the following equations:

$$W_j(n+1) = W_j(n) + \Delta W_j, \tag{4}$$

$$\Delta W_j = \eta y_j x - g(y_j) W_j, \tag{5}$$

where η is the learning rate, y_j is the output, and $g(y_j)$ can be found with the following equation:

$$g(y_j) = \eta y_i = \eta h_{ij}(x). \tag{6}$$

Figure 2 demonstrates the neuron-to-neuron relations. Typically, neighbors classify adjacent samples. The topology of the SOMs includes i neurons arranged in a hexagonal grid. Each neuron has acquired the ability to represent a distinct state class with neighboring neurons usually representing identical classes.

TABLE I. Data preprocessing process.

Data preparation
Input: $x \in R^{k \times j}$ contain time (j), temperature (j), and irradiance (j), PV output at previous step (j − 1)
Output: $y \in R^j$ PV output at previous step (j)
1: j = 1, …, 8760
2: k = 4
3: $\hat{x} = \{Remove\ x(all, j)\ and\ y(j)\ if\ x(irr(j), j) = 0\}$
$\hat{x} \in R^{k \times j_{new}}$ is an $(k \times j_{new})$ matrix

TABLE II. Self-organization map process.

SOM
Input: $\hat{x} \in R^{k \times j_{new}}$ is an $(k \times j_{new})$ matrix
Output: $i(\hat{x})$ is neighborhood i as Eq. (1)
1: for j = 1:1:8760
2: $N(\hat{x}_j) = \{i,\ if\ i(\hat{x}_j)$ closest to $N(i)\}$ as shown in Fig. 1.

Journal of Renewable and Sustainable Energy ARTICLE scitation.org/journal/rse

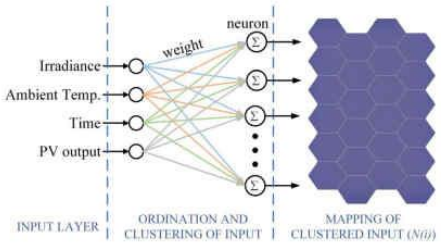


FIG. 1. SOM structure.

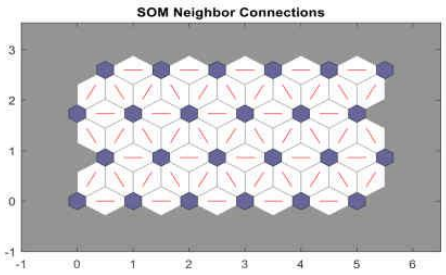


FIG. 2. SOM neighbor connections.

C. LSTM architecture

Hochreiter and Schmidhuberin introduced the LSTM in 1997.[22] A conventional LSTM comprises input, hidden, and output layers. The hidden layer comprises a collection of memory cells equipped with input and output gates. Then, Gers *et al.* enhanced the LSTM by introducing a new gate in the memory cell called the forget gate. These gates regulate the flow of information through a memory cell. The central component of the hidden layer is a memory block, which consists of a collection of memory cells that share the same gate units. Figure 3 illustrates the architecture of the LSTM block.[23]

Given an input $(x_t | t = 1, \ldots, T)$ with t frames, where i_t is the static feature of the t frame, the standard LSTM[14] is used to learn a sequence of hidden states $(h_t | t = 1, \ldots, T)$ to describe the dynamic of this input. A standard LSTM mainly consists of an input gate, a forget gate, an output gate, an input modulation gate, and a memory cell state, and one standard LSTM unit at time step t can be represented as follows:

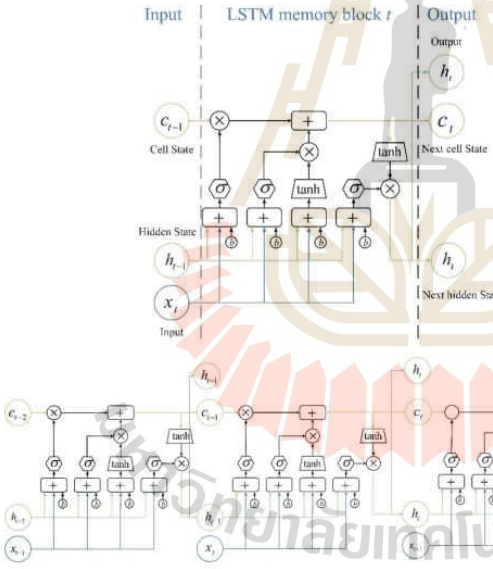


FIG. 3. The structure of the LSTM memory block.

Journal of Renewable and Sustainable Energy ARTICLE scitation.org/journal/rse

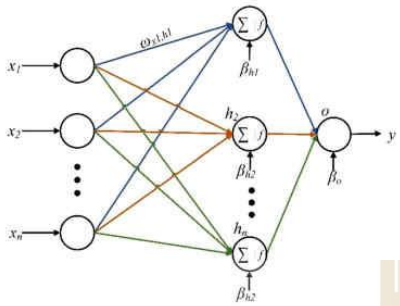


FIG. 4. Architecture of FNNs.

$$f_t = \sigma(W_{hf} X_t + W_{hf} h_{t-1} + b_f), \tag{7}$$

$$i_t = \sigma(W_{hf} X_t + W_{hi} h_{t-1} + b_i), \tag{8}$$

$$c_t^k = f_t c_{t-1} + i_t \tanh(W_{xc} X_t + W_{hc} h_{t-1} + b_c), \tag{9}$$

$$o_t^k = \sigma(W_{xo} X_t + W_{ho} h_{t-1} + b_o), \tag{10}$$

$$h_t^k = o_t \tanh(c_t), \tag{11}$$

where i_t, f_t, o_t, g_t, and c_t are the input gate, forget gate, output gate, input modulation gate, and memory cell state, respectively; σ is a sigmoid function, which denotes an elementwise product; W_k are weight matrices; and b is the bias vector. Precisely, the input gate controls the contributions of the newly arrived input data at time step t for updating the memory cell. In contrast, the forget gate f_t determines how much of the contents of the previous state c_{t-1} contribute to deriving the current state c_t. The output gate o_t learns how the output of the LSTM unit at time step t should be derived from the current state of the memory cell c_t.

D. Feed-forward neural network architecture

The primary aim of this network is to train the feed-forward neural network (FNN) to obtain the optimal weights and biases for improving the network's performance. The weights and biases of the training process are adjusted using a Levenberg–Marquardt backpropagation. The learning error computation function may also be used asmean absolute error (MAE), Root mean square error (RMSE), MSE, and mean absolute percentage error (MAPE). In this work, MSE is used and can be calculated with the following equation:

$$E_j = \sum_{i=1}^{n} MSE\left(y^j_{forecast,i}, y^j_{actual,i}\right), \tag{12}$$

where E_j is the training error at training iteration jth, $y^j_{actual,i}$ is the actual output at sample ith, x_i is the input, and n is the number of the training sample.

The workflow diagram of FNNs is shown in Fig. 4.

The forecasting output at training sample jth can be calculated using the following equation:

$$y_j = \sum_{j=1}^{n} \omega_{k,j} f(h_k) + \beta_j, \tag{13}$$

where $\omega_{k,j}$ are the weights from a hidden node kth to an output node at iteration jth, β_j is the bias of the output node at jth, and $f(h_k)$ is the outcome of hidden h_k. The fitness function is estimated using the training error (E), which can be calculated as follows:

$$\text{Fitness}(x) = \text{Minimize}\, E(x). \tag{14}$$

E. PV Forecasting framework

The historical data were processed for the PV forecasting framework to obtain the highly efficient data and remove outliers. Then, the processed data were input to SOM to classify the data type and use both processed and classified data to train the DL model, as shown in Fig. 5.

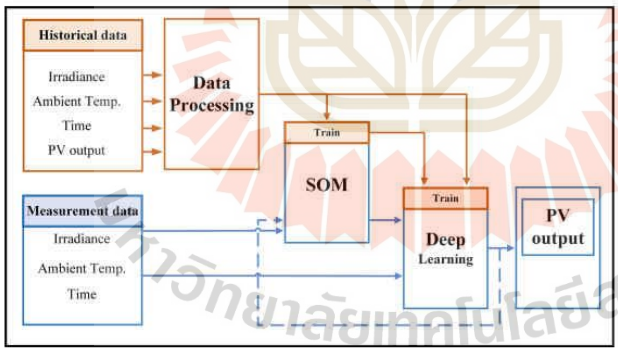


FIG. 5. Proposed PV power generation forecasting framework.

Journal of Renewable and Sustainable Energy ARTICLE scitation.org/journal/rse

F. Performance factor

To validate the performance of the proposed forecasting model, MAE, RMSE, and MAPE were evaluated in the following equations:

$$\mathrm{MAE} = \frac{\sum_{j=1}^{n} |y_{\mathrm{real}} - y_{\mathrm{forecast}}|}{n}, \tag{15}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{j=1}^{n} (y_{\mathrm{real}} - y_{\mathrm{forecast}})^2}{n}}, \tag{16}$$

$$\mathrm{MAPE} = \sum_{j=1}^{n} \frac{|y_{\mathrm{real}} - y_{\mathrm{forecast}}|}{y_{\mathrm{real}}} \times 100\%, \tag{17}$$

where y_{real} is the actual power from the PV plant, $y_{forecast}$ is the forecasted power from the PV, and n is the number of samples.

III. CASE STUDY AND CONFIGURATION FOR DL MODELS

A. Case study dataset

The dataset used in this research consisted of historical data from a PV plant for one year or 8760 h, including time, irradiance, ambient temperature, cell temperature, and power output. The hourly mean irradiance and power output from the PV plant, along with standard deviations, are shown in Fig. 6. The test system is a rooftop solar power plant with 14 MWp in Nakhon Ratchasima province, Thailand, in 2020. The peak generation is 13.20 MW during the training period, while the lowest peak day is 2.45 MW. The average peak generation capacity is 8.20 MW with a standard deviation of approximately 5.4 MW. The peak power generation capacity for the testing set is 13.20 MW, while the lowest peak day is 2.45 MW. The average peak generation capacity is 8.20 MW with a standard deviation of approximately 5.4 MW.

Figure 6 illustrates the user inputs in the training set and compares solar irradiance and PV power generation in the training set, validating set, and test set. Figure 6(a) shows the average solar irradiance, temperature, and PV power generation in a day during the training set

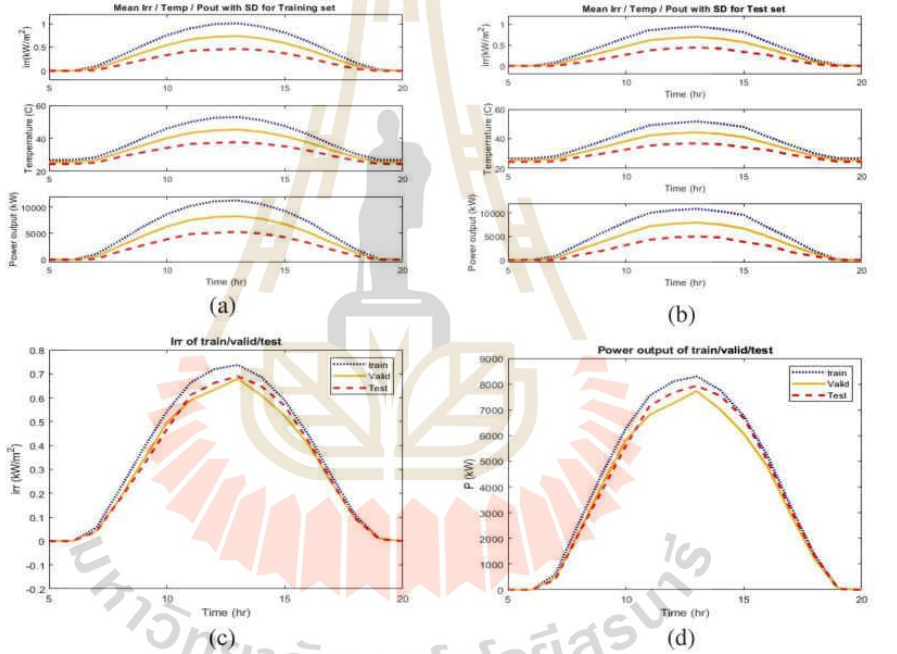


FIG. 6. (a) Irradiance, temperature, and power of the training set. (b) Irradiance, temperature, and power of the test set. (c) Irradiance of the training set, validation set, and test set. (d) Power output of the training set, validation set, and test set.

Journal of Renewable and Sustainable Energy ARTICLE scitation.org/journal/rse

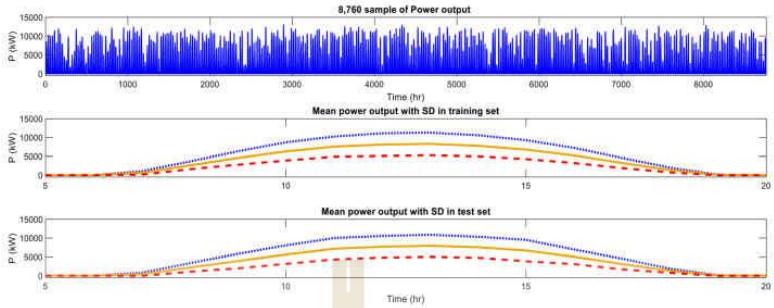


FIG. 7. The power output of the training set, validation set, and test set.

period. The average peak solar irradiance in a day is 0.6 kW/m^2 with a standard deviation of 0.5 kW/m^2. For temperature, the average peak in a day is 40 °C with a standard deviation of 15 °C. For PV power generation, the average peak in a day is 11 MW with a standard deviation of 7 MW. Figure 6(b) shows the average solar irradiance, temperature, and PV power generation in a day during the test set period. The average peak solar irradiance in a day is 0.6 kW/m^2 with a standard deviation of 0.5 kW/m^2. For temperature, the average peak in a day is 40 °C with a standard deviation of 15 °C. For PV power generation, the average peak in a day is 11 MW with a standard deviation of 7 MW (Fig. 7).

IV. RESULTS AND DISCUSSION

The simulation result will be discussed in two parts: (1) classification using the SOM and (2) power output forecasting results trained by DL. The SOM classes showing each of the four input features are presented in Fig. 8(a) in a weight plane representation, a visual representation of the weights that bind each input to one of the 24 neurons in the 6 × 4 hexagonal grid. Darker shades denote heavier weights than lighter ones. When four inputs have similar weight planes (their color gradients can be identical or inverted), they are strongly correlated. Additionally, it was discovered that when all four variables are weighted equally, the weight ratio of light intensity to output power tends to be equal, which is the opposite of the temperature trend. In terms of time, it carries a disproportionate amount of weight near the peak of light intensity and output power. As a result, Fig. 8 indicates a Euclidean interval between each neuron's class and its neighborhoods. The bright connections denote closely connected areas of the input space. However, the dark connections denote groups representing regions of the function space that are separated by few or no members. Extended boundaries with dark connections dividing vast areas of the input space suggest that the groups on either side of the boundary represent neighborhoods with significantly different characteristics.

Figure 9 shows the classes associated with each neighborhood and the number of classes included within each class. Neuronal areas with many hits correspond to groups that reflect identical densely populated regions of the function space. In contrast, areas with few hits signify parts of the feature space that are sparsely inhabited. After analyzing the correlation between the weights in Fig. 8 and the

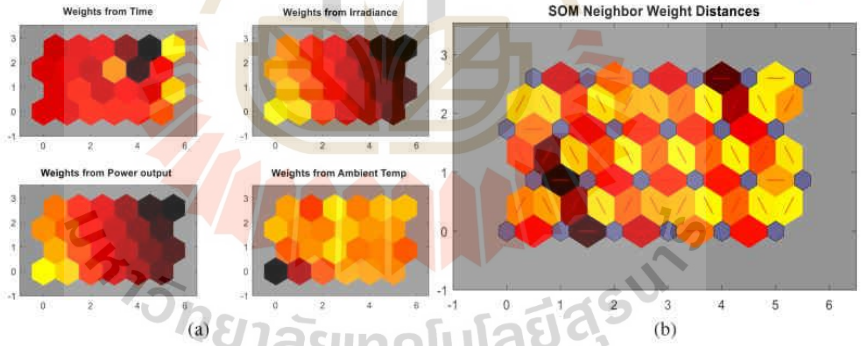


FIG. 8. (a) Weights from inputs; (b) SOM neighbor weight distances.

Journal of Renewable and Sustainable Energy ARTICLE scitation.org/journal/rse

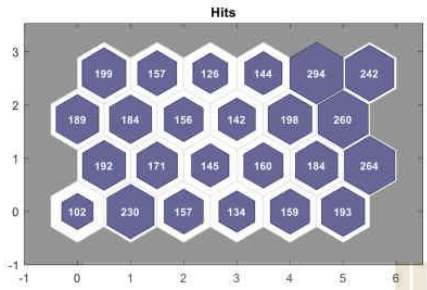


FIG. 9. Sample hits of SOM.

grouping results in Fig. 9, it was determined that the classifications were fairly distributed in the 4 × 6 plane but possessed a more significant fraction than the other classes, which indicated that the solar cells in the upper-right plane were more likely to generate energy.

After classifying the dataset, the resulting class data are taken as one of the DL features, including the FNN and LSTM. We then compared the prediction results from the proposed model to the conventional model that is non-classified. Only the results during the power generation period are shown (678 h). The forecasting results via LSTM are presented in Fig. 10(a) along with the forecasting results and error values. The highest error was 3104.5 kW, and the average error was 371.77 kW. The forecasting results via FNNs are presented in Fig. 10(b) along with the resulting and error values. The highest error was 853.51 kW, and the average error was 225.28 kW. The forecasting results via LSTM-SOM are presented in Fig. 10(c) along with the resulting error values. The highest error was 1371 kW, and the average

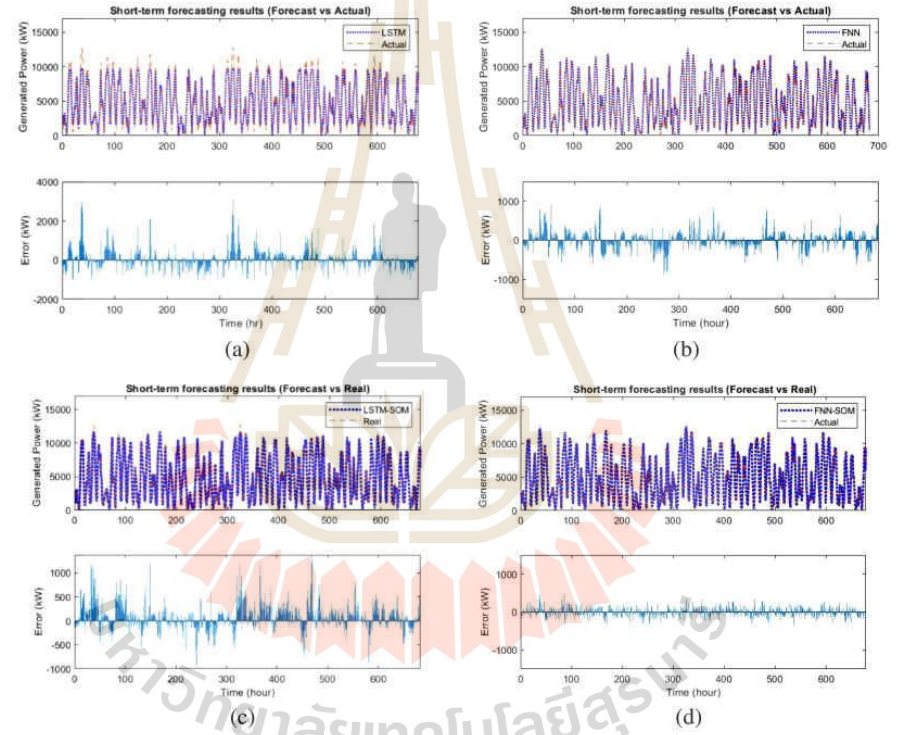


FIG. 10. The generated power forecast and error of (a) FNN, (b) LSTM, (c) LSTM-SOM, and (d) FNN-SOM.

Journal of Renewable and Sustainable Energy ARTICLE scitation.org/journal/rse

TABLE III. Comparing the simulation results of FNN/LSTM/LSTM-SOM/FNN-SOM. The italic values denote the best performance method.

Methods	MAPE (%)	MAE (kW)	RMSE (kW)
FNN	12.92	205.01	266.72
LSTM	17.38	371.77	551.05
FNN-SOM	*4.56*	*106.69*	*131.32*
LSTM-SOM	7.55	216.95	301.18

error was 216.95 kW. The forecasting results via FNN-SOM are presented in Fig. 10(d) along with the resulting error values. The highest error was 458.77 kW, and the average error was 106.69 kW. The forecast results during the peak period contain more errors than in other periods for all cases.

The simulation results are summarized in Table III. The FNN has a MAPE of 12.92%, a MAE of 205.01 kW, and a RMSE of 266.72 kW, whereas LSTM has 17.38%, a MAE of 371.77 kW, and an RMSE of 551.05 kW. FNN-SOM has an MAPE of 4.56%, an MAE of 106.69 kW, and an RMSE of 131.32 kW, whereas LSTM-SOM has an MAPE of 7.55%, an MAE of 216.95 kW, and an RMSE of 301.18 kW. As shown in the simulation results, the model that had used the clustering method to cluster the dataset before being used in the training process was more accurate than the conventional method, especially in the peak power production period.

V. CONCLUSION

This article presents an alternative technique to improve the accuracy of deep-learning-based short-term PV power-generation forecasting models by clustering the input data using a self-organization map (SOM) and data-preprocessing. The most widely used forecasting methods in this field were simulated and compared to the proposed method to validate its efficiency. Particularly in FNNs for fitting problems, a SOM was used to map a set of numeric inputs to a set of numeric targets. The simulation results indicated that clustering via the SOM provided better PV power generation forecasting and can be used in place of the FNN and LSTM, which are DL-based forecasting methods. In addition, the proposed method can work effectively even with a few inputs system.

ACKNOWLEDGMENTS

This study was supported by Suranaree University of Technology.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Nitikorn Junhuathon: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (lead); Resources (equal); Software (lead); Validation (equal); Visualization (equal); Writing – original draft (lead). **Keerati Chayakulkheeree:** Conceptualization (equal); Data curation (equal); Formal analysis (supporting); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Resources (equal); Supervision (lead); Validation (equal); Visualization (equal); Writing – review and editing (lead).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] B. Kroposki *et al.*, "Achieving a 100% renewable grid: Operating electric power systems with extremely high levels of variable renewable energy," IEEE Power Energy Mag. **15**(2), 61–73 (2017).
[2] G. Fu, J. Liu, and R. Liu, "Quantitative analysis of the feasibility of realizing the transformation to clean energy for China's energy increment by 2035," in *2018 International Conference on Power System Technology (POWERCON), 6–8 November* (IEEE, 2018), pp. 510–515.
[3] Y. Gabdullin, C. Xerri, B. Azzopardi, K. Cilia, and G. Portelli, "Solar photovoltaics penetration impact on a low voltage network a case study for the Island of Gozo, Malta," in *2018 IEEE Power & Energy Society General Meeting (PESGM), 5–10 August* (IEEE, 2018), pp. 1–5.
[4] T. Aziz and N. Ketjoy, "Enhancing PV penetration in LV networks using reactive power control and on load tap changer with existing transformers," IEEE Access **6**, 2683–2691 (2018).
[5] T. Jamal, C. Carter, T. Schmidt, G. M. Shafullah, M. Calais, and T. Urmee, "An energy flow simulation tool for incorporating short-term PV forecasting in a diesel-PV-battery off-grid power supply system," Appl. Energy **254**, 113718 (2019).
[6] M. Q. Raza, M. Nadarajah, and C. Ekanayake, "Demand forecast of PV integrated bioclimatic buildings using ensemble framework," Appl. Energy **208**, 1626–1638 (2017).
[7] X. Ren, N. Yang, B. Ye, Y. Yao, and C. Gao, "Stochastic planning model for increment distribution network considering CVaR and wind power penetration," in *2019 IEEE Innovative Smart Grid Technologi—s-Asia (ISGT Asia), 21–24 May* (IEEE, 2019), pp. 1358–1363.
[8] C. Feng, M. Cui, B. M. Hodge, S. Lu, H. F. Hamann, and J. Zhang, "Unsupervised clustering-based short-term solar forecasting," IEEE Trans. Sustainable Energy **10**(4), 2174–2185 (2019).
[9] T. Mutavhatsindi, C. Sigauke, and R. Mbuvha, "Forecasting hourly global horizontal solar irradiance in South Africa using machine learning models," IEEE Access **8**, 198872–198885 (2020).
[10] L. Ge, Y. Xian, J. Yan, B. Wang, and Z. Wang, "A hybrid model for short-term PV output forecasting based on PCA-GWO-GRNN," J. Mod. Power Syst. Clean Energy **8**(6), 1268–1275 (2020).
[11] C. Wan, J. Zhao, Y. Song, Z. Xu, J. Lin, and Z. Hu, "Photovoltaic and solar power forecasting for smart grid energy management," CSEE J. Power Energy Syst. **1**(4), 38–46 (2015).
[12] X. Wang, "Forecasting short-term wind speed using support vector machine with particle swarm optimization," in *2017 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), 16–18 August* (IEEE, 2017), pp. 241–245.
[13] A. Asrari, T. X. Wu, and B. Ramos, "A hybrid algorithm for short-term solar power prediction—sunshine state case study," IEEE Trans. Sustainable Energy **8**(2), 582–591 (2017).
[14] L. Gutiérrez, J. Patiño, and E. Duque-Grisales, "A comparison of the performance of supervised learning algorithms for solar power prediction," Energies **14**(15), 4424 (2021).
[15] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches," Energies **11**(7), 1636 (2018).
[16] M. S. Hossain and H. Mahmood, "Short-term photovoltaic power forecasting using an LSTM neural network and synthetic weather forecast," IEEE Access **8**, 172524–172533 (2020).
[17] H. Zhou, Y. Zhang, L. Yang, Q. Liu, K. Yan, and Y. Du, "Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism," IEEE Access **7**, 78063–78074 (2019).

[18]M. N. Akhter, S. Mekhilef, H. Mokhlis, and N. M. Shah, "Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques," IET Renewable Power Gener. 13(7), 1009–1023 (2019).
[19]C. Wan, J. Lin, Y. Song, Z. Xu, and G. Yang, "Probabilistic forecasting of photovoltaic generation: An efficient statistical approach," IEEE Trans. Power Syst. 32(3), 2471–2472 (2017).
[20]M. Massaoudi *et al.*, "An effective hybrid NARX-LSTM model for point and interval PV power forecasting," IEEE Access **9**, 36571–36588 (2021).
[21]J. Wang, H. Zhong, X. Lai, Q. Xia, Y. Wang, and C. Kang, "Exploring key weather factors from analytical modeling toward improved solar power forecasting," IEEE Trans. Smart Grid **10**(2), 1417–1427 (2019).
[22]S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput. **9**(8), 1735–1780 (1997).
[23]X.-H. Le, H. V. Ho, G. Lee, and S. Jung, "Application of long short-term memory (LSTM) neural network for flood forecasting," Water **11**(7), 1387 (2019).

BIOGRAPHY

Mr. Nitikorn Junhuathon was born on February 3, 1995 in Hua Thalaeng city, Nakhonratchasima province, Thailand. He obtain a Bachelor's degree and Master' degree in electrical engineering from Suranaree university of technology, from 2014 - 2017 and 2017 – 2019, respectively. After graduating, he was a lecturer in electrical engineering at Bangkok Thonburi University for 1 year (2019-2020). Then, He move to be a lecturer in electrical engineering at Faculty of Engineering, Rajamangala University of Technology Thanyaburi until now (2020 - present).