MINISTRY OF EDUCATION AND TRAINING

HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION

FACULTY OF MECHANICAL ENGINEERING

HCMUTE

PROJECT OF ROBOTICS AND AI

PREDICTING ELECTRICITY CONSUMPTION USING AI MODELS

Supervisor:	**Dr. Trần Vũ Hoàng**
Students:	**Ngô Văn Môn**
ID:	**21134011**
Class:	**21134NT**

Ho Chi Minh City, December 2024

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION

FACULTY OF MECHANICAL ENGINEERING

DEPARTMENT OF MECHATRONICS

PROJECT OF ROBOTICS AND AI

PREDICTING ELECTRICITY CONSUMPTION USING AI MODELS

Supervisor:	**Dr. TRAN VU HOANG**
Student:	**NGO VAN MON**
Student ID:	**21134011**
Class:	**21134NT**
Year of Admission:	**2021 - 2025**

Ho Chi Minh City, December 2024

COMMITMENT

- Project: PREDICTING ELECTRICITY CONSUMPTION USING AI MODELS
- Supervisor: Dr. Tran Vu Hoang
- Students: Ngo Van Mon
- ID: 21134011
- Phone number: 0399728330
- Project report submission date: 12/2024
- Commitment: "I hereby affirm that this graduation thesis is the product of my own research and effort. I have not duplicated any content from published sources without proper acknowledgment. In the event of any infringement, I take full responsibility for my actions."

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to Ho Chi Minh City University of Technology and Education, particularly the distinguished members of the Mechanical Engineering Faculty. Their steadfast support and invaluable guidance have been essential throughout my academic journey, significantly contributing to the completion of an important part of my training program. The faculty's exceptional teaching has provided me with not only in-depth technical knowledge but also a strong foundation for academic and professional growth.

I am especially grateful to Dr. Tran Vu Hoang, whose expertise and mentorship have been pivotal in addressing the challenges of this project. His thoughtful guidance and insightful feedback have greatly enhanced the scope and quality of my research.

As with any academic work, errors are inevitable, and I welcome any constructive feedback and suggestions from the academic community to further improve the comprehensiveness of this report. Your input will be invaluable in refining my findings.

Sincerely,
Ngo Van Mon

ABSTRACT

Efficient energy management is a critical aspect of modern society, especially as electricity demand continues to grow and sustainability goals become increasingly important. Accurate electricity consumption forecasting is essential for optimizing energy distribution, reducing costs, and preventing system overloads. This project focuses on developing AI-based models to predict electricity consumption, providing valuable insights for energy management systems.

The primary goal of the project is to design AI model capable of analyzing historical electricity consumption data and predicting future usage trends. This model is compatible to handle the complexities of time series data, such as trends, seasonality, and irregular patterns, ensuring high accuracy in forecasts.

The implementation process involves data collection, preprocessing, feature engineering, and model optimization to achieve optimal performance. Additionally, a user-friendly interface has been developed to present the prediction graph of model.

The project results demonstrate that AI models can effectively predict electricity consumption patterns, offering a reliable tool for energy providers and policymakers to optimize resource usage and sustainable energy development.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	**A**rtificial **I**ntelligence
ANN	**A**rtificial **N**eural **N**etwork
ARIMAX	**A**uto **R**egressive **I**ntegrated **M**oving **A**verage
DL	**D**eep **L**earning
EVN	**V**ietnam **E**lectricity
GRU	**R**ecurrent **N**eural **N**etwork
GUI	**G**raphical **U**ser **I**nterface
LSTM	**L**ong **S**hort **T**erm **M**emory
MAPE	**M**ean **A**bsolute **P**ercentage **E**rror
ML	**M**achine **L**earning
MSE	**M**ean **S**quared **E**rror
RNN	**R**ecurrent **N**eural **N**etwork
RMSE	**R**oot **M**ean **S**quared **E**rror
TCN	**T**emporal **C**onvolutional **N**etwork

CHAPTER 1. INTRODUCTION

1.1. Thesis background

In Vietnam, the increasing demand for electricity, coupled with the limitations of the power generation infrastructure, has led to frequent power shortages, especially during the hot summer months. These power shortages not only disrupt daily life but also affect industrial activities, services. The need for efficient energy management has become more urgent than ever, with growing concerns about the sustainability of energy sources and meet electricity demand.

Figure 1.1. Checking the quality of chicken eggs at Ngoc Mung Poultry Breeding Joint Stock Company (Dong Anh district)

(source: [1])

Recent advances in artificial intelligence (AI) and machine learning (ML) have opened up new possibilities for forecasting electricity consumption. By analyzing historical consumption patterns and identifying underlying trends and seasonal fluctuations, AI models can predict future electricity demand with high accuracy. This predictive capability helps agencies and organizations in Vietnam, such as Vietnam Electricity (EVN) as well as the Government and local authorities to plan electricity supply more effectively, forecast periods of high demand and reduce the risk of power shortages, thereby ensuring stable electricity supply to residential, industrial and commercial sectors, especially during the hot summer months when electricity demand is high.

The project aims to develop AI-based models designed to predict electricity consumption in Vietnam, in order to mitigate the risk of power shortages during peak periods. By leveraging algorithms to analyze historical data and predict consumption trends, the system will help power suppliers and managers optimize their supply strategies, reduce the impact of shortages, and contribute to more sustainable energy management in Vietnam.

1.2. Scientific and practical significances

- Scientific significance: The study establishes the research and development of artificial intelligence applications in the field of time series data prediction. From there, develop and conduct in-depth research on data processing techniques and specialized models for time series forecasting.
- Practical significance: The application of AI in predicting future electricity consumption will contribute an important part to the country's development.

1.3. Objectives

This project aims to forecast Vietnam's future electricity consumption by collecting and analyzing monthly data, training and comparing models to select the best one, and developing a Graphical User Interface (GUI) application to visualize the predictions.

1.4. Research methods

- **Descriptive Research**: Investigating and surveying scientific papers related to the topic of energy consumption prediction.
- **Analytical Research**: Collecting data and analyzing the factors that have a significant impact on energy consumption.
- **Comparative Research**: Studying the advantages and disadvantages of AI models used to train time series data.

1.5. Structure of the report

The report contains four chapters:

- **CHAPTER 1. INTRODUCTION:** Brief introduction to the study.
- **CHAPTER 2: THEORETICAL BASIS**: This section studies and surveys the methods of predicting electricity consumption of the papers done with models including deep learning and non-deep learning. In addition to training

the models, this section will also present theories on analyzing and processing time series data. The theory of LSTM model and implementation for AI systems will also be discussed in this section.

- **CHAPTER 3: PROPOSED METHOD**: Providing the design requirements for the trained AI model, the GUI software that allows for the prediction observation, and the proposed method to achieve that goal.
- **CHAPTER 4: EXPERIMENTAL RESULT**: Present the project implementation process, results achieved and limitation.
- **CHAPTER 5: CONCLUSION AND RECOMMENDATION:** Evaluate the achievements and limitations of the project, and propose a direction for project development.

CHAPTER 2. THEORETICAL BASIS

2.1. Electricity consumption forecasting classification

Forecasting electricity consumption is a crucial aspect of energy management, helping electricity supply and government plan for future electricity energy needs. The classification of electricity consumption forecasting can be divided based on length forecast. Each type of forecasting serves a specific purpose in energy planning and decision-making processes. Below are the classifications accroading Edoka et al (2023) [2]:

- Very Short-Term Forecasting (VSTLF)
  - Lead Time: Few minutes to half an hour.
  - Applications: Real-time grid control and security evaluation.
- Short-Term Forecasting (STLF)
  - Lead Time: Half an hour to a few hours.
  - Applications: Spinning reserve allocation, unit commitment, and maintenance scheduling.
- Medium-Term Forecasting (MTLF)
  - Lead Time: Few days to a few weeks.
  - Applications: Seasonal planning for dry and rainy seasons.
- Long-Term Forecasting (LTLF)
  - Lead Time: Few months to a few years.
  - Applications: Planning the growth of generation capacity and infrastructure development.

This classification is also the basis for my reference in system design and proposed method in the next chapter.

2.2. The methods used in forecasting electricity consumption

2.2.1. Non-deep learning approaches methods

Before the emergence of deep learning, research on time series prediction was mainly focused on traditional machine learning and statistical methods. Below are some notable studies.

Romeo Djimasbe et al. (2024) developed an ARIMAX model to predict electricity consumption at Kotoka International Airport (KIA) in Ghana [3]. The model integrates time series data of airport electricity usage with exogenous variables, such as weather conditions and air passenger traffic.

Prasad at al. (2023) focus on predicting electricity consumption using a hybrid model that combines Random Forest Regression and Linear Regression [4]. The authors utilize smart meter data, which provides real-time, granular electricity usage information, for model trainning. By merging the strengths of two ML models, the study aims to improve prediction precision.

Non-deep learning methods are effective in forecasting electricity consumption, especially for datasets with clear trends or seasonality. These approaches, including statistical and traditional machine learning techniques, are valued for their simplicity, interpretability, and lower computational requirements. While they perform well on linear or moderately complex data, their limitations become evident when handling highly nonlinear relationships or large, multidimensional datasets.

2.2.2. Forecasting electricity consumption by Deep learning (DL) model

The use of Deep learning (DL) in predicting electricity consumption is a well-explored area, as reflected in the significant volume of existing research. Below are some notable articles related to my project.

Hadjout et al. (2021) developed an ensemble deep learning framework to predict electricity consumption in the Algerian market [5]. Their approach combines Deep Learning (DL) models including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Temporal Convolutional Network (TCN) to focus on improving the prediction accuracy by combining predictions from different architectures. Although this work also aims to predict electricity consumption using advanced AI techniques, it differs from our project in that it focuses on model design but not on features that have a large impact on the prediction.

Li (2021) proposed a deep learning-based framework for energy consumption prediction, focusing on household electricity use in China. The study applied multiple discrete models, including LSTM, GRU, and TCN, to analyze multidimensional time series data incorporating weather characteristics [6]. By comparing the performance of the models, Li determined that TCN was the most effective model in capturing temporal patterns and achieving high forecast accuracy. Although this study is similar to our electricity consumption forecasting goal, but it focuses on optimizing forecasts for households, rather than on a national scale.

Lee et al. (2020) developed an artificial neural network (ANN)-based urban growth model to predict energy consumption patterns in Vietnam [7]. Their method integrates urban growth factors such as population and nighttime light intensity to provide more accurate regional energy demand forecasts. The model was trained using data from 1995, 2000, 2005, and 2010, and the model predictions for 2015 were validated against actual light distribution data. Although the project has a similar goal of predicting Vietnam's electricity consumption based on relevant features, it only uses a simple ANN model and has not applied DL models specifically for time series prediction.

2.3. Time Series Data Analysis

2.3.1. Overview of time series data

The data used to train the AI model to predict future electricity consumption is in the form of time series, so the basis for initial implementation is to clearly understand the definition and characteristics of this type of data.

Time series data is a collection of values recorded or gathered at specific points in time, usually in a continuous sequence. The distinctive feature of this type of data is its dependence on temporal order, meaning the position of each data point in the timeline is crucial. In practice, time series data appears in various fields, such as daily stock prices, monthly sales figures, yearly climate data, or hourly energy consumption [8].

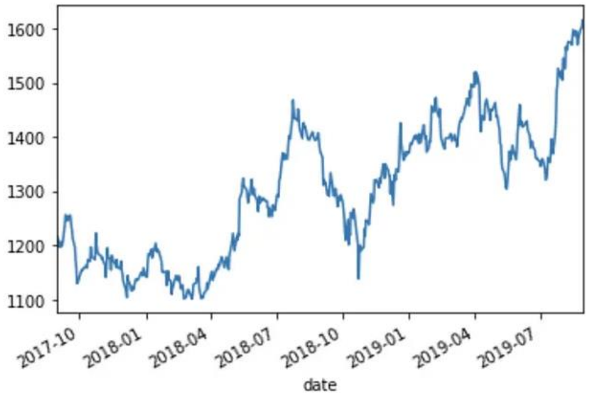


Figure 2.1. Stock price values (source: [9])

Time series data can be categorized based on its structure and method of collection in several ways:

By number of variables:

- Univariate Data: Focuses on a single variable or factor, such as daily recorded temperatures.
- Multivariate Data: Involves multiple variables or factors collected simultaneously, such as temperature, humidity, and rainfall recorded daily.

By Time Intervals:

- Regular Data: Collected at fixed, consistent intervals, such as hourly, daily, or monthly measurements.
- Irregular Data: Collected at unpredictable or undefined intervals, often tied to specific events, such as data gathered during special occurrences or incidents.

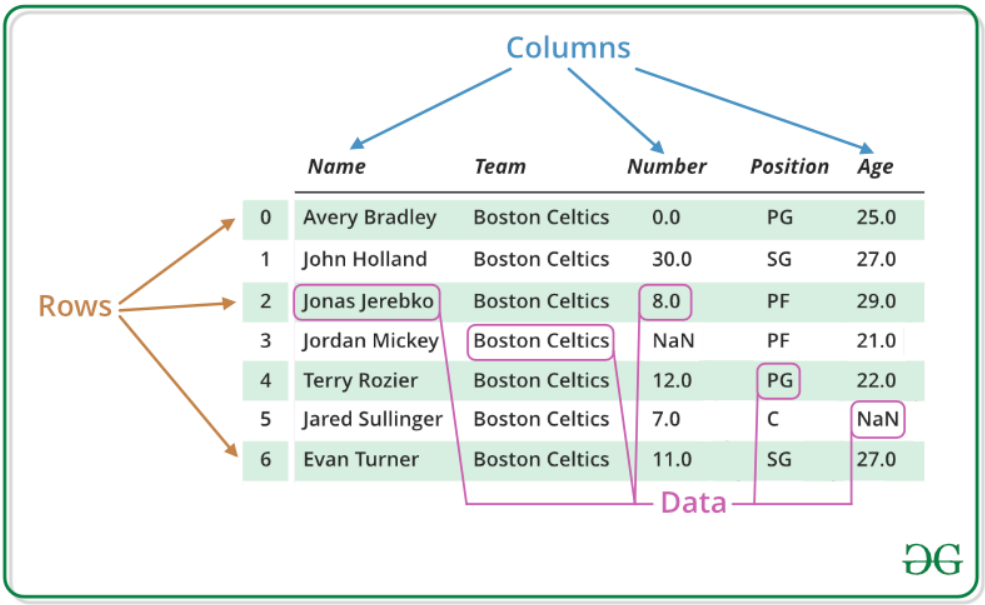
The characteristics of time series data includes:

- Trend: Reflects long-term changes or movements in the data over time.
- Seasonality: Represents patterns that repeat at fixed intervals.
- Cyclic Patterns: Reflect fluctuations that are not fixed in duration and are often influenced by factors like economic cycles.
- Irregular Components: Represent random and unpredictable variations.

2.3.2. Methods for analyzing time series data

To be able to feed data into AI models for effective training, the first step is to analyze and understand the collected time series data. Below are some theories that I have learned.

First I mention the data collection and cleaning time series data. It is usually collected from various sources such as databases, APIs or CSV files. The commonly used programming language is of course Python and the Pandas library is used to load this data into a suitable format for analysis.



	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

Figure 2.2. Pandas Dataframe

(source: [10])

Missing values and outliers can greatly influence the accuracy of data analysis, making it crucial to address them properly. To handle missing values, approaches like forward fill, backward fill, or interpolation can be applied to estimate and replace gaps in the data. For outliers, statistical techniques such as Z-score or Interquartile Range (IQR) are commonly used to identify and remove anomalies, ensuring the dataset's reliability and consistency.

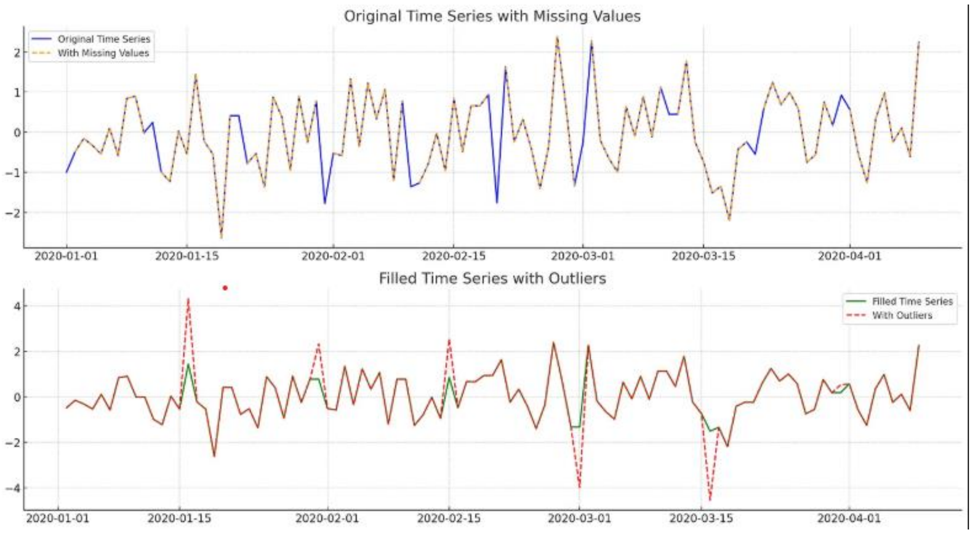


Figure 2.3. Filled Time Series with Outliers

(source: [8])

Data transformation is the process of changing raw data to highlight important features and prepare the data for analysis. Smoothing techniques such as moving averages help reduce noise and highlight long-term trends in data. Differencing is used to remove trends and seasonality, turning time series into static data, suitable for models such as ARIMA. Additionally, scaling as Min-Max Scaling and normalization as Z-score Normalization help bring data into a certain range, which improves the performance of ML and DL algorithms. These techniques play an important role in improving the quality and efficiency of data analysis.

2.4. AI model for electricity consumption forecasting

Through surveying articles, I found that models such as LSTM, ARIMAX, TCN, RNN, GRU are quite commonly used in research works. However, because the LSTM model is often proposed more, I will review its theory.

LSTM is considered a model capable of processing time series data effectively thanks to its intelligent information storage and management mechanism, helping to capture long-term relationships and make accurate predictions in complex problems. *Figure 2.7* below describes the architecture of this DL model.

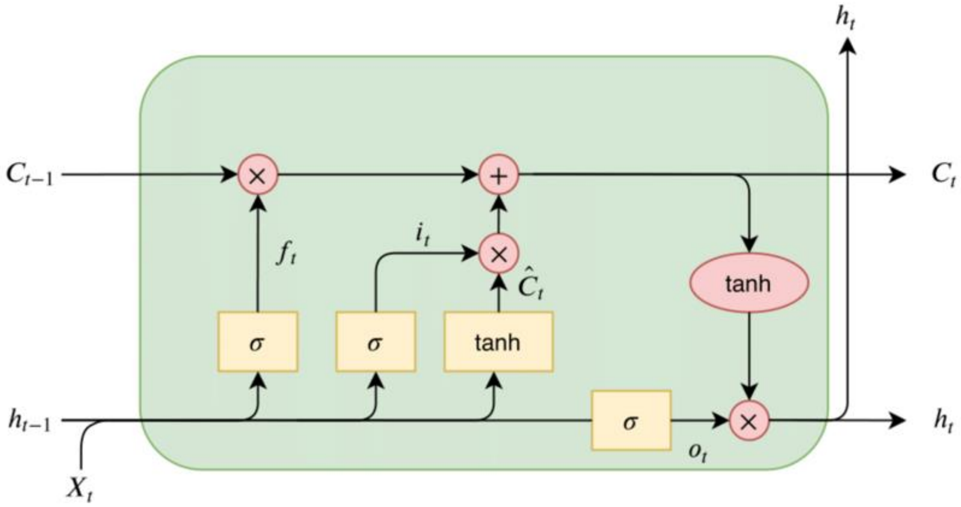


Figure 2.4. Long Short Term Memory Unit architecture

(source: [11])

The architecture of an LSTM includes the following main components:

- Forget Gate (f_t): This gate decides which information from the previous cell state (C_{t-1}) should be discarded. It uses a sigmoid activation function to scale the importance of each piece of information between 0 (discard completely) and 1 (retain fully).

- Input Gate (i_t): This gate determines which new information will be added to the cell state. It works in tandem with the candidate cell state ($\hat{C}_t$), which generates potential updates for the memory using a tanh activation function.
- Cell State (C_t): Acts as the long-term memory of the LSTM, storing information across time steps. The cell state is updated by combining the retained old information (controlled by the forget gate) with the new information (controlled by the input gate).
- Output Gate (o_t): Determines which part of the cell state will be used to generate the hidden state (h_t), which represents the current output of the LSTM. This output is a filtered version of the updated cell state, modulated by the output gate and passed through a tanh activation function.

Operation of the LSTM model as below:

The first step in an LSTM is deciding which information will be discarded from the cell state, which is performed by a sigmoid layer called the "forget gate." Its input is h_{t-1} and x_t, and it outputs a value in the range [0, 1] for each state C_{t-1}. If the value is 1, the information is fully retained; if the value is 0, the information is completely discarded.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t\,] + b_f\right) \qquad (2.1)$$

The next step is deciding what new information will be stored in the cell state. This is done through two layers: a sigmoid layer called the "input gate" that decides which values will be updated, and a tanh layer that generates a new vector C_t that can be added to the state. These two components are then combined to create an update for the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t\,] + b_i) \qquad (2.2)$$

$$\hat{C}_t = tanh(W_C \cdot [h_{t-1}, x_t\,] + b_C) \quad (2.3)$$

Then, we update the previous cell state C_{t-1} to the new cell state C_t as follows:

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \qquad (2.4)$$

Finally, we calculate the output value based on the cell state, but it is a filtered version. First, we apply the sigmoid layer to decide which part of the cell state will be output, then push the cell state through tanh and multiply it by the output of the sigmoid gate.

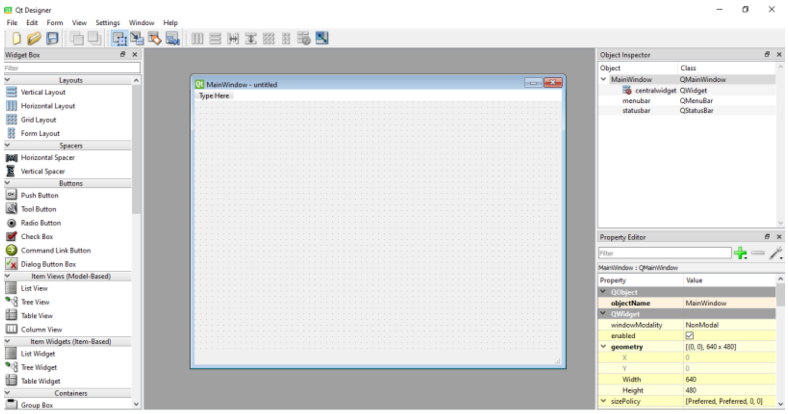
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t\,] + b_o) \qquad (2.5)$$

$$h_t = o_t * tanh(C_t) \qquad (2.6)$$

Through these steps, the LSTM network is capable of learning and storing information over long periods of time, retaining the important parts of the data sequence and discarding the unnecessary ones.

2.5. QT Designer application interface design software

After completing the steps of data processing and training the AI model, we need to consider the step of deploying the model to a GUI software. For predictive AI applications that do not require fast response, deploying on a computer using the Python programming language is completely suitable for Frameworks such as Tkinter, Turtle, Kivy, ... but the most prominent is QT Designer. QT Designer software stands out with the advantages of being compatible with Windows interface support, simple layout, easy customization and only needing to load the designed interface file (.ui format) to put into use in the main program.



Object	Class
MainWindow	QMainWindow
centralwidget	QWidget
menubar	QMenuBar
statusbar	QStatusBar

Property	Value
QObject	
objectName	MainWindow
QWidget	
windowModality	NonModal
enabled	☑
geometry	[(0, 0), 640 x 480]
X	0
Y	0
Width	640
Height	480
sizePolicy	[Preferred, Preferred, 0, 0]

Figure 2.5. Main interface of QT Designer

When starting QT Designer, the interface is displayed as shown in *Figure 2.5*. With toolbars on both sides, it is convenient for designing application interface software. It is very easy to create a software with buttons, entry boxes, display areas, users just need to drag and drop related properties to create the desired interface.

CHAPTER 3. PROPOSED METHOD

3.1. Requirement of system design

3.1.1. AI model for forecasting electricity consumption

With the goal of forecasting Vietnam's national electricity consumption demand using the Long-Term Forecasting (LTLF) model, helping EVN and the Vietnamese Government plan electricity production and develop electricity infrastructure to meet future electricity consumption needs, the designed AI model must forecast electricity consumption demand for each month in the next few months to a year (12 months) with a Mean Absolute Percentage Error (MAPE) of less than 10%.

3.1.2. GUI software for AI implementation

The GUI software that allows viewing the AI model's predictions will allow the user to enter the last time of the prediction, then the software integrating the trained AI model will calculate and display a graph of the predicted future power consumption, and also allows the user to save the values displayed on the graph.

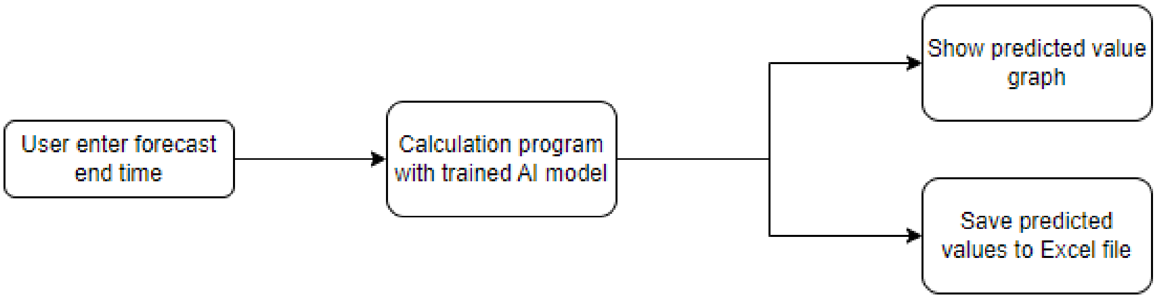

User enter forecast end time
Calculation program with trained AI model
Show predicted value graph
Save predicted values to Excel file

Figure 3.1. GUI program operation

3.2. Proposed AI model

I conducted a survey and research on models commonly used in articles about time series forecasting in general and electricity consumption forecasting in particular. The following comparison table summarizes the evaluation of different models based on key metrics and their brief performance remarks follow the *Table 3.1*

Model	Strengths	Weaknesses	Performance
ARIMAX	Effective for linear trends and seasonality	Limited in handling non-linear and complex data	Performs well with simple and linear patterns but struggles with highly dynamic and complex datasets
RNN	Good for sequential data, simple structure	Struggles with long-term dependencies	Adequate for capturing short-term trends but often underperforms on long-term forecasts due to vanishing gradients
GRU	Handles long-term dependencies better than RNN, faster training than LSTM.	May underperform compared to LSTM in certain tasks	Balances accuracy and efficiency but might lack precision for highly complex time series data.
TCN	Parallel processing, handles long sequences well.	May require more data preprocessing	Strong in parallel computations and long sequences but requires careful design and preprocessing efforts
LSTM	Captures long-term patterns, robust for time series data	Higher training time, requires parameter tuning	Consistently provides the best performance in handling sequential and non-linear patterns in forecasting

Table 3.1. Performance comparison of AI models for forecasting

The system needs to predict long-term electricity consumption with low error, ensuring the ability to effectively handle historical data and complex, nonlinear external data such as weather, seasonality or socio-economic indicators. With these requirements, LSTM is considered the most suitable model thanks to its superior ability to capture long-term and nonlinear relationships, ensuring high forecasting performance and flexibility in integrating influencing factors into the prediction system. Therefore, LSTM will be selected for this project.

Regarding input data processing before training, I propose a method according to *Figure 3.2.* Electricity consumption data is combined with external factors such as weather data and temporal information (month of the year) to create a forecasting dataset. These inputs are normalized to scale all values within a consistent range, ensuring effective learning and minimizing the impact of differing measurement units. After normalization, the data is transformed into a time series format suitable for the LSTM forecasting model.

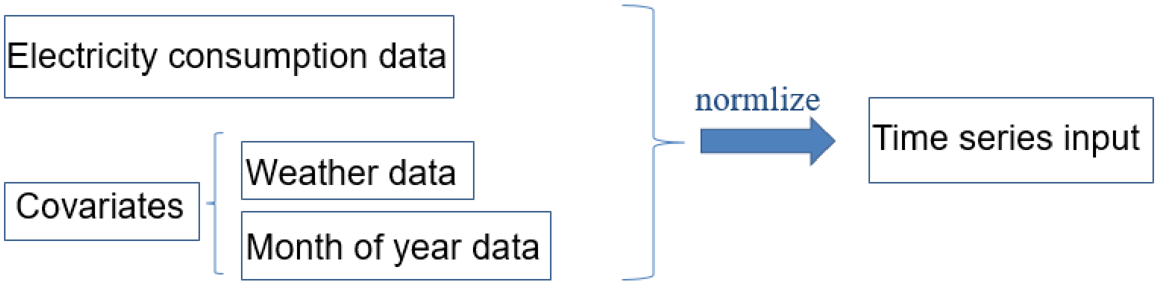


Figure 3.2. Input data pipeline

Scaling data before training the model helps to balance the input values, accelerate convergence, and improve model stability. It reduces the influence of unit differences, avoids numerical precision problems, and makes data processing more efficient, thereby improving prediction accuracy and performance. I will perform data normalization according formula (3.1)

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3.1)$$

Where:

- X is the original value of the data
- X_{min} and X_{max} are the minimum and maximum values in the data
- X_{scaled} is the normalized value, which lies within the range [0; 1]

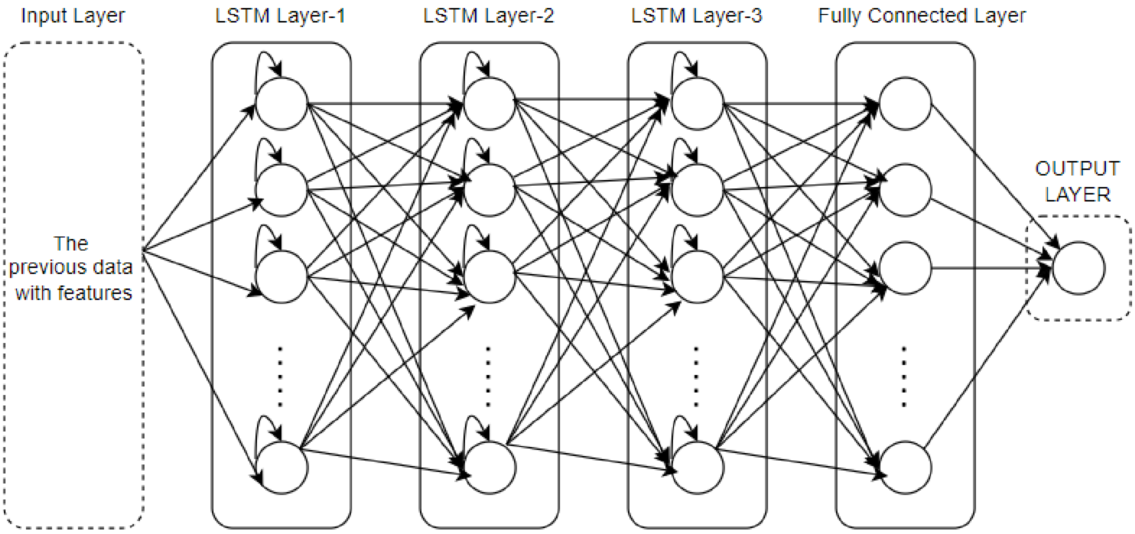


Figure 3.3. The architecture overview of proposed model

Based on the image the architecture overview of proposed LSTM as *Figure 3.3*, here is a detailed analysis of the tasks of each layer and the associated mathematical formulas:

- Input Layer:
  - ➢ Purpose: This layer receives the raw data, formats it, and passes it to the subsequent layers.
  - ➢ Example: For electricity consumption prediction, each time step may include features like temperature, month of year, ...
- Hidden Layers: (LSTM Layers)
  - ➢ Hidden size = 64: Number of LSTM units in each layer (size of the hidden state vector).
  - ➢ Dropout = 0.3: 30% of the units are randomly dropped during training to reduce overfitting.
  - ➢ Purpose: LSTM layer learn temporal relationships between the steps in the time series, storing contextual information through hidden states h_t

and memory states C_t. Stacking multiple LSTM layers helps learn more abstract representations from the input data.

➢ Mathematical Formulation:

Forget Gate: Determines which information from the previous state should be discarded.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t\,] + b_f\right) \qquad (3.2)$$

Input Gate: Decides which information from the current input should be stored.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t\,] + b_i) \qquad (3.3)$$

$$\hat{C_t} = tanh(W_C \cdot [h_{t-1}, x_t\,] + b_C) \quad (3.4)$$

Update Cell State:

$$C_t = f_t * C_{t-1} + i_t * \hat{C_t} \qquad (3.5)$$

Output Gate: h_t is the final hidden state, representing learned information about the input x_t

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t\,] + b_o) \qquad (3.6)$$

$$h_t = o_t * tanh(C_t) \qquad (3.7)$$

- Fully Connected Layer

➢ Output chunk length = 1: Only predicts one value (the next time step).

➢ Output size = 1: The output is a single scalar value (electricity consumption).

➢ Purpose: Aggregates information from the LSTM layers and maps it to a single output value.

➢ Mathematical Formulation:

$$\hat{y} = W_{fc} \cdot h_t + b_{fc} \qquad (3.8)$$

Where:

W_{fc}: Weights of the Fully Connected layer.

b_{fc}: Bias term.

- Output Layer: Generates the final prediction: Outputs a single real-valued number $\hat{y}$.

Summary of model architecture follows *Table 3.2*

Layer Type	Input Shape	Output Shape	Additional Parameters
Input	(batch_size, periodicity, features)	(batch_size, periodicity, features)	Periodicity: number of time steps
Hidden Layer	(batch_size, periodicity, features)	(batch_size, periodicity, 64)	Hidden size = 64, dropout = 0.3
Hidden Layer	(batch_size, periodicity, 64)	(batch_size, periodicity, 64)	Hidden size = 64, dropout = 0.3
Hidden Layer	(batch_size, periodicity, 64).	(batch_size, periodicity, 64)	Hidden size = 64, dropout = 0.3
Fully Connected	(batch_size, periodicity, 64)	(batch_size, 1, 1)	Output chunk length = 1, output size = 1
Output	(batch_size, 1, 1)	(batch_size, 1, 1)	Final output prediction

Table 3.2. Proposed model architecture

However, in the experiment process, I can reselect the best parameters for model.

I propose using the Adam optimizer for the LSTM model in time series forecasting because Adam automatically adjusts the learning rate, allowing the model to converge quickly and stably.

About the loss function, I choose Mean Squared Error (MSE), as it measures the difference between the predicted and actual values, guiding the model to minimize this error during optimization. For the electricity consumption forecasting task, it is commonly used due to its ability to penalize larger errors more severely, making it suitable for time series forecasting problems. The formular of this function shows as (3.2)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (3.9)$$

Where:

- MSE: is mean squared error
- n: is the number of data points
- y_i: is the actual value of electricity consumption at time i
- $\hat{y}_i$: is the predicted value at time i

CHAPTER 4. EXPERIMENTAL RESULT

4.1. Environment setting

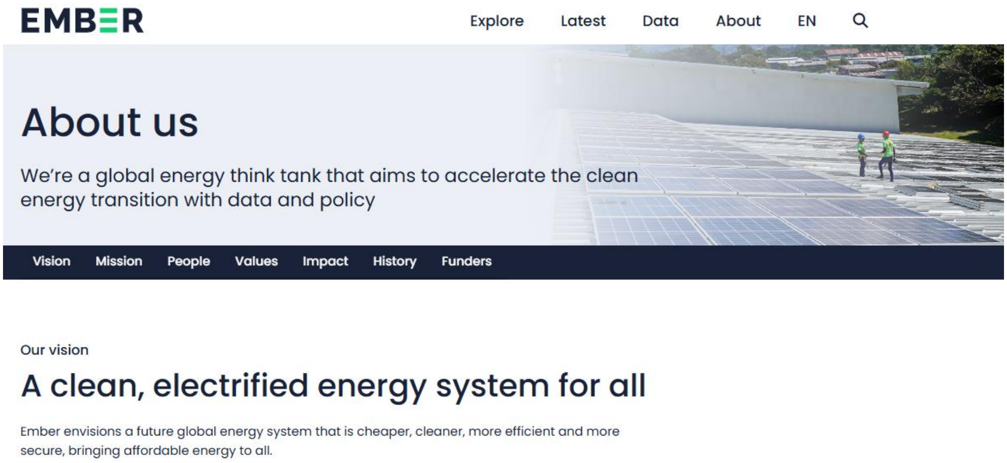
The entire data processing, model training process and model implement are performed on:

- Hardware: Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz 2.20 GHz.
- Operating system: Window 10
- Software: Python 3.11.9, Visual Studio Code editor.

4.2. Data collection and preprocessing

4.2.1. Electricity consumption data of Vietnam

To collect data for training the model to forecast Vietnam's electricity consumption, we need to collect data from a reputable source such as EVN. However, because EVN does not provide this dataset specifically through .csv file but only publishes it through annual and monthly reports, data collection is extremely difficult and requires a long time. So I turned to Ember, a global energy research and advocacy organization. Ember is a reputable non-profit organization, globally recognized for its transparent analytical reports, partnerships with major organizations such as the IEA and the EU, and regularly cited on international media platforms such as the BBC, The Guardian, and Reuters. This organization indirectly collects electricity data through reports from energy organizations around the world, including EVN, thereby synthesizing energy data sets in .csv file format for easy use. Therefore, this is a reputable and reliable data source, so I decided to collect data from this data source.



About us

We're a global energy think tank that aims to accelerate the clean energy transition with data and policy

Our vision

A clean, electrified energy system for all

Ember envisions a future global energy system that is cheaper, cleaner, more efficient and more secure, bringing affordable energy to all.

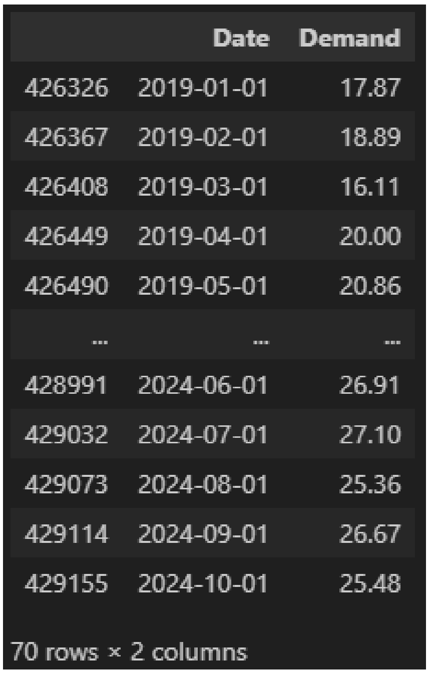
Figure 4.1. Ember's website image

(source: [12])

I have collected Vietnam's electricity consumption data from the Electricity dataset on Ember website. This is energy data of all countries in the world collected on monthly frequency. This dataset includes [13]:

- Electricity generation (TWh), provided both by fuel type and aggregated.
- Electricity net imports (TWh).
- Electricity demand (TWh), calculated as the sum of power production and net imports.
- Installed power generation capacity (GW), broken down by fuel type.
- Emissions from electricity generation (Mt CO2e), calculated from IPCC emissions factors.

To process the data used for model training, only choosing data about Electricity demand. I have extracted Vietnam's electricity data from the data set of all countries in the world with the time starting from January 2019, ending in October 2024 with monthly frequency, then saving data on .csv file.



	Date	Demand
426326	2019-01-01	17.87
426367	2019-02-01	18.89
426408	2019-03-01	16.11
426449	2019-04-01	20.00
426490	2019-05-01	20.86
...	...	...
428991	2024-06-01	26.91
429032	2024-07-01	27.10
429073	2024-08-01	25.36
429114	2024-09-01	26.67
429155	2024-10-01	25.48

70 rows × 2 columns

Figure 4.2. Demand Vietnam electricity

4.2.2. Weather data of Vietnam

In addition, I also collect data that greatly affects the country's electricity consumption such as temperature, humidity, and rainfall to support the prediction. I choose Vietnam Online website to collect weather data.

Vietnam Weather

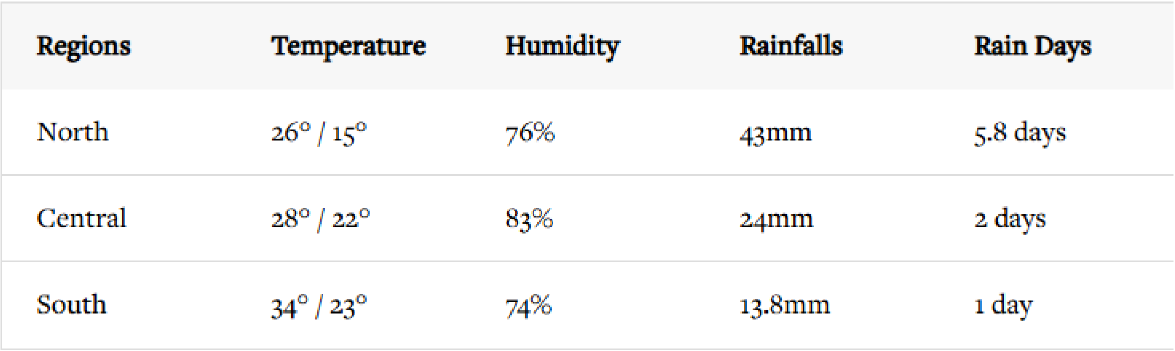
Overview | When to go | Weather by regions | Weather by month | Weather forecast

A weather and climate overview for Vietnam including current weather forecast for major cities and when to visit.

Figure 4.3. Vietnam Online's website image

(source: [14])

This data includes data on temperature limits, average humidity, rainfall, and rainy days of Vietnam by month of the year.

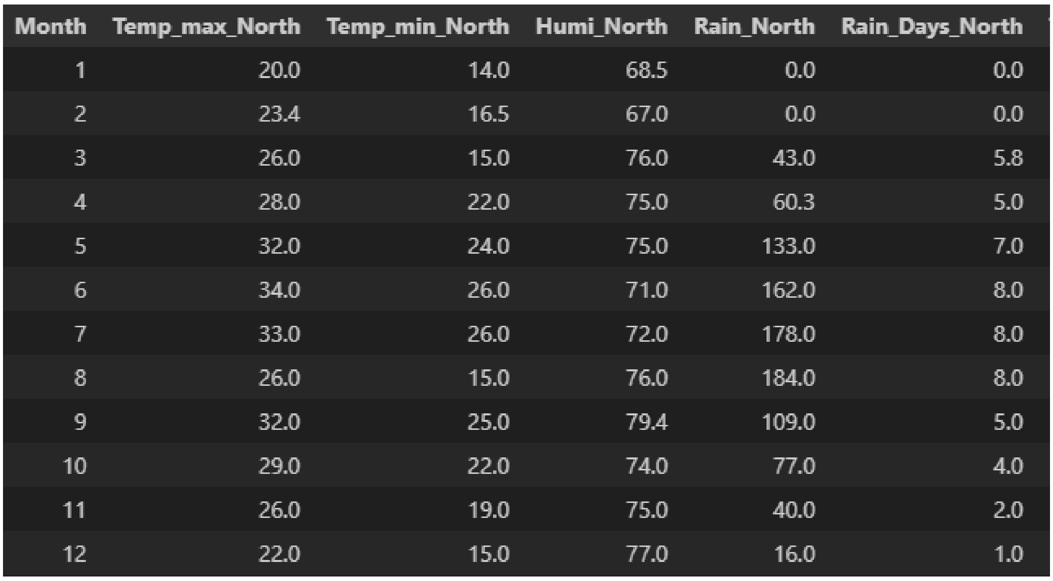


Regions	Temperature	Humidity	Rainfalls	Rain Days
North	26° / 15°	76%	43mm	5.8 days
Central	28° / 22°	83%	24mm	2 days
South	34° / 23°	74%	13.8mm	1 day

Figure 4.4. Vietnam weather data in March

(source: [15])

The data from this website is split into smaller features by month. For example, the data for each month in the North region is divided into: Temp_max_North (maximum temperature), Temp_min_North (minimum temperature), Humi_North (average humidity), Rain_North (rainfalls), Rain_Days_North (rain days). Similar for Central and South region. Then this data is saved on .csv file for data processing and model training.



Month	Temp_max_North	Temp_min_North	Humi_North	Rain_North	Rain_Days_North
1	20.0	14.0	68.5	0.0	0.0
2	23.4	16.5	67.0	0.0	0.0
3	26.0	15.0	76.0	43.0	5.8
4	28.0	22.0	75.0	60.3	5.0
5	32.0	24.0	75.0	133.0	7.0
6	34.0	26.0	71.0	162.0	8.0
7	33.0	26.0	72.0	178.0	8.0
8	26.0	15.0	76.0	184.0	8.0
9	32.0	25.0	79.4	109.0	5.0
10	29.0	22.0	74.0	77.0	4.0
11	26.0	19.0	75.0	40.0	2.0
12	22.0	15.0	77.0	16.0	1.0

Figure 4.5. Vietnam weather data

4.3. Data processing and model training

4.3.1. Data processing

After collecting and preprocessing the data, I loaded the saved .csv files and plotted them on a chart to observe Vietnam's electricity consumption from January 2019 to October 2024 with monthly frequency as shown in *Figure 4.6*.

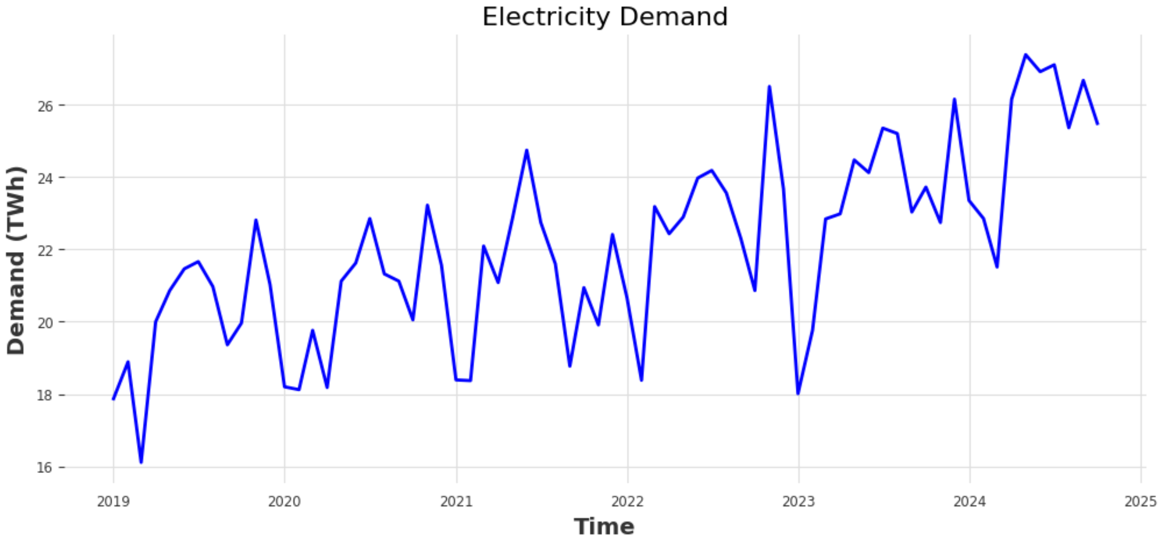


Figure 4.6. Electricity demand graph

I split this dataset into train dataset for training phase and valid dataset for validation phase with proportions of 70%, 30% respectively. The train dataset starts from January 2019 to December 2022, and the valid dataset includes the remaining months in original dataset.

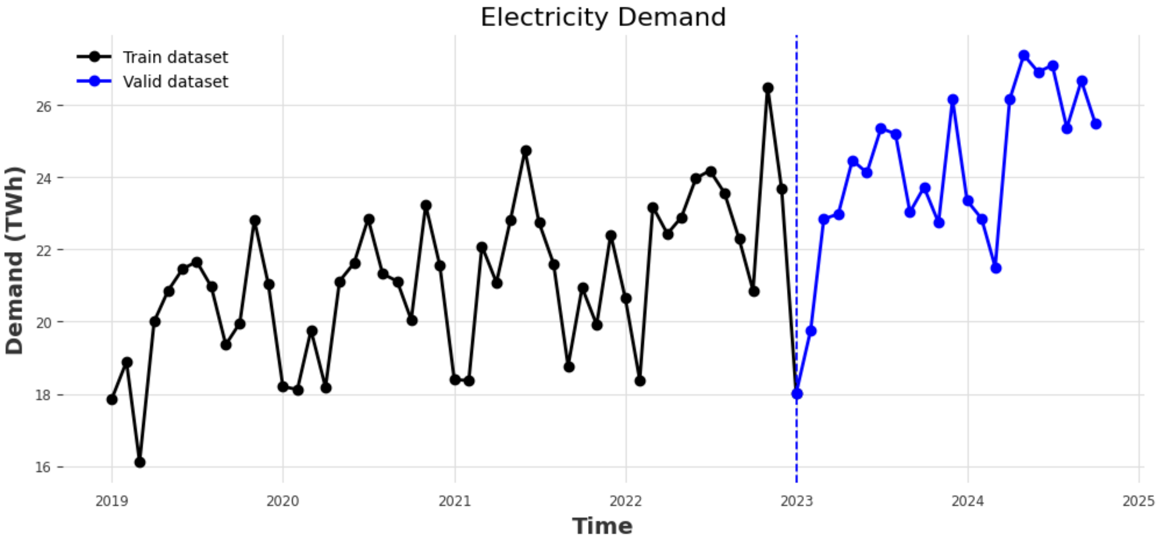


Figure 4.7. The dataset splitting

To avoid revealing valid data set information during model training, I only analyze data on the training data set to understand the data and optimize the parameters in the time series prediction model.

Firstly, I analyze the trend of data as show in *Figure 4.8* . We can see that Vietnam's electricity consumption data tends to increase every month.

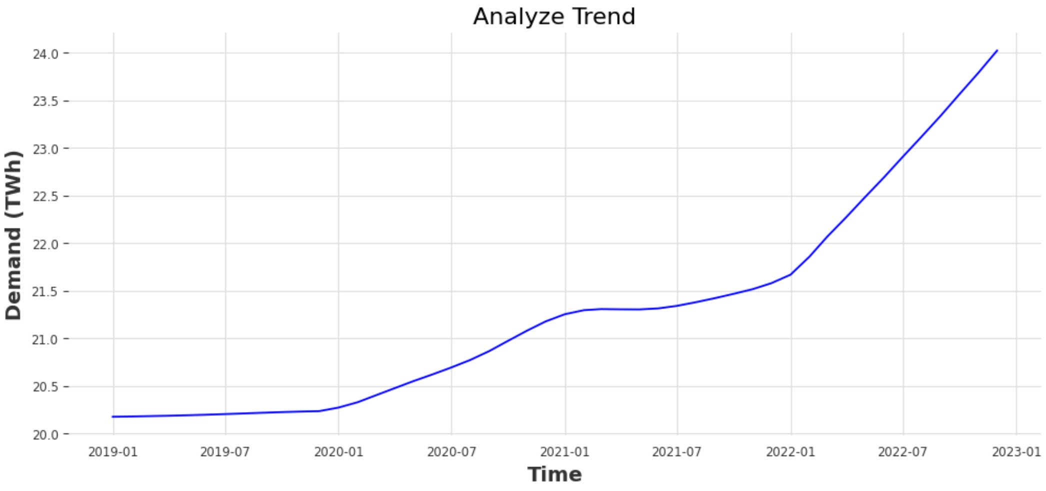


Figure 4.8. The trend analysis

The data shows a steady growth trend in energy demand (Demand) from 2019 to 2022, with clear changes from year to year. From the beginning of 2019 to the end of 2020, energy demand increased slowly and remained at a nearly stable level, showing a low growth rate during this period. However, from 2021 onwards, the trend began to increase more strongly, especially in 2022, demand grew rapidly and clearly, reaching its peak at the end of 2022.

Then, I analyze the seasonality of the data and found that the data is seasonal and has a cycle of 7 months as show in *Figure 4.9* . This is useful for designing forecast models in terms of the number of inputs and outputs for prediction.

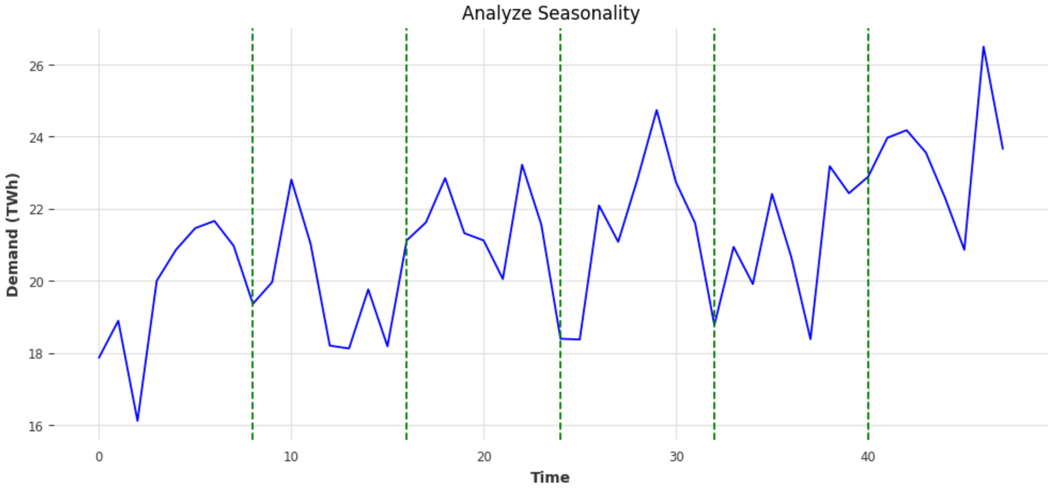


Figure 4.9. The seasonality analysis

In addition to the main data on electricity consumption in each month, I also incorporate auxiliary variables, and also normalize them and incorporate them into the main data. They include:

- **Weather data** for each month in Vietnam, which I have collected and preprocessed. These weather variables are important because they can have a direct impact on electricity consumption, such as increased energy use for cooling during hot months or heating during colder periods, as well as variations in energy demand based on weather conditions.
- **Time attribute covariates**: These include information derived from the timestamps, such as the month of the year, encoded using one-hot encoding. This representation allows the model to identify seasonal patterns or periodic trends in electricity consumption that correlate with specific months.

4.3.2. Evaluation metrics model

To evaluate the performance of the model, I choose commonly used model evaluation metrics for time series prediction including: Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), R-squared.

MAPE is a common metric in regression and time series forecasting. It calculates the average absolute percentage difference between the actual values and the predicted values. The lower the MAPE, the better the model fits the data. To calculate MAPE, I use the following formula:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \quad (4.1)$$

Where:

- $MAPE$: is mean absolute percentage error
- n: is the number of data points
- y_i: are observed values
- $\hat{y}_i$: are predicted values

RMSE measures the square root of the average squared differences between predicted values and actual values. A lower RMSE indicates better model performance. To calculate RMSE, I use the following formula:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (4.2)$$

Where:

- $RMSE$: is root mean squared error
- n: is the number of data points
- y_i: are observed values
- $\hat{y}_i$: are predicted values

R-squared, or the coefficient of determination, is a statistical measure that indicates the proportion of variance in the dependent variable that is explained by the independent variables in a model. An R-squared value closer to 1 indicates a better fit of the model to the data. To calculate R-squared, I use the following formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (4.3)$$

Where:

- R^2: is the coefficient of determination
- y_i: are observed values
- $\hat{y}_i$: are predicted values
- $\bar{y}$: is the mean of the observed values

4.3.3. Model training

Through the process of analyzing and processing data, I have determined that the data cycle is 7 months. Therefore, the input and output design of the model for predicting electricity consumption will be as shown in *Figure 4.10.*

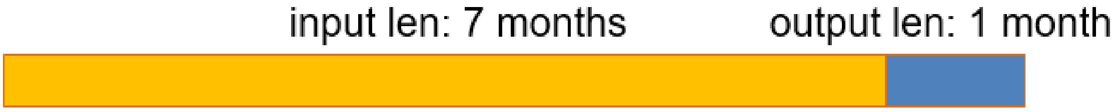


Figure 4.10. Design prediction intervals

So my prediction model will predict the electricity consumption of a month in the future using the model with the pre-trained weights and the input will be the data of the previous 7 months. At the same time, as I analyzed above, coraviates will also be added to the model to improve the accuracy of electricity prediction.

First, I will train the data with the proposed power prediction models in the theoretical basis section to find the best model based on the evaluation indexes MAPE, RMSE and R-squared. By dividing the data set into two parts training and validation above, I will train the models with the training set and evaluate the indexes of the model performance on the valid set. After the training process, the prediction on the valid set will also follow the principle as shown in *Figure 4.13.* However, when the model makes predictions for more than 1 output, the input and the previously predicted output will be the input for the prediction and continue similarly for the prediction of the final output.

The models that I will use for training and testing to choose the optimal model include: LSTM, GRU, RNN, TCN and ARIMAX. I use the Darts library for designing models with parameters tuned to fit the data. Darts is a user-friendly Python library for forecasting and anomaly detection on time series. It contains many different models, from classic models like ARIMA to deep neural networks [16]. The library has integrated models with the architecture as the theoretical basis I have given, the current work is to fine-tune to find the parameters for the models to perform best on my dataset. Below are the parameters for each model that I have fine-tuned.

Model LSTM:

- Number of layers: 1
- Number of neurons in each layer: 175
- Dropout: 0.3
- input_chunk_length: 7
- output_chunk_length: 1
- Activation function: Tanh, Sigmoid
- Optimizer: Adam
- Loss function: Mean Squared Error
- Epochs: 184
- Batch size: 64
- Leaning rate: 0.001

Model GRU:

- Number of layers: 1
- Number of neurons in each layer: 150
- Dropout: 0.4
- input_chunk_length: 7
- output_chunk_length: 1
- Activation function: Tanh, Sigmoid
- Optimizer: Adam

- Loss function: Mean Squared Error
- Epochs: 200
- Batch size: 64
- Leaning rate: 0.001

Model RNN:

- Number of layers: 1
- Number of neurons in each layer: 170
- Dropout: 0.2
- input_chunk_length: 7
- output_chunk_length: 1
- Activation function: Tanh, Sigmoid
- Optimizer: Adam
- Loss function: Mean Squared Error
- Epochs: 150
- Batch size: 64
- Leaning rate: 0.001

Model TCN:

- Number of layers:
- Kernel size: 5
- Number of filters: 3
- Dilation rate: 3
- Dropout: 0.01
- input_chunk_length: 7
- output_chunk_length: 1
- Number of neurons: 1 output neuron
- Activation function: ReLU
- Optimizer: Adam
- Loss function: Mean Squared Error
- Epochs: 199

- Batch size: 16

Model ARIMAX:

- (p, d, q) = (0, 1, 0)
- Loss function: Mean Absolute Percentage Error

After training, we get the performance metrics of the models as *Table 4.1.*

Metric	GRU	LSTM	RNN	TCN	ARIMAX
MAPE	7.890920	**4.509252**	9.664353	11.561025	8.463822
RMSE	2.089941	**1.468630**	2.502647	2.929865	2.239708
R-squared	0.197653	**0.603796**	-0.150518	-0.576847	0.078538

Table 4.1. Model metrics comparison

With the statistics in the table above, I have the following comments:

- **LSTM**: It has the lowest MAPE and RMSE, and the highest R-squared (0.603796). LSTM is the most effective model in forecasting power consumption.
- **GRU**: Second best performance with MAPE of 7.89 and R-squared of 0.197653. Lower than LSTM but still better than the rest of the models.
- **RNN**: It gives worse results with high MAPE (9.66) and negative R-squared (-0.150518), indicating it is not suitable for this data.
- **TCN**: The worst performance with MAPE is 11.56 and the most negative R-squared (-0.576847), indicating that the model is not suitable for this data.
- **ARIMAX**: As a traditional model, it gives average results with MAPE of 8.46 and R-squared close to 0 (0.078538), showing that the model does not describe the relationship in the data well.

Through this, I have verified that LSTM is the best model for the final training of the Vietnam electricity consumption forecasting model.

At this stage, I no longer divide the dataset into 2 sets including the train and valid sets, but divide the dataset into 3 sets: train (70%), valid (15%), test (15%) for the processes of training the model, valid the model and evaluate the model on a data set that is completely hidden during the training process.

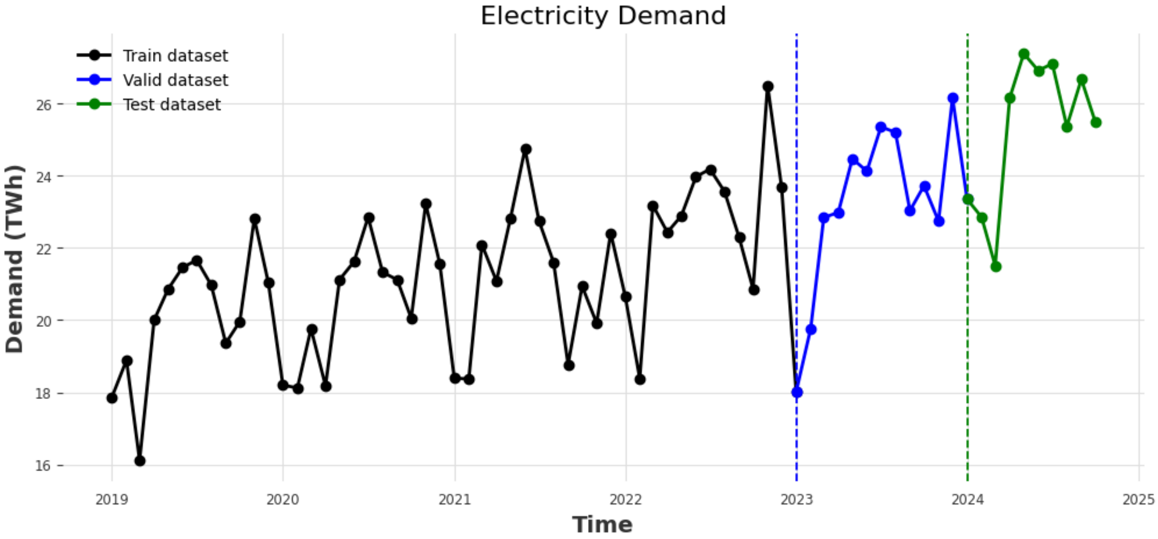


Figure 4.11. The dataset splitting for LSTM model training

In this phase, I will fine-tune the parameters of the LSTM model so that it learns effectively on the new validation set. The best parameters of the LSTM model are found through the Grid Search method as follows:

- Number of layers: 1
- Number of neurons in each layer: 175
- Dropout: 0.2
- input_chunk_length: 7
- output_chunk_length: 1
- Activation function: Tanh, Sigmoid
- Optimizer: Adam
- Loss function: Mean Squared Error
- Epochs: 184
- Batch size: 64
- Leaning rate: 0.001

Next, we will evaluate the change in prediction error of the entire training and validation sets over each epoch and the performance of the final trained model applied to the validation set.

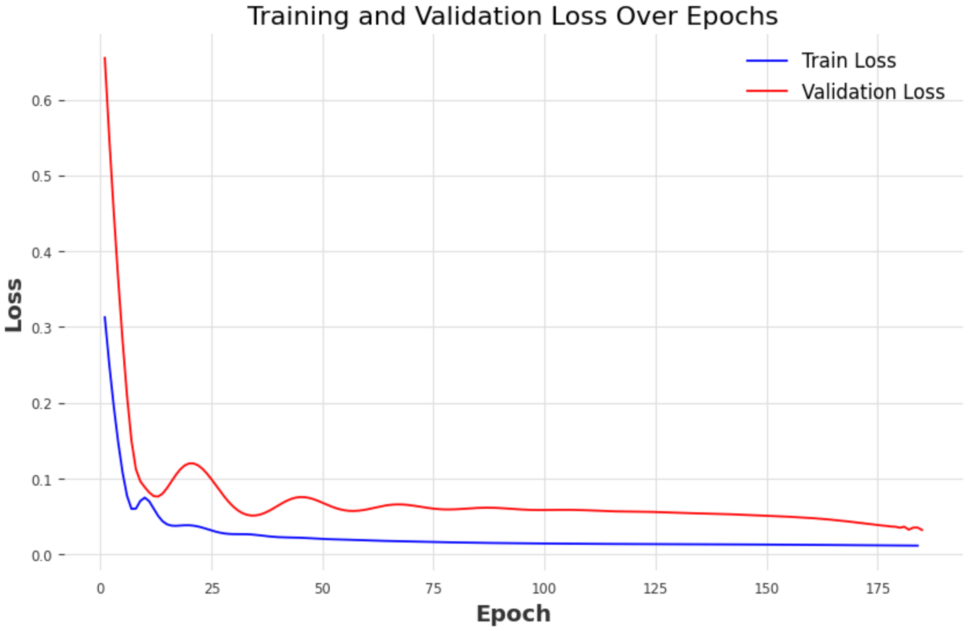


Figure 4.12. Training and Validation loss over epochs

The graph in *Figure 4.12* shows that the loss on both sets has converged, which proves that the training stop of the model with the parameters and number of epochs is reasonable.

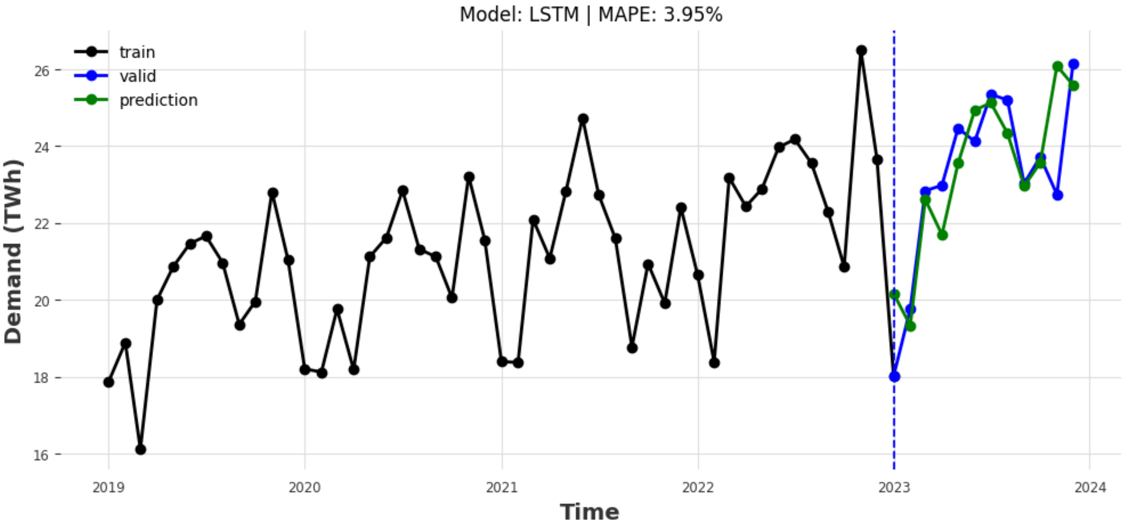


Figure 4.13. The performance of model on validation dataset

The graph in Figure 4.13 shows that the trained LSTM model predicted quite accurately on the validation set with a relatively small percentage of error, only 3.95%.

Finally, it is time to test the model's performance on a test dataset. Unlike the validation set, the test set does not reveal the model's performance information during training. Therefore, testing the prediction on this dataset will be the most intuitive. However, the AI model training is being performed on time series data, so it is necessary to load the validation set into the trained model to retrain the model for predicting the test dataset.

Here are the model evaluation metrics for predicting the test set:

- MAPE : 7.1826
- RMSE : 1.8315
- R-squared : 0.0924

The evaluation results of the LSTM model show that MAPE = 7.18% and RMSE = 1.83, reflecting a relatively small forecast error and acceptable model accuracy. However, R-quared = 0.0924 is quite low, indicating that the model only explains about 9.24% of the variance of the data and does not capture the entire variation in the test set well. This limitation may be due to the incomplete addition of coraviates for the model to forecast more accurately.

The graph in *Figure 4.14* visually describes the model's performance on the test set.

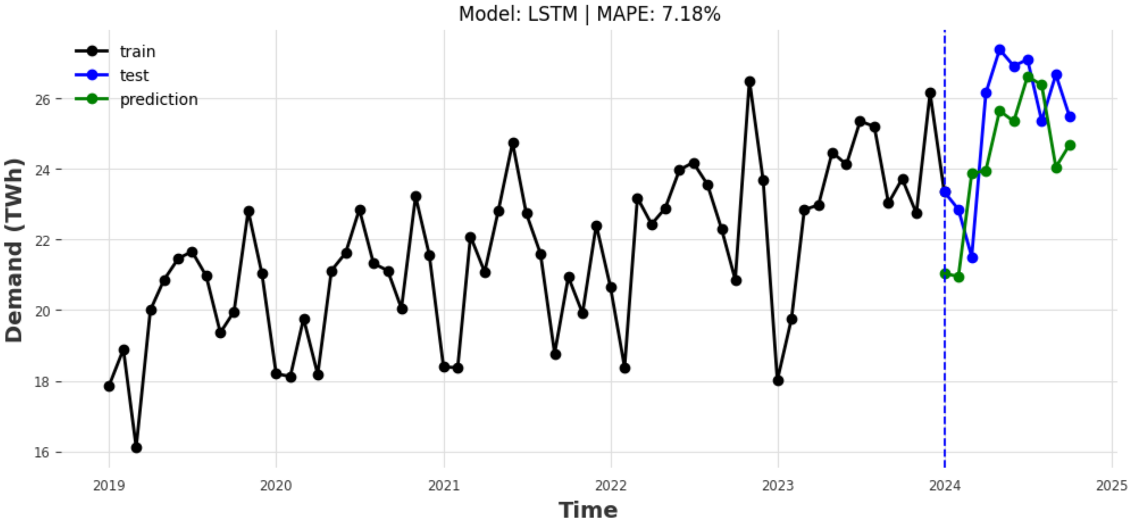
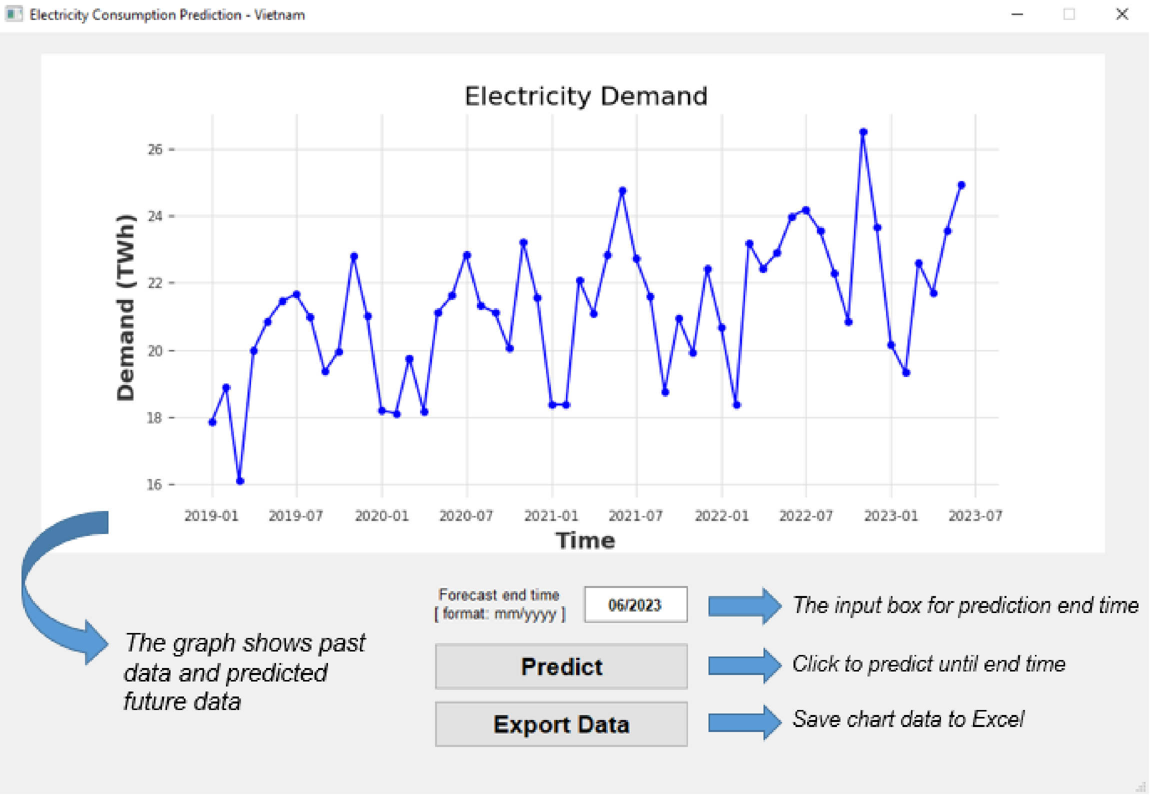


Figure 4.14. The performance of model on test dataset

4.4. Implement model on GUI

After training and testing the best model, I save the model weights and use them to deploy on a GUI software that allows users to observe the model's predictions. As

I proposed the method of creating GUI software for deploying AI models in the theoretical basis section, I will continue to use QT Designer for this work. The interface of GUI software is shown in *Figure 4.15*.



Forecast end time
[format: mm/yyyy] 06/2023

The input box for prediction end time

Click to predict until end time

Save chart data to Excel

The graph shows past data and predicted future data

Figure 4.15. GUI software interface

According to the above interface, the program has the following main features:

- Displays the power consumption chart (including known time and predicted time).
- Allows users to enter the predicted end month in the required format.
- Performs power consumption prediction with the integrated LSTM model (Predict button).
- Exports the power consumption chart data to an Excel file (Export Data button).

Demo of program operation as shown in *Figure 4.16.*

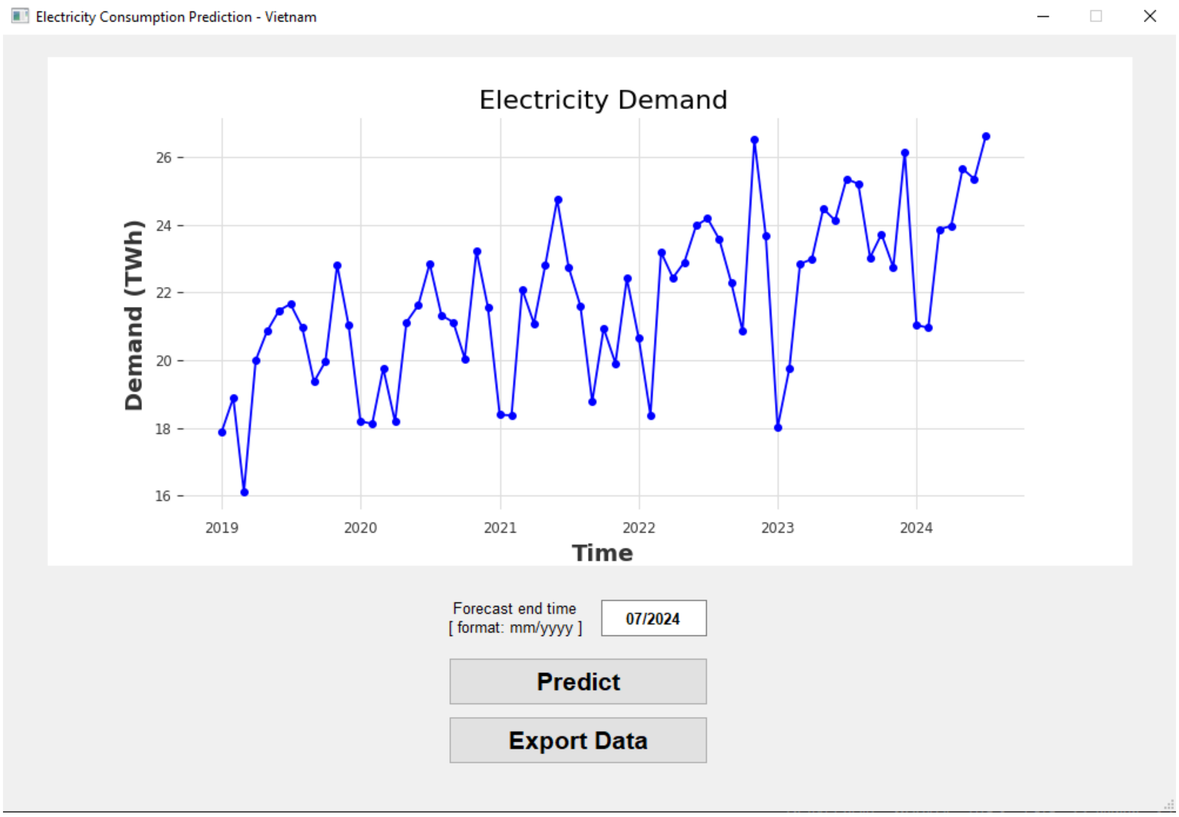


Figure 4.16. Demo of the program operation

After the user enters the final time for Vietnam's electricity consumption prediction in July 2024 and presses the "Predict" button, the integrated LSTM model will calculate to predict the future electricity consumption values. After the calculation is complete, the graph showing the known values and the predicted values will be shown. The user can press the "Export Data" button to save the values shown in the graph to an Excel file to store the information and serve as a basis for evaluating the performance of the model when the predicted values will be known in reality in the future.

4.5. Limitation

Using input data including historical electricity consumption, monthly index data and monthly weather data, the performance of the trained LSTM model to forecast Vietnam's electricity consumption is not really high. This is due to the lack of external data that greatly affects electricity consumption such as urbanization, the performance of people's electricity-consuming electronic devices, fluctuations in production, etc.

CHAPTER 5. CONCLUSION AND RECOMMENDATION

Thus, with the proposed plan of using LSTM model to predict Vietnam's electricity consumption along with data collection, processing and model training and evaluation methods, I have successfully trained the model with MAPE of 7.18% (under 10%). In addition, the GUI software that allows users to view the model's predictions has also been built according to design requirements.

However, the project still has some problems. One of them is that the collection of data that greatly affects electricity consumption is limited when there is only data on Vietnam's weather by month. Therefore, to improve this project in the future, I need to collect and process other feature data on urbanization, population growth or the performance of electrical equipment to be able to include in training to help the model predict more accurately.

REFERENCES

[1] "Ảnh hưởng do mất điện: Sản xuất nông nghiệp gặp 'khó'," *Báo Hànộimới*, Jun. 15, 2023. [Online]. Available: https://hanoimoi.vn/anh-huong-do-mat-dien-san-xuat-nong-nghiep-gap-kho-621525.html.

[2] E. O. Edoka, V. K. Abanihi, H. E. Amhenrior, E. M. J. Evbogbai, L. O. Bello, and V. Oisamoje, "Time Series Forecasting of Electrical Energy Consumption Using Deep Learning Algorithm," *Nigerian Journal of Technological Development*, vol. 20, 2023.

[3] R. Djimasbe, S. Gyamfi, C. D. Iweh, and B. N. Ribar, "Development of an ARIMAX model for forecasting airport electricity consumption in Accra-Ghana: The role of weather and air passenger traffic," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 9, 2024.

[4] B. R. Prasad, D. Siddaiah, T. A. Bakerel-ebiary, S. N. Kumar, and K. Selvakumar, "Forecasting electricity consumption through a fusion of hybrid random forest regression and linear regression models utilizing smart meter data," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 21, 2023.

[5] D. Hadjout, J. F. Torres, A. Troncoso, A. Sebaa, and F. Martínez-Álvarez, "Electricity consumption forecasting based on ensemble deep learning with application to the Algerian market," *Energy*, vol. 243, 2021.

[6] Y. Li, "Energy consumption forecasting with deep learning," *Journal of Physics: Conference Series*, vol. 2711, 2021.

[7] H. Y. Lee, K. M. Jang, and Y. Kim, "Energy Consumption Prediction in Vietnam with an Artificial Neural Network-Based Urban Growth Model," *Energies*, vol. 13, no. 17, pp. 4282, 2020.

[8] N. Islam, "Comprehensive Guide to Time Series Data Analytics and Forecasting with Python," *Medium*, Jul. 27, 2024. [Online]. Available: https://medium.com/@nomannayeem/comprehensive-guide-to-time-series-data-analytics-and-forecasting-with-python-2c82de2c8517.

[9] "Stock Price Forecasting," *Medium*, Oct. 9, 2019. [Online]. Available: https://medium.com/analytics-vidhya/stock-price-forecasting-i-analysing-time-series-e7157563cbce.

[10] "Pandas DataFrame," *GeeksforGeeks*, Nov. 28, 2024. [Online]. Available: https://www.geeksforgeeks.org/python-pandas-dataframe/.

[11] R. Hu and T. Luo, "LSTM model," *ResearchGate*. [Online]. Available: https://www.researchgate.net/publication/377180235/figure/fig2/AS:11431281215931394@1704474373244/LSTM-model-There-are-several-essential-notations-important-to-understanding-this-model.png.

[12] "About us," *Ember*. [Online]. Available: https://ember-energy.org/about/.

[13] "Methodology," *Ember*. [Online]. Available: https://ember-energy.org/data/monthly-electricity-data/.

[14] "Weather by month," *Vietnam Online*. [Online]. Available: https://www.vietnamonline.com/weather.html.

[15] "Temperature and humidity," *Vietnam Online*. [Online]. Available: https://www.vietnamonline.com/weather/march.html.

[16] "Time Series Made Easy in Python," *Darts*. [Online]. Available: https://unit8co.github.io/darts/.