Day-Ahead Electricity Prices Prediction in France Using Deep Learning

Mengou Zhao

A project report submitted to the Auckland University of Technology
in partial fulfilment of the requirements for the degree of
Master of Computer and Information Sciences (MCIS)

2023

School of Engineering, Computer & Mathematical Sciences

Abstract

In the prediction of electricity prices, machine learning has proven to be significant and meaningful. The algorithms including Support Vector Regression (SVM), Convolutional Neural Networks (CNN), and other statistical-based methods and deep learning algorithms are being widely experimented. In this project report, we propose to use Long Short Term Memory (LSTM) algorithm, Autoregressive Integrated Moving Average (ARIMA), and Vector Autoregressive Moving Average with Exogenous Regressors (VARMAX) to predict the day-ahead electricity prices in France and compare the performance of these three algorithms. We introduce details on research design, research methods, research resources, and result comparisons. We achieved a mean squared error 29.24 and a root mean squared error of 5.24 using the method we proposed. Additionally, we recommend using a 30-day training dataset for daily operations, as this produced a mean squared error of 3.47, a mean absolute percentage error 0.78, and a root-mean-square error 1.35 in predicting the following 24-hour day-ahead electricity prices, which significantly improved the accuracy.

Keywords: LSTM, ARIMA, VARMAX, Day-ahead electricity price

Table of Contents

List of Figures

List of Tables

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgments), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature: Date: 2023

Acknowledgment

I would like to express my gratitude to my husband for his unwavering spiritual support. He always manages to lift my spirits and remains patient even when I am stressed. Additionally, he has taken excellent care of our furry babies, Archie the cat and April the dog, while I am fully immersed in this project.

I would also like to express my deepest gratitude to my primary supervisor Dr Wei Qi Yan. His one-of-a-kind teaching approach and strict standards motivated me to complete the project with both excellence and efficiency. Moreover, he imparted the state-of-the-art techniques that have proven beneficial in the industry. I believe I could not complete my study without Dr Yan's supervision and instructions.

Mengou Zhao

Auckland, New Zealand

2023

Chapter 1
Introduction

This chapter is composed of five parts: The first part introduces the background and motivations, the second part includes the research question, followed by the contributions, objectives, and structure of this report.

1.1 Background and Motivation

In today's world, Internet of Things (IoT), sensor networks (RFID), and artificial intelligence (AI) (Yamin, 2019) are essential to our daily lives. To enpower these technologies, a few sources of energy have become necessary, without which our lives would come to a halt. Electricity is one such source. Electricity was first discovered through natural phenomena like lightning and friction (Forrester, 2016). With the invention of the electric generator and the application of various electric devices for commercial purposes, the electricity industry was established.

The prediction of electricity prices helps the suppliers and the retailers to achieve a higher profit as well as to deliver a better service to the customers. However, the electricity market has shown a bunch of uncommon features compared to other markets. According to the research, supply and demand conditions (Vahidinasab, Jadid, & Kazemi, 2008), date-related effectors (Lago, De Ridder, Vrancx, & De Schutter, 2018; De Schutter, 2018), weather conditions, and storage availability (Cartea & Figueroa, 2005) all affect the electricity prices.

Machine learning methods are introduced in predicting diverse problems. Among which statistical based algorithms are wildly used in time series problems. However, the traditional machine learning algorithms are hard to perform on complex problems such as the electricity prices prediction. Therefore, deep learning techniques are introduced. Deep learning allows software to learn how to perform tasks on its own, and has proven to be advantageous when dealing with non-linear problems (Parloff, 2016). ChatGPT, which was recently released, has made transfer learning techniques popular. By transferring known knowledge to unfamiliar problems, these techniques can significantly reduce the need for data collection and labelling, as well as re-training efforts.

In this project report, we seek a deep learning based method to predict day-ahead electricity prices in France in a high-efficiency and high-quality manner.

1.2 Research Questions

The research questions of this report are,

(1) Which machine learning algorithm is able to perform the day-ahead electricity prices prediction in France with high accuracy?

(2) Does deep learning based algorithm perform better than the statistical algorithms with the day-ahead electricity prices prediction in France?

(3) How can we accurately predict the day-ahead electricity prices in a way that is practical and beneficial for operations?

This project aims to predict day-ahead electricity prices in France using deep learning techniques. After carefully selecting the appropriate methods, we conducted experiments and adjusted the hyperparameters to improve the accuracy of the predictions. Moreover, we took the daily operation of electricity providers into consideration, proposed, tested, and recommended a best practice for predicting the following 24-hour day-ahead electricity prices in France. In the end, we applied transfer learning techniques to predict the day-ahead electricity prices in other EU members states. Our hard work paid off - we created a highly accurate prediction model.

1.3 Contributions

The focus of this project is on predicting the day-ahead electricity prices in France based on deep learning. This method proves to be more efficient and effective compared to traditional statistical methods. By the end of the report, we were able to (1) train the model with minimal data; (2) use LSTM with the best hyperparameters; (3) generate day-ahead electricity prices for the next 24 hours in France; and (4) apply the pre-trained model to predict the day-ahead electricity prices in other EU members states, including Germany, Belgium and the United Kingdom.

1.4 Objectives of This Report

Initially, we compare and review various methods for predicting electricity prices. Next, we examine the patterns exhibited by day-ahead electricity prices in France. Afterwards, we build a deep learning model that is both efficient and effective in predicting day-ahead electricity prices in France and compare the resulting model to the traditional statistical model.

1.5 Structure of This Report

The structure of this report is described as follows:

- Chapter 2: Literature review and discussion of relevant studies on the European electricity markets, the day-ahead electricity prices, Long Short Term Memory, Auto-Regression Integrated Moving Average, and transfer learning.
- Chapter 3: Introduction of research methods, which includes experimental design and methodologies.
- Chapter 4: Implementation of proposed algorithms, collection of experimental data, and demonstration of research outcomes through figures and tables. Limitations of the proposed methods will also be detailed.
- Chapter 5: Summary and analysis of experimental results.
- Chapter 6: Conclusion and statement of future work.

Chapter 2
Literature Review

The focus of this report is on day-ahead electricity prices prediction in France using deep learning, this chapter will introduce a plenty of traditional methods and the relevant knowledge of deep learning.

2.1 Introduction

A time series refers to a set of observations that are recorded over a period of time in a sequential manner (Chatfield, 2000). Time series data is characterised by large data size, high dimensionality, and the need for continuous updates (Fu, 2011). Day-ahead electricity prices can be categorised as a time series problem. There are a couple of various algorithms and methods that have been experimented with in the day-ahead electricity prices prediction.

2.2 Day-ahead Electricity Price

In the recent couple of decades, with the liberalisation of electricity markets (Fanone, Gamba, & Prokopczuk, 2013) (Lago et al., 2018), electricity prices are more market-oriented. The supply and demand conditions became the drive of the price changes. In addition, with the development of renewable energy generated electricity, the electricity providers must maintain a balance between generation and consumption(Ahmad, Zhang, & Yan, 2020). Electricity providers have emphasised the importance of electricity price prediction due to these factors.

The prediction of electricity prices can be categorised into three types depending on the time interval. They are short-term forecasting, medium-term forecasting, and long-term forecasting (Zahid et al., 2019). Short-term forecasting is mostly used because it gives better accuracy of prediction as compared to others. Short-term forecasting analyses the hourly or minute electricity prices. Among which the day-ahead price prediction being the most common. Day-ahead price is the price at which electricity is bought and sold in the wholesale market for delivery the next day.

2.3 The European Electricity Markets

The Target Electricity Model (TEM) was rolled out by the end of 2015 by the European Union for further integrating EU electricity markets. Improving the efficiency of cross-border trade over interconnectors is a crucial component of TEM (Newbery, Strbac, & Viehoff, 2016). This can lead to more efficient use of generation capacity, resulting in fewer instances of large idle capacity, which can result in significant cost savings, particularly during peak periods which may vary between member states. Consequently, the integration of electricity markets in Europe can lead to significant improvements in efficiency and welfare for consumers and industries in the region (Böckers, Haucap, & Heimeshoff, 2013).

France, along with Germany, Belgium, and the United Kingdom, is a member state of the European Union. As a result, there is potential for interconnections and cross-border trade between these member states in Europe.

2.4 Auto-Regression Integrated Moving Average

Traditionally, various statistical methods perform a crucial role in the prediction of time series data. Moving Average (MA) calculates the discrepancies or mistakes from previous sets of data and using them to determine the upcoming or current values in the sequence, Autoregressive (AR) calculates the regression of previous series and predicts the present or future values in the series, and Auto-regression Integrated Moving Average (ARIMA) combines the MA and AR models and a differencing pre-processing step to the sequence in order to make it stationary (Fuller, 2009).

We explain how each observation is influenced by the p previous observations using AR. When $p = 1$,AR can be explained as,

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + e_t \qquad (2.1)$$

where at time t, Y_t represents the current observed value, while Y_{t-1} denotes the previous observed value at time $t-1$, e_t represents a random error, while c and ϕ_1 are both constants. If the value of p is greater than one, more observed values of the series can be included on the right side of the equation. When $d = 0$ or by modelling the differences if $d = 1$ or $d = 2$ between consecutive observations. In practice, the value of d is seldom greater than 2. We explain how every observation is influenced by the previous q errors using MA. If $q = 1$, MA can be explained as,

$$Y_t = c + \theta_1 e_{t-1} + e_t \tag{2.2}$$

where at time t, e_t represents the random error while e_{t-1} refers to the previous random error at time $t-1$. If q is greater than one, additional errors may be added on the right-hand side of the equation.

If the three parts are combined, it results in a wide variety of ARIMA models (Hyndman, 2001). The ARIMA is one of the most widely used methods due to its outstanding performance in the accuracy (Siami-Namini, Tavakoli, & Namin, 2018) (Karabiber & Xydis, 2019), and has been performed well for quite a long time. This method models the next step in the sequence as a linear function of the differenced observations and residual errors at prior time steps.

2.5 Long Short-Term Memory

Deep learning is more and more popular and have been one of the hottest topics to research due to their significant high performance. Recurrent Neuron Networks (RNN) utilised the sequential information in the networks (Pouyanfar et al., 2018). It has repeated layers to feed signals from previous timesteps back into the network. As Figure 2.1 shows in an RNN unit, the input x_t and the output h_t are interconnected. The arrows looping back in the RNN unit indicate that the current output is stored as the next input (Sherstinsky, 2020).

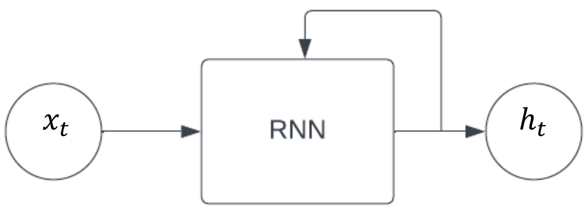

Figure 2.1 The unit of RNN

The RNN networks can be illustrated by Figure 2.2. This approach takes into account not just the effect of the previous state on the current condition; but also, the influence of all the past states on the current one. Therefore, it performs well on sequential data.

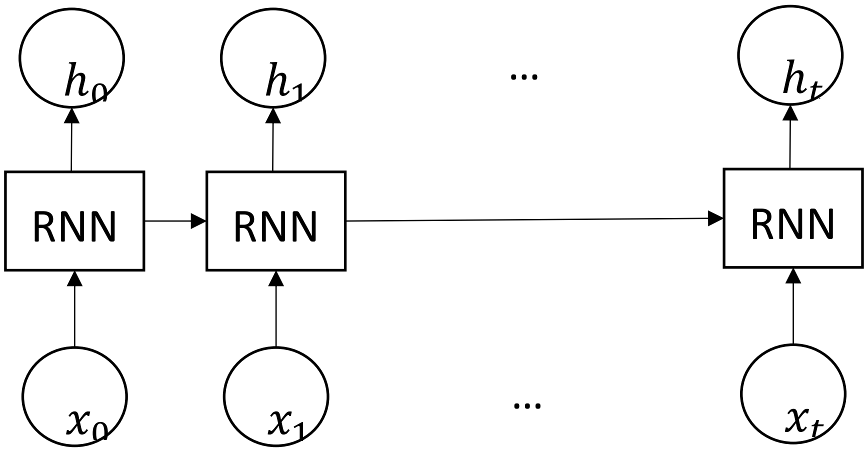

Figure 2.2 RNN networks

However, RNNs are limited to looking back in time for approximately ten timesteps therefore it is called short-term memory. To tackle this, Long-short term memory is introduced with extended plausible and capable to handle more than a thousand timesteps (Staudemeyer & Morris, 2019).

2.6 Prediction on Day-ahead Electricity Prices

Machine learning methods have been employed to forecast day-ahead electricity prices, as evidenced by the comparison chart in Table 2.1.

An experiment was conducted on the electricity prices of Pennsylvania-New Jersey-Maryland (PJM) using Artificial Neural Networks, resulting in a 1.97 mean absolute error. In comparison with other algorithms such as ARIMA, the ANN produced more precise results by achieving a mean absolute percentage error of 6.42, whereas ARIMA reached 11.94 (Vahidinasab et al., 2008).

ARIMA (Karabiber and Xydis published, 2019) was utilised to forecast the day-ahead electricity prices in Denmark's western region. When compared to Trigonometric Seasonal Box-Cox Transformation with ARIMA residuals Trend and Seasonal Components (TBATS) and Artificial Neural Networks, ARIMA produced the most accurate results with a mean absolute error of 33.24 and a root mean squared error of 39.53 (Karabiber & Xydis, 2019).

Various models were tested on day-ahead prices, including Multilayer Perceptron

(MLP), Enhanced Convolutional Neural Networks (ECNN), and Random Forest (RF). Results indicate that the ECNN model demonstrated exceptional accuracy, achieving a 0.99 mean absolute error and 0.99 root mean squared error (Vahidinasab et al., 2008).

Experiments were conducted on PJM billing data (Kuo and Huang, 2018) using various algorithms. The combined use of CNN and LSTM (EPNET) resulted in an impressive 8.84 mean absolute error and a 17.9 root mean squared error (Kuo & Huang, 2018).

In 2019, a bidirectional long short-term memory (BRIM) model was created and evelauted for predicting day-ahead electricity prices in Germany (Chen, Wang, Ma, & Jin, 2019). The model achieved a 6.29 mean absolute error and 15.56 mean absolute percentage error (Chen, Wang, Ma, & Jin, 2019).

Gunduz, et al created an ANN transfer model to forecast day-ahead electricity prices in different European countries, such as France, Belgium, Germany, and Nord Pool. The model has been optimised and produced an mean absolute error of 4.21, an mean absolute percentage error of 12.29, and an root mean squared error of 7.98 specifically in France (Gunduz, Ugurlu, & Oksuz, 2023).

After compared previous studies, it was discovered that deep learning techniques have been extensively tested for predicting day-ahead electricity prices. However, the results have shown that the prediction may not be stable or accurate enough. The objective of this study is to find a method that can achieve stability and accuracy simultaneously.

Table 2.1 Comparison of previous studies

Data set	Algorithm	Mean Absolute Error (MAE)	Mean Absolute Percentage Error (MAPE)	Root Mean Square Deviation (RMSE)	Author and Year
Prices in Pennsylvania–New Jersey–Maryland in 2002	ANN ARIMA	1.97	6.42 11.94		(Vahidinasab et al., 2008)
Day-ahead electricity price	TBATS ANN	37.51 41.41		45.01 48.41	(Karabiber & Xydis, 2019)

Data set	Algorithm	Mean Absolute Error (MAE)	Mean Absolute Percentage Error (MAPE)	Root Mean Square Deviation (RMSE)	Author and Year
in Denmark-West region	ARIMA	33.24		39.53	
Day-ahead electricity price from 1 November 2016 to 15 January 2017	MLP ECNN (Enhanced CNN) RF	1.43 0.99 1.91	4.9 18.18 7.31	1.19 0.99 1.38	(Zahid et al., 2019)
Electric Power Markets (PJM) regulation zone preliminary billing data in 2017	SVM RF DT MLP CNN LSTM EPNET(combined CNN and LSTM)	28.98 9.20 9.74 9.86 9.80 9.85 8.84		34.28 19.47 24.88 18.98 18.9 18.9 17.9	(Kuo & Huang, 2018)
Day-ahead electricity prices in German from 2011 to 2018	BRIM(bidirectional long short-term memory) LSTM-DNN	6.29 10.20	15.56 23.24		(Chen et al., 2019)
Day-ahead electricity prices in various Europe, including Belgium, France, Germany, and Nord Pool	ANN Belgium France Germany Nord Pool	 5.66 4.21 4.22 1.89	 15.5 12.29 16.88 6.93	 10.99 7.98 6.83 4.51	(Gunduz et al., 2023)

2.7 Transfer Learning

In the past few decades, transfer learning has become a state-of-the-art learning technique. Traditional machine learning relies on vast amounts of training data to develop a model that can predict future outcomes. In contrast, transfer learning seeks to apply existing knowledge from known data to unknown data. The transfer leaning can be categorised into two main groups: homogeneous transfer learning and heterogeneous transfer learning (Weiss, Khoshgoftaar, & Wang, 2016). Homogeneous transfer learning works when a dataset exists that is related to, but not an exact match for the domain of interest. Homogeneous transfer learning can be applied to construct a predictive model for the target domain, provided that the input feature space remains the same. On the contrary, heterogeneous transfer learning refers to the situation where the source and target domains are represented in different feature spaces.

Using knowledge learned from France, the prediction of day-ahead electricity prices in other European countries can be classified under homogeneous transfer learning, specifically instance-based transfer learning.

Chapter 3
Methodology

The main content of this chapter is to clearly articulate research methods, which satisfy the objectives of this report. The chapter mainly covers the details of research methodology for day-ahead electricity prices prediction in France using deep learning which will be clearly introduced with the confident and imaginative use of the feature description methods.

3.1 Method Design

The design methodology is illustrated in Figure 3.1, which includes the following main steps: data collection, data preparation, feature selection, training and testing, hyperparameter tuning, and evaluation. In this chapter, we will provide a detailed overview of each step.

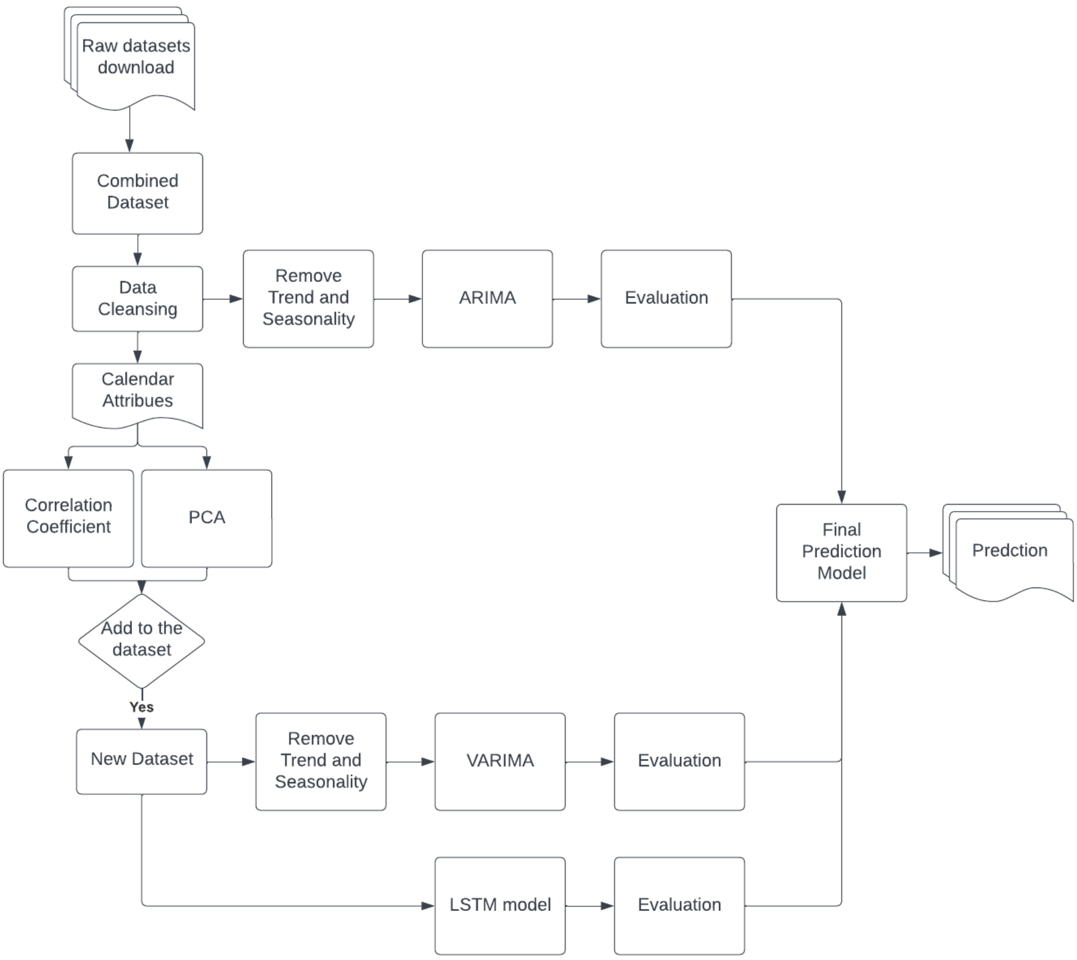

Figure 3.1 The method design

3.2 Data Collection

We obtained the data from the ENTSO-E Transparency Platform (https://www.entsoe.eu/data/). ENTSO-E is a reliable provider of market data for European electricity markets. The transparency platform was launched in 2015, offering a wide range of data, such as day-ahead prices, generation and consumption loads in various countries in Europe.

Microsoft Power BI is a data analysis tool that provides multiple functions for data transformation and data visualisation. We utilised the Power Query (Webb, 2014) functions to create an external dataset with calendar date features, generated from the Date module. The functions, including Month and Day of Week, provide the numerical month and day of the week (starting from Sunday) respectively. Additionally, we utilised the column split function in Power Query to separate the date and time from the timestamp column. We used the Append function to merge the raw data by year into one dataset comprising of three years' worth of data.

3.3 Explore Data Analysis

Exploratory data analysis (EDA) is a crucial step in uncovering dataset patterns. To analyse the data, we utilised the line chart visual in Power BI (Ferrari & Russo, 2016) to display trends.

3.4 Data Preparation

Before applying algorithms, it is crucial to clean the data. There are various methods to accomplish this, including replacing any missing data with the mean or median of other data. Additionally, some methods were created for big data cleansing such as Cleanix (Wang et al., 2014) and KATARA(Chu et al., 2015) etc. However, the simplest approach is to remove any missing or illegal data; while we need to keep in mind that this action may decrease the size of the dataset and affect the performance of the algorithm.

In this experiment, external data was introduced, and we need to add its features to the dataset. To accomplish this, we utilised the relationship function in Power BI to combine the calendar with the prices. A relationship was established as depicted in Figure 3.2, connecting the date in both tables with a one-to-many relationship.

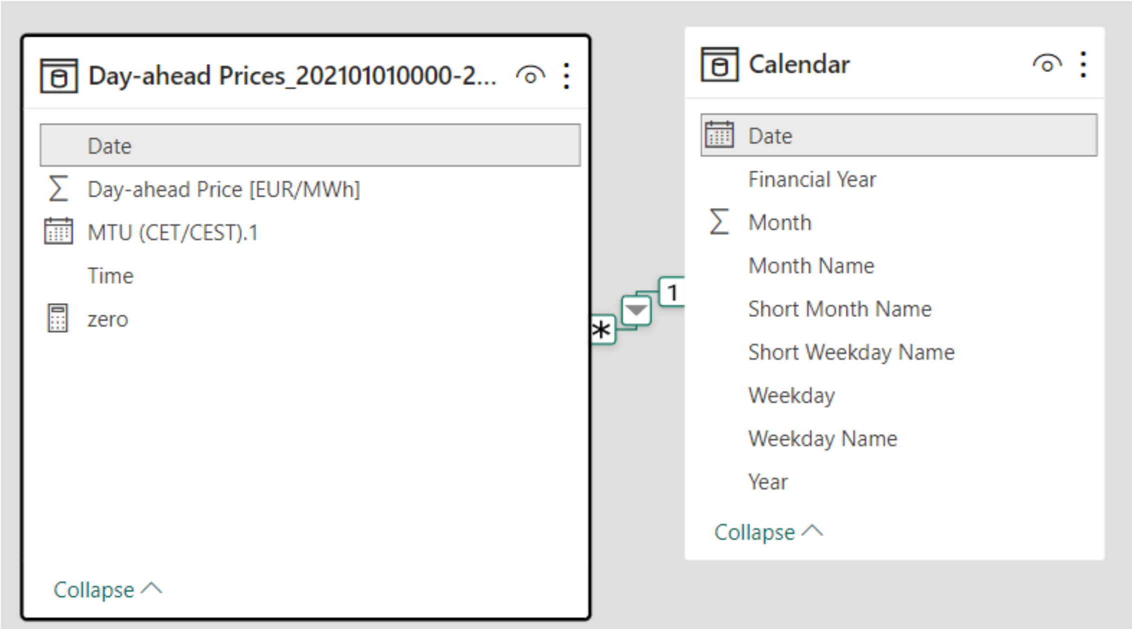

Figure 3.2 The relationship between date in Calendar and date in Day-ahead Electricity Prices

3.5 Feature Selection

In this project, we took use of the Pearson correlation coefficient to determine if there is a relationship between two variables. Correlation refers to the extent of association between two variables (Asuero, Sayago, & González, 2006). The Pearson correlation coefficient is a way to measure linear correlation. It gives the ratio between the covariance of two variables and the product of their standard deviations as equation 3.1 shows. The correlation is quantified with a number that ranges from -1 to $+1$; with 0 indicating no correlation, -1 indicating a perfect correlation, and -1 indicating a negative correlation (Akoglu, 2018). As the value moves away from 0, the strength of the correlation increases, whether it is positive or negative.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y} \tag{3.1}$$

PCA, or principal component analysis, is a technique used to reduce the number of variables in a dataset while retaining the maximum amount of relevant information (Daffertshofer, Lamoth, Meijer, & Beek, 2004). PCA provides the percentage of variance explained by each selected component. The main purpose of using PCA is to identify the most significant variance present in the dataset, therefore, to remove the less significant variable. This calculation involves three steps: (1) standardising the data using equation 3.2; (2) computing the covariance matrix; and (3) determining the eigenvectors and eigenvalues.

$$Z = \frac{value - mean}{standard\ deviation} \tag{3.2}$$

3.6 Algorithms

3.6.1 LSTM

Long short-term memory is a similar control flow as a recurrent neural network, as it processes data passing on information as it propagates forward. LSTM is one of the most advanced networks to process temporal sequences, due to complex features such as non-linearity, non-stationarity, and sequence correlation (Staudemeyer & Morris, 2019). A LSTM cell can be illustrated as Figure 3.3. The information is sent to three different gates where it undergoes processing through the application of activation functions. Once processed, the results from each gate are multiplied and added together before being passed on to the next cell.

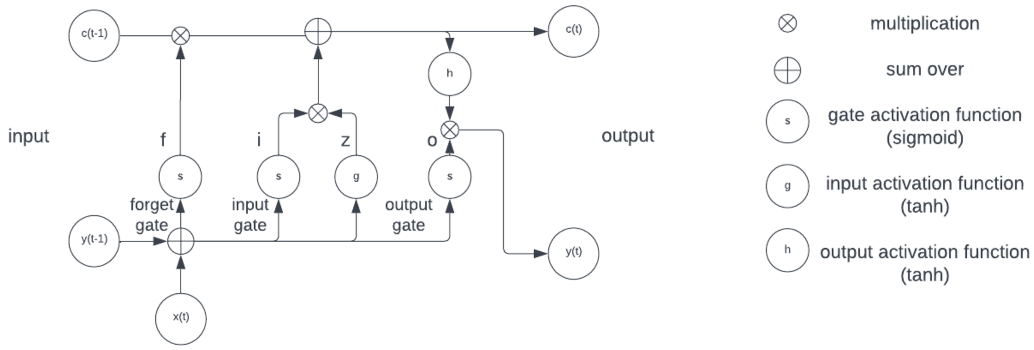

Figure 3.3 LSTM Cell

We are use of the LSTM class in library Keras which is built in Python, in which the activation function is tanh as equation 3.3 illustrates and the recurrent activation function is sigmoid as equation 3.4 shows. The LSTM works well as it introduces three gates, the input gate, the forget gate and the output gate(Van Houdt, Mosquera, & Nápoles, 2020).

The activation function is expressed as,

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.3}$$

which returns a value between -1 and $+1$ to regulate the network in the input gate.

The recurrent activation function is expressed as,

$$Sigmoid(x) = \frac{1}{1+e^{-x}} \tag{3.4}$$

which returns a value between 0 and 1, and is used to decide whether the factors passed through are important or not in the forget gate.

The input gate regulates the factors passed through by using the activation function and evaluates the factors that are important or not by using the recurrent function. Then combining the results from the two functions as the output of the input gate.

Lastly, the output gate calculates the current cell state by multiplying the value returned by the forget gate and the current cell state to drop meaningless factors, and then adding the output of the input gate to pass it to the next cell.

We now discuss three crucial hyperparameters: the number of LSTM units, the number of layers, and the choice of optimiser. The number of LSTM units impacts the layer's ability to learn patterns from sequential data by determining its complexity and capacity. More units can capture more complex dependencies, but this also increases computational requirements and the possibility of overfitting. The depth of an LSTM model is determined by the number of layers, with deeper networks capturing more complex patterns. However, this also requires more computational resources. The choice of optimiser determines the update rules that affect the speed and stability of training. A widely used optimisation algorithm in deep learning is the Adam optimiser, which combines the advantages of both the Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). This optimiser adjusts the learning rates for each parameter by considering the moving averages of the gradients and squared gradients. The optimiser applied in this project is Adam.

3.6.2 ARIMA

Another algorithm applied in this project is ARIMA. We removed the seasonality and the trend of the dataset so that ARIMA can be applied. A series is considered stationary if its mean remains constant despite a change in time origin, resulting in the expected value being consistent across all time points. Additionally, the variable's variance must remain stable throughout time, and neither the mean nor variance should depend on the time of measurement (Vasileiadou & Vliegenthart, 2014). The three parameters d refers to the number of differencing transformations required by the time series to get stationary; p refers the lag order; and q refers the order of the moving average. The q and p will be tuned based on the value of Auto Correlation Function (ACF) and Partial Correlation Function (PCAF).

The Auto Correlation Function takes all past observations into account, regardless of their impact on the present or future time period. It calculates the correlation between the current time period, denoted as t and a previous time period denoted as $t-k$. This calculation includes all the intervals, or lags, between these two time periods. The correlation is calculated using the Pearson Correlation formula.

The correlation between two variables y_1 and y_2 is expressed as,

$$\rho = \frac{Cov(y_1,y_2)}{\sigma_1\sigma_2} \tag{3.5}$$

where cov stands for coefficient of variation and σ_{1},σ_2 are their standard deviations.

The PACF calculates the partial correlation between time periods t and $t-k$, but it only considers the direct impact of time lags on future time periods. For instance, if today's stock price depends on the stock price from three days ago, it may not take into account yesterday's closing price. Therefore, we only consider the relevant time lags between t and $t-k$ and ignore any insignificant time lags in between. To obtain each partial correlation, we conduct a series of regressions in the following form:

$$\tilde{y}_t = \phi_{21}\tilde{y}_{t-1} + \phi_{22}\tilde{y}_{t-2} + e_t \tag{3.6}$$

where $\tilde{y}_t$ is the original series minus the sample mean $y_t - \bar{y}$. The estimate of ϕ_{22} will give the value of the partial autocorrelation of order 2. The optimal value of p is determined by whether the PACF meets the cut-off point, while the optimal value of q is determined by whether the ACF meets the cut-off point.

3.6.3 VARMAX

In this project, the third algorithm utilised is the Vector Autoregressive Moving Average with Exogenous Regressors (VARMAX). Equation (3.7) represents the mathematical formulation for VARMAX model.

$$y_t = a + X_t.b + A_1 y_{t-1} + \cdots + A_p y_{t-p} + \varepsilon_t + B_1 \varepsilon_{t-1} + \cdots + B_q \varepsilon_{t-q} \tag{3.7}$$

where at time t, the variable y_t represents a k×1 vector of response time series variables, consisting of k elements. The constant vector a, also with k elements, serves as an offset. The k×r matrix X_t represents exogenous terms at each time t, where r is the number of exogenous series. The constant vector b, with a size of r, acts as the regression coefficients. The product X_t multiplied by b results in a vector of size k. For each j, the k×k parameter matrices A_j and B_j are autoregressive and moving average matrices, respectively. There are p autoregressive matrices and q moving average matrices. Finally, the ε_t represents the vector error (Kadiyala & Kumar, 2014).

3.7 Program Implementation

We utilised various libraries including pandas, numpy, sklearn, matplot, matplotlib, keras, statsmodels, and datetime to develop the program in Jupyter notebook using Python 3.7.7. The results were run on a computer with Intel(R) Core (TM) i7-9700KF CPU @ 3.60GHz with 32 GB RAM.

3.8 Evaluation Methods

The evaluation methods in this report are Mean Squared Error (MSE), Root Mean Square Deviation (RMSE), and Mean Absolute Percentage Error (MAPE). MSE is calculated as the average of the squared forecast error values. The mathematical formulation is as equation 3.8 shows. By squaring the forecast error values, they are forced to be positive and larger errors are given more weight. This means that very large or outlier forecast errors are squared, resulting in a larger mean squared error score. Essentially, this score penalises models that make larger incorrect forecasts, resulting in worse performance.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{3.8}$$

The mathematical formulation of RMSE is as follows,

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

where the RMSE is the square root of MSE, they are monotonically correlated. Therefore, following the same evaluation of MSE, the smaller the RMSE the better the model. The mathematical formulation of MAPE is as follows,

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{Y_t - X_t}{Y_t}\right| \tag{3.10}$$

which has an intuitive interpretation in terms of relative error, making it useful for tasks where sensitivity to relative variations is more important than absolute variations. However, there are some drawbacks to using MAPE. Its use is restricted to strictly positive data by definition and it is biased towards low forecasts, making it unsuitable for predictive models where large errors are expected(Chicco, Warrens, & Jurman, 2021). The smaller the MAPE the better the model.

3.9 Transfer Learning

This paper used transfer learning, which can be defined as follows: a domain D consists of a feature space X and a marginal probability distribution $P(X)$, where $X = \{x_1, \dots, x_n\} \in X$. A task T consists of two components: a label space Y and a predictive function f. The function is trained using feature vector and label pairs $\{x_i, y_i\}$, where $x_i \in X$ and $y_i \in Y$. The set of data used as the source domain is denoted as D_s consisting of data instances $\{(x_{s1}, y_{s1}), \dots, (x_{sn}, y_{sn})\}$, where $x_{si} \in X_s$ represents the ith data instance of D_s and $y_{si} \in Y_s$ represents its corresponding class label. Similarly, the target domain is denoted as D_t, consisting of data instances $\{(x_{t1}, y_{t1}), \dots, (x_{tn}, y_{tn})\}$, where $x_{ti} \in X_t$ represents the ith data instance and $y_{ti} \in Y_t$ represents its corresponding class label. Additionally, the source task is represented as T_s, the target task as T_t, the source predictive function as f_s, and the target predictive function as f_t (Zhuang et al., 2020).

Transfer learning is the process of improving the predictive function f_t of a target domain D_t by utilising related information from a source domain D_s with a corresponding source task T_s. This is achieved by transferring knowledge from the source domain D_s utilising the task T_s to the target domain D_t, where $D_s \neq D_t$ or $T_s \neq T_t$ (Zhuang et al., 2020).

Chapter 4
Results

The main content of this chapter is to demonstrate the experimental results. In the end, this chapter will also discuss the limitations of the project.

4.1 Data Collection and EDA

The yearly raw data was downloaded from ENTSO-E, and then combined into one on Power BI. It contains hourly day-ahead electricity prices in France from 2019 to 2022 with two columns, the time stamp and the day-ahead electricity price in euro per megawatt-hour (EUR/MWh). The final dataset has 35,068 rows and five columns.

The EDA as Figure 4.1 shows the prices were stable from 2019 to June 2021, with an average of 40.32. However, prices have significantly increased since July 2021, with an average of 159.01(294.59% increase) in 2021 and 275.88(584.23% increase) in 2022.

It is interesting to note that there were 207 rows with negative or zero prices. Additionally, there were 4 (0.01%) missing values in the four-year period. Lastly, there was a spike in average price at 7 and 8 in the morning on the 4th of April 2022, reaching 2850.39 (933.2% higher than normal).

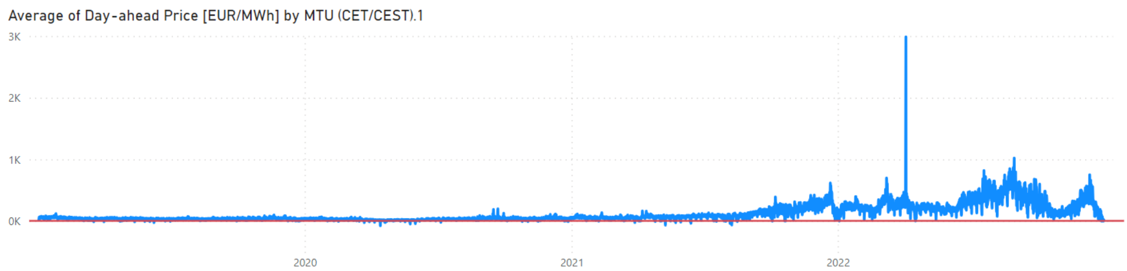

Figure 4.1 Average day-ahead electricity prices from 2019 to 2022

The combined dataset now included three calendar date attributes: time, month, and weekday. According to Figure 4.2, the highest prices were observed at 8 AM in the morning and 7 PM in the evening. Additionally, the highest prices were recorded in August and December. On the other hand, prices were lower on weekends compared to weekdays.

Figure 4.2 Day-ahead electricity prices by Time Month and Weekday from 2019 to 2022

4.2 Data Preparation and Feature Selection

In Section 4.1, we noted that the missing values only comprised 0.01% of the dataset, which is too insignificant to impact the results. Therefore, we removed them from the dataset.

We have tested the correlation coefficient, as shown in Figure 4.3. Despite the fact that the date and time attributes appear to correlate with the prices during EDA, the correlation coefficient indicated that they are not significant. It is worth noting that all the attributes are derived from the time stamp column, which already contains this information.

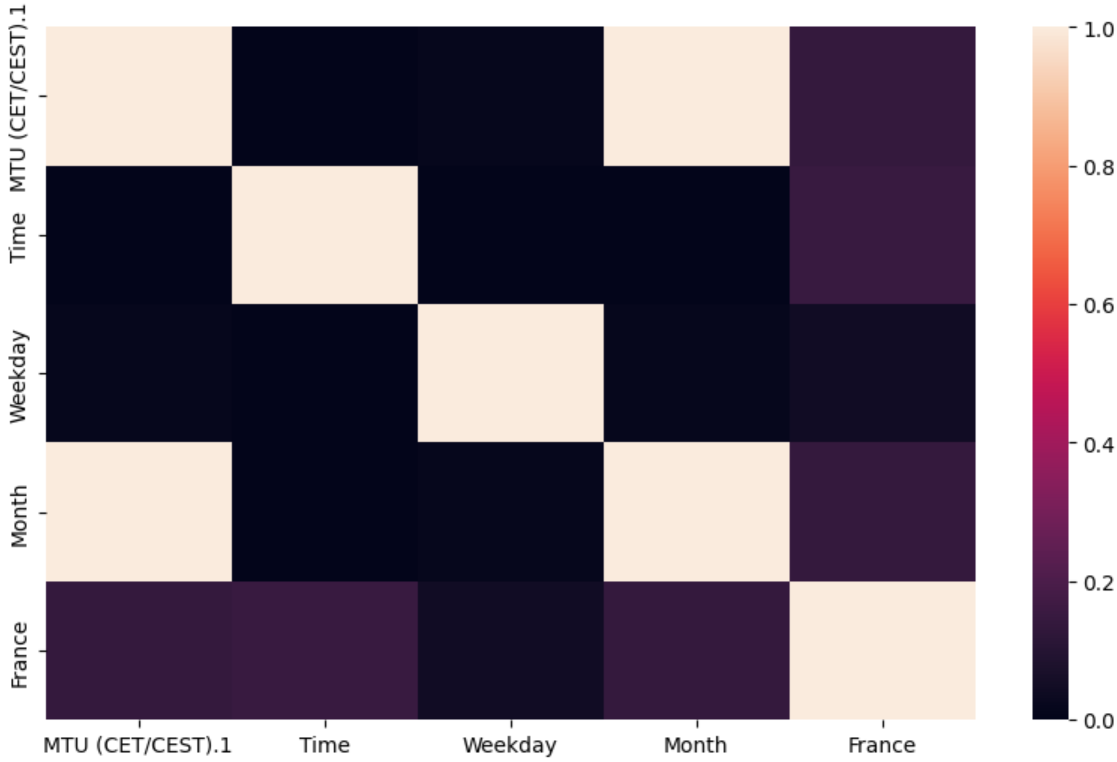

Figure 4.3 Correlation Coefficient

By applying PCA selection, the correlation coefficient results are further validated. Figure 4.4 shows that the top component is the time stamp, followed by the time column.

```
Principal Component 1:
[0.6934618241393993, 0.6933555370705626, 0.19216138770954333, 0.02939188763253542, 0.02406066686081691]

Principal Component 2:
[0.7231890967328714, 0.6466487692165499, 0.18823017169976441, 0.10857714639160909, 0.10781143436921452]
```

Figure 4.4 PCA Scores

After the evaluation, three datasets have been prepared for the experiment. They were: 1) dataset A, which contains three years' worth of day-ahead prices from 1/1/2019 to 31/12/2022 in France; 2) dataset B, which contains one year's worth of day-ahead prices from 1/1/2022 to 31/12/2022 in France; and 3) dataset C, which also contains one year's worth of day-ahead prices from 1/1/2022 to 31/12/2022 in France, but with the calendar date attributes included. More details about each data set can be found in Table 4.1.

Table 4.1 Datasets experimented

Dataset	Date Range	Column	Row
A_training	1/1/2019 to 31/12/2021	2	26,304
A_testing	1/1/2022 to 31/12/2022	2	8,760
B_training	1/1/2022 to 30/6/2022	2	4,343
B_testing	1/7/2022 to 31/12/2022	2	4,417
C_training	1/1/2022 to 30/6/2022	5	4,343
C_testing	1/7/2022 to 31/12/2022	5	4,417

To apply ARIMA and VARMAX, the datasets need to be prepared by analysing their seasonality and trend. Since this was a short-term price prediction and the prices were recorded hourly, a half-day lag (lag = 12) was chosen for the experiment. Figure 4.5 reveals a significant upward trend, and there is an obvious up-and-down seasonal pattern with a half-day lag.

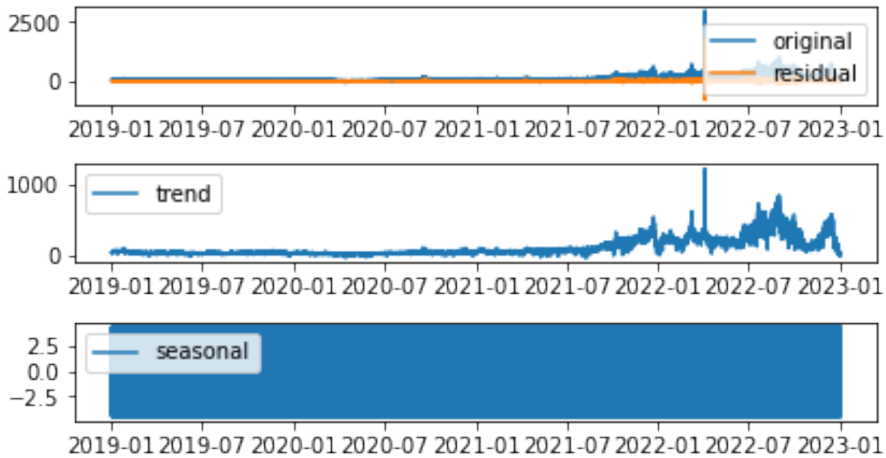

Figure 4.5 Seasonality and trend analysis

4.3 Hyperparameter Tuning

4.3.1 LSTM

When working with LSTM, there are two crucial hyperparameters to consider: epoch and batch. The epochs refer to the number of times the learning algorithm passes through the entire training dataset. Each sample in the training dataset is given an opportunity to update the internal model parameters during an epoch. An epoch is made up of one or more batches.

The batch size refers to the amount of samples that are processed before updating the model. It is important to note that the batch size should be at least one and no greater than the number of samples in the training dataset. There are no set rules for configuring these parameters. Experiments with different values to determine what works best are adopted and used in this project.

4.3.2 ARIMA

When it comes to differencing, a value of two might be best, but we decided to plot the original dataset in both cases (as shown in Figure 4.6). We did not notice a significant difference in the selection of d, so we opted for a simpler model with a better performance by selecting a value of one for d.

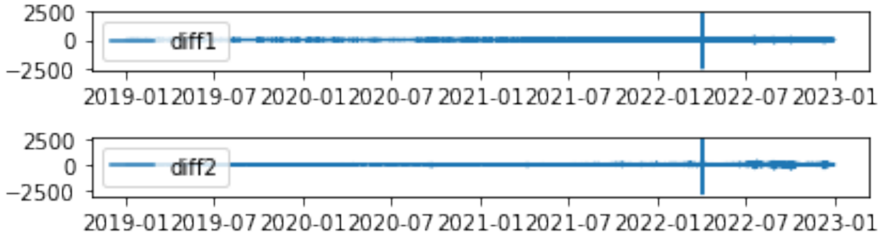

Figure 4.6 The selection of d

In Chapter 3, we were introduced to ACF and PACF, which are calculated and displayed in Figures 4.7 and 4.8. The ACF and PACF both show that only the first lag is significantly outside of the limit, and although the second lag is slightly outside as well, it is not too far. As a result, we chose to set the value of both p and q to one.

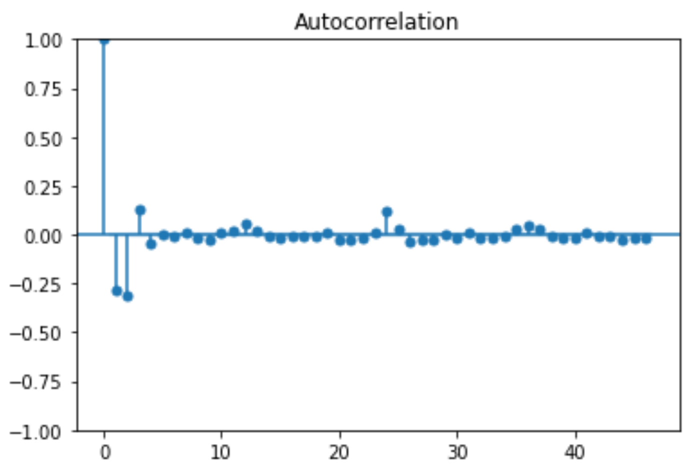

Figure 4.7 The selection of q

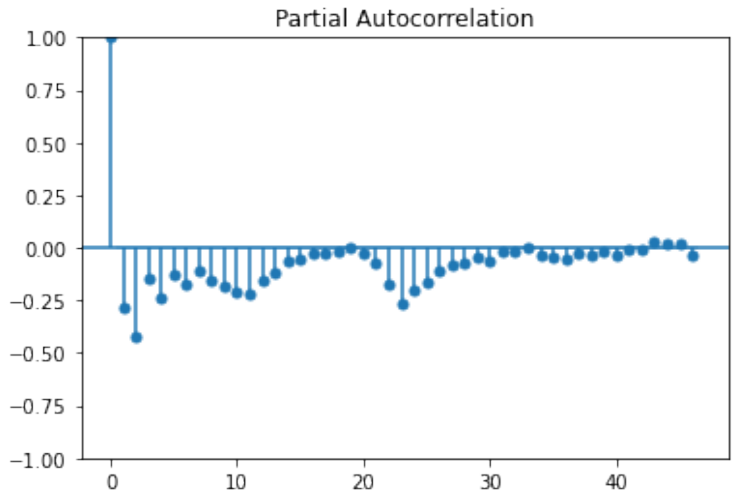

Figure 4.8 The selection of p

4.4 Results and Comparison

Table 4.2 displays the results of running and testing various algorithms with different datasets. It is evident that algorithms utilising dataset B have demonstrated superior outcomes, which further validated our exploratory data analysis (EDA) by revealing a significant increase in electricity prices since 2022. Based on the data, it appears that the optimal outcome was reached with two layers, 150 units, 250 epochs, and a batch size of 32. The resulting mean squared error was 29.24 and the root mean squared error was 5.45. The entire process took three minutes and 0.7 seconds to execute.

Table 4.2 Algorithms applied and evaluation

Dataset	Algorithm	Hyperparameters	Execution Time	MSE	RMSE
A	ARIMA	p=1 d=1 q=1	2.4s	2349.29	275.93
A	LSTM	Layers=1 Units=150 epochs=150 batch size=16	6m10s	3,640.99	60.34
A	LSTM	Layers=1 Units=200 epochs=200 batch size=16	12m32s	2,970.89	54.51

Dataset	Algorithm	Hyperparameters	Execution Time	MSE	RMSE
A	LSTM	Layers=1 Units=200 epochs=200 batch size=32	8m45s	2,633.38	51.32
A	LSTM	Layers=2 Units=250 epochs=250 batch size=32	22m44.2s	1710.43	41.36
B	ARIMA	p=1 d=1 q=1	0.7s	1527.28	203.95
B	LSTM	Layers=1 Units=150 epochs=150 batch size=16	54s	247.37	15.73
B	LSTM	Layers=1 Units=200 epochs=200 batch size=16	1m24s	186.78	13.67
B	LSTM	Layers=1 Units=200 epochs=200 batch size=32	49s	99.92	10
B	LSTM	Layers=1 Units=250 epochs=250 batch size=32	1m23s	54.45	7.38
B	**LSTM**	**Layers=2** **Units=250** **epochs=250** **batch size=32**	**3m 0.7s**	**29.24**	**5.41**
C	VARIMA	p=1 d=1 q=1	2m13s	1,527.28	39.08
C	LSTM	Layers=1 Units=150 epochs=150 batch size=16	1m16s	31,751.46	178.19
C	LSTM	Layers=1 Units=200 epochs=200 batch size=16	1m58s	29,545.47	171.89
C	LSTM	Layers=1 Units=200	1m22s	30,146.96	173.63

Dataset	Algorithm	Hyperparameters	Execution Time	MSE	RMSE
		epochs=200 batch size=32			
C	LSTM	Layers=2 Units=250 epochs=250 batch size=32	5m2s	30,836.25	175.6

The LSTM network is structured as Figure 4.9 shows.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 1, 250)            252000

 lstm_1 (LSTM)               (None, 250)               501000

 dense (Dense)               (None, 1)                 251

=================================================================
Total params: 753,251
Trainable params: 753,251
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.9 Model structure

The visualisation of the best prediction adopted using LSTM with data set B and having two layers, 250 units, the number of epochs equals 250 and the number of batch size equals 32 is displayed in Figure 4.10.

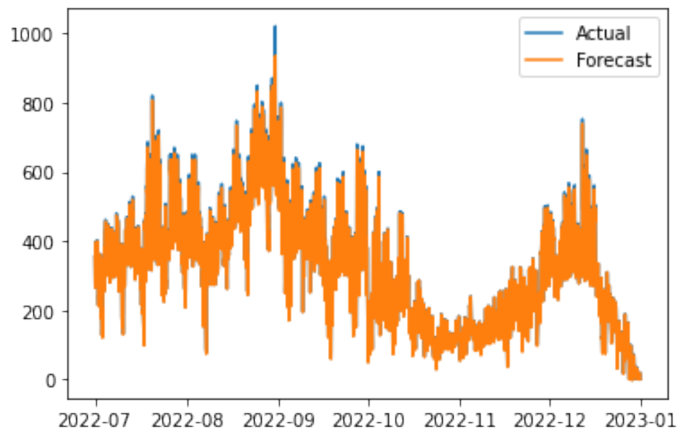

Figure 4.10 Predictions

Here we conducted an analysis of the amount of training data. In the daily operation, we want to minimize the size of training data so that it lowers down the time needed for daily prediction and it increases the performance of the model at the same time. We experimented with datasets of varying sizes, including 30 days, 15 days, 7 days, 3 days, and 1 day. Our goal was to predict 24-hour future outcomes. We selected three datasets randomly for each interval from dataset B and have displayed the results in Table 4.3. Based on the data, it appears that the optimal outcome was achieved with a 30-day interval, resulting in a 20.28 mean squared error, 0.9 mean absolute percentage error, and 3.64 root-mean-square error.

As shown in Figure 4.11, the three 30-day training datasets have been visualised to predict the day-ahead electricity prices for the next 24 hours.

Table 4.3 Evaluation on the impact of different length of training data sets

Dataset	Shape	Algorithm	Parameters	MSE	MAPE	RMSE
30 days	**Training 720 rows Testing 24 rows**	**LSTM**	**layer=2 units=250 epochs=250 batch size=32**	**3.47**	**0.78**	**1.35**
15 days	Training 360 rows Testing 24 rows	LSTM	Layer=2 Units=250 epochs=50 batch size=2	13.6	1.42	3.63
7 days	Training 168 rows Testing 24 rows	LSTM	Layer=2 Units=250 epochs=50 batch size=2	852.9	9.58	24.73
3 days	Training 72 rows Testing 24 rows	LSTM	Layer=2 Units=250 epochs=25 batch size=1	4367.16	28.18	65.37
1 day	Training 24 rows Testing 24 rows	LSTM	Layer=2 Units=250 epochs=25 batch size=1	17058.43	59.66	120.79

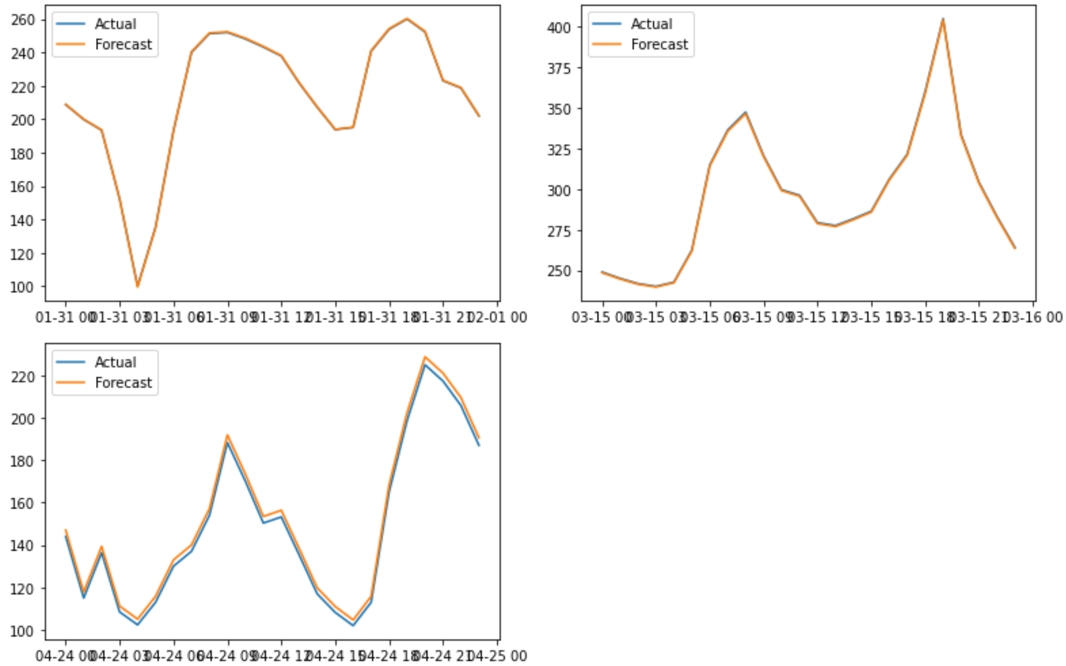

Figure 4.11 Final predictions of using 30-day training dataset

4.5 Transfer Learning

Upon completion of the project, we implemented transfer learning techniques by utilising training dataset B to fine-tune a pre-existing LSTM model. We then applied this model to forecast the day-ahead electricity prices in Germany, Belgium, and the UK. The results of this transfer learning approach can be found in Table 4.4, showcasing its impressive success.

Table 4.4 Transfer leaning on other countries in Europe and evaluation

Training Dataset	Testing Dataset	Algorithm	Parameters	MSE	RMSE
B_training	Germany 1/7/2022 – 31/12/2022	LSTM	Layer=1 Units=250 epochs=250 batch size=32	12.15	3.49
B_training	Belgium 1/7/2022 – 31/12/2022	LSTM	Layer=1 Units=250 epochs=250 batch size=32	28.0	5.29
B_training	UK 1/7/2022 – 31/12/2022	LSTM	Layer=1 Units=250 epochs=250 batch size=32	1.28	1.13

Figure 4.12 displays the visualisation of transfer learning applied on day-ahead electricity prices prediction in Germany, Belgium, and UK.

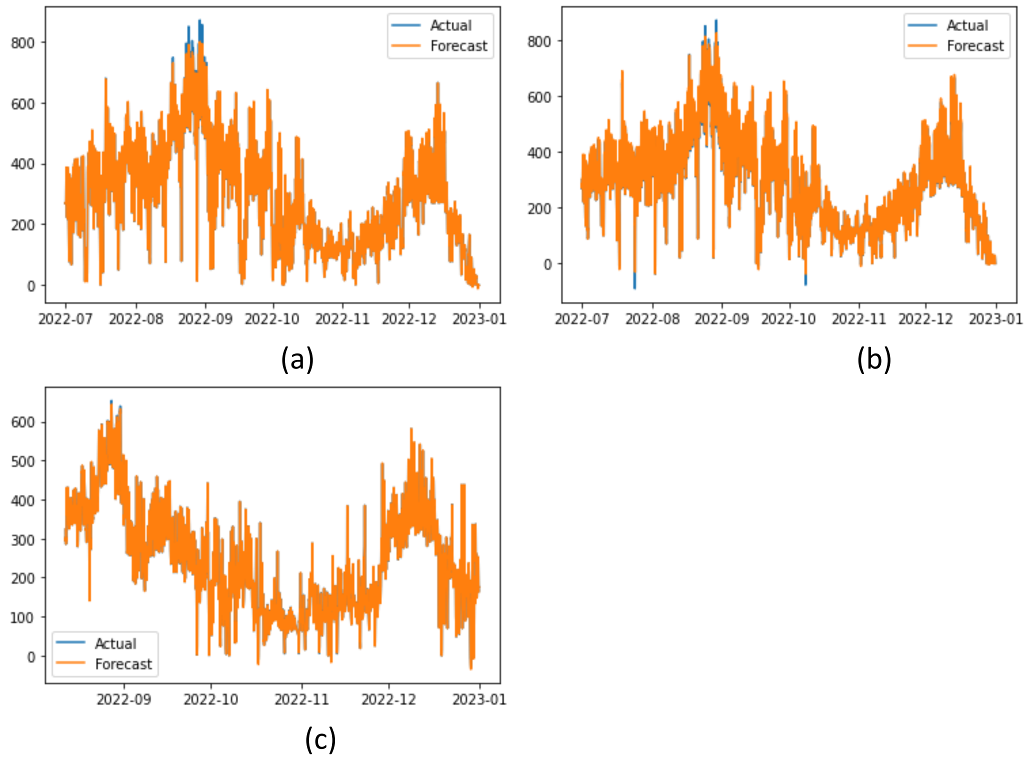

Figure 4.12 Transfer learning on day-ahead electricity prices in (a) Germany, (b) Belgium, and (c) UK

4.6 Limitations of this ResearchProject

The LSTM is suitable and feasible to predict the day-ahead electricity prices in France. However, the limitations of this paper are: (1) the experiments focus on short-term predictions; it may not take the patterns of yearly trend into account; and (2) it may not be responsible to other influences to the whole electricity markets such as social problems and the debate of using nuclear electricity generation etc.

Chapter 5
Analysis and Discussions

In this chapter, experimental results are analysed and compared. Comparisons of the results under various conditions will be mentioned.

5.1 Analysis

To summarise, our research found that the LSTM model is effective in predicting day-ahead electricity prices in France. The mean squared error we achieved was 29.24 and the root mean squared error was 5.24, both of which were better than the results achieved by traditional statistical methods including ARIMA with a mean squared error of 1527.28 and a root mean squared error of 203.95, and VARMAX with a mean squared error of 1527.28 and a root mean squared error of 39.08.

We also recommend using a 30-day training dataset for daily operations, as this produced a mean squared error of 3.47, a mean absolute percentage error of 0.78, and a root mean square error of 1.35. These results are significantly better than those reported in previous studies outlined in Chapter 2.

Lastly, we analysed the transfer learning techniques on performing the day-ahead electricity prices prediction in other state members in European Union.

5.2 Discussions

We conducted experiments to compare the performance of LSTM and ARIMA in predicting day-ahead electricity prices in France. Our analysis of the two algorithms' performance under different conditions revealed that LSTM generally outperforms ARIMA. However, when using dataset C, we discovered that the performance of LSTM decreased and was worse than the performance of VARMAX. We concluded that LSTM may be sensitive to insignificant attributes, making feature selection an important step before using the LSTM algorithm. However, this requires to be further experimented and proven.

Chapter 6
Conclusion and Future Work

In this chapter, we will summarise the subject and method of this project and propose new research direction according to the result and insufficiency of the experiment, preparing for the future work.

6.1 Conclusion

In conclusion, the purpose of this report is to propose and implement a method to predict the day-ahead electricity prices in France using deep learning. Three major tasks were defined and implemented: (1) LSTM, ARIMA, and VARMAX were tested on various length of dataset; the results of each were evaluated and compared; (2) a test on the impact of different length of dataset for predicting the following 24-hour day-ahead electricity prices were conducted; and (3) we tested transfer learning technique to the day-ahead electricity prices in other countries in the European Union. By doing so, three research questions were answered: (1) LSTM performs effectively and efficiently on day-ahead electricity prices prediction in France by achieving a mean squared error 29.24 and the root mean squared error 5.24; (2) deep learning based algorithm performs better than statistical algorithms on complex problems like the day-ahead electricity prices prediction in France; and (3) in practice, a 30-day training dataset is sufficient to predict the next 24-hour day-ahead electricity prices in France.

Furthermore, the experiment on transfer learning has been highly successful in predicting day-ahead electricity prices in Germany, Belgium, and the UK.

6.2 Future Work

In order to improve our predictions, we plan to undertake the following tasks in the future: (1) analysing whether negative electricity prices, and the spike that occurred in April 2022 affects the performance of the prediction; (2) examining the relationship between electricity storage and pricing by analysing load data, as discussed in Chapter 1; and (3) to divide the analysis by season and different times of the day.

References

Ahmad, T., Zhang, H., & Yan, B. (2020). A review on renewable energy and electricity requirement forecasting models for smart grid and buildings. Sustainable Cities and Society, 55, 102052.

Akoglu, H. (2018). User's guide to correlation coefficients. Turkish Journal of Emergency Medicine, 18(3), 91-93.

Asuero, A. G., Sayago, A., & González, A. (2006). The correlation coefficient: An overview. Critical Reviews in Analytical Chemistry, 36(1), 41-59.

Böckers, V., Haucap, J., & Heimeshoff, U. (2013). Benefits of an integrated European electricity market: DICE Discussion Paper.

Cartea, A., & Figueroa, M. G. (2005). Pricing in electricity markets: a mean reverting jump diffusion model with seasonality. Applied Mathematical Finance, 12(4), 313-335.

Chatfield, C. (2000). Time-series forecasting: CRC press.

Chen, Y., Wang, Y., Ma, J., & Jin, Q. (2019). BRIM: An accurate electricity spot price prediction scheme-based bidirectional recurrent neural network and integrated market. Energies, 12(12), 2241.

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science, 7, e623.

Chu, X., Morcos, J., Ilyas, I. F., Ouzzani, M., Papotti, P., Tang, N., & Ye, Y. (2015). KATARA: Reliable data cleaning with knowledge bases and crowdsourcing. Proceedings of the VLDB Endowment, 8(12), 1952-1955.

Daffertshofer, A., Lamoth, C. J., Meijer, O. G., & Beek, P. J. (2004). PCA in studying coordination and variability: a tutorial. Clinical Biomechanics, 19(4), 415-428.

Doborjeh, M., Liu, X., Shen, Y. et al. (2023) Prediction of Tinnitus treatment outcomes based on EEG Sensors and TFI score using deep learning, Sensors.

Fanone, E., Gamba, A., & Prokopczuk, M. (2013). The case of negative day-ahead electricity prices. Energy Economics, 35, 22-34.

Ferrari, A., & Russo, M. (2016). Introducing Microsoft Power BI: Microsoft Press.

Forrester, R. (2016). History of electricity. Available at SSRN 2876929.

Fu, T.-C. (2011). A review on time series data mining. Engineering Applications of Artificial Intelligence, 24(1), 164-181.

Fuller, W. A. (2009). Introduction to statistical time series: John Wiley & Sons.

Gunduz, S., Ugurlu, U., & Oksuz, I. (2023). Transfer learning for electricity price forecasting. Sustainable Energy, Grids and Networks, 34, 100996.

Herrera, A., Beck, A., Bell, D., Miller, P., Wu, Q., Yan, W. (2008) Behavior analysis and prediction in image sequences using rough sets. International Machine Vision and Image Processing Conference (pp.71-76)

Hyndman, R. J. (2001). ARIMA processes. Datajobs: Data science knowledge.

Kadiyala, A., & Kumar, A. (2014). Vector time series models for prediction of air quality inside a public transportation bus using available software. Environmental Progress & Sustainable Energy, 33(4), 1069-1073.

Karabiber, O. A., & Xydis, G. (2019). Electricity price forecasting in the Danish day-ahead market using the TBATS, ANN and ARIMA methods. Energies, 12(5), 928.

Kuo, P.-H., & Huang, C.-J. (2018). An electricity price forecasting model by hybrid structured deep neural networks. Sustainability, 10(4), 1280.

Lago, J., De Ridder, F., Vrancx, P., & De Schutter, B. (2018). Forecasting day-ahead electricity prices in Europe: The importance of considering market integration. Applied energy, 211, 890-903.

Liang, S., Yan, W. (2022) A hybrid CTC+Attention model based on end-to-end framework for multilingual speech recognition. *Springer Multimedia Tools and Applications.*

Liu, J., Yan, W. (2022) Crime prediction from surveillance videos using deep learning. Aiding Forensic Investigation Through Deep Learning and Machine Learning Frameworks. IGI Global.

Lu, J., Nguyen, M., Yan, W. (2021) Sign language recognition from digital videos using deep learning methods. *International Symposium on Geometry and Vision.*

Newbery, D., Strbac, G., & Viehoff, I. (2016). The benefits of integrating European electricity markets. Energy Policy, 94, 253-263.

Nguyen, M., Yan, W. (2022) Temporal color-coded facial-expression recognition using convolutional neural network. *International Summit Smart City 360°: Science and Technologies for Smart Cities.*

Parloff, R. (2016). Why Deep Learning Is Suddenly Changing Your Life. Fortune. New York: Time Inc.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., . . . Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys (CSUR), 51(5), 1-36.

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, 132306.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018). A comparison of ARIMA and LSTM in forecasting time series. IEEE International Conference on Machine Learning and Applications (ICMLA).

Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. arXiv preprint arXiv:1909.09586.

Vahidinasab, V., Jadid, S., & Kazemi, A. (2008). Day-ahead price forecasting in restructured power systems using artificial neural networks. Electric Power Systems Research, 78(8), 1332-1342.

Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. Artificial Intelligence Review, 53, 5929-5955.

Vasileiadou, E., & Vliegenthart, R. (2014). Studying dynamic social processes with ARIMA modeling. International Journal of Social Research Methodology, 17(6), 693-708.

Wang, H., Li, M., Bu, Y., Li, J., Gao, H., & Zhang, J. (2014). Cleanix: A big data cleaning parfait. ACM International Conference on Conference on Information and Knowledge Management.

Wang, X., Yan, W. (2019) Human gait recognition based on self-adaptive hidden Markov model. *IEEE/ACM Transactions on Biology and Bioinformatics*.

Wang, X., Yan, W. (2019) Human gait recognition based on frame-by-frame gait energy images and convolutional long short-term memory. International Journal of Neural Systems.

Webb, C. (2014). Power query for power BI and Excel: Apress.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. Journal of Big data, 3(1), 1-40.

Yamin, M. (2019). Information technologies of 21st century and their impact on the society. International Journal of Information Technology, 11(4), 759-766.

Yan, W. (2019) Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics. Springer London.

Yan, W. (2021) Computational Methods for Deep Learning: Theoretic, Practice and Applications. Springer London.

Zahid, M., Ahmed, F., Javaid, N., Abbasi, R. A., Zainab Kazmi, H. S., Javaid, A., . . . Ilahi,

M. (2019). Electricity price and load forecasting using enhanced convolutional neural network and enhanced support vector regression in smart grids. Electronics, 8(2), 122.

Zheng, K., Yan, Q., Nand, P. (2017) Video dynamics detection using deep neural networks. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., . . . He, Q. (2020). A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1), 43-76.