

Article

Multistep Forecasting of Power Flow Based on LSTM Autoencoder: A Study Case in Regional Grid Cluster Proposal

Fachrizal Aksan ^{1,*}, Yang Li ² , Vishnu Suresh ¹  and Przemysław Janik ¹

¹ Faculty of Electrical Engineering, Wrocław University of Science and Technology, 50-370 Wrocław, Poland; vishnu.suresh@pwr.edu.pl (V.S.); przemyslaw.janik@pwr.edu.pl (P.J.)

² Department of Energy Distribution and High Voltage Engineering, Brandenburg University of Technology Cottbus-Senftenberg, 03046 Cottbus, Germany; liyang@b-tu.de

* Correspondence: fachrizal.aksan@pwr.edu.pl

Abstract: A regional grid cluster proposal is required to tackle power grid complexities and evaluate the impact of decentralized renewable energy generation. However, implementing regional grid clusters poses challenges in power flow forecasting owing to the inherent variability of renewable power generation and diverse power load behavior. Accurate forecasting is vital for monitoring the imported power during peak regional load periods and surplus power generation exported from the studied region. This study addressed the challenge of multistep bidirectional power flow forecasting by proposing an LSTM autoencoder model. During the training stage, the proposed model and baseline models were developed using autotune hyperparameters to fine-tune the models and maximize their performance. The model utilized the last 6 h leading up to the current time (24 steps of 15 min intervals) to predict the power flow 1 h ahead (4 steps of 15 min intervals) from the current time. In the model evaluation stage, the proposed model achieved the lowest RMSE and MAE scores with values of 32.243 MW and 24.154 MW, respectively. In addition, it achieved a good R^2 score of 0.93. The evaluation metrics demonstrated that the LSTM autoencoder outperformed the other models for multistep forecasting task in a regional grid cluster proposal.

Keywords: multistep power flow forecast; LSTM autoencoder; regional grid cluster proposal



Citation: Aksan, F.; Li, Y.; Suresh, V.; Janik, P. Multistep Forecasting of Power Flow Based on LSTM Autoencoder: A Study Case in Regional Grid Cluster Proposal. *Energies* **2023**, *16*, 5014. <https://doi.org/10.3390/en16135014>

Academic Editor: Saeed Golestan

Received: 23 May 2023

Revised: 24 June 2023

Accepted: 27 June 2023

Published: 28 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Background

The reduction of greenhouse gas emissions is imperative, and a viable means to achieve this is by promoting the integration of renewable energy (RE) into power grids [1]. The proliferation of decentralized energy systems in electricity grid networks, mainly through the deployment of wind generators and photovoltaic (PV) systems, has fundamentally transformed the power supply system, transitioning it from a centralized and unidirectional structure to a decentralized and bidirectional structure. However, because a considerable proportion of such systems are connected to the power grid at low voltage levels, novel challenges arise, including issues related to energy management and bidirectional power flow [2]. This is because the variability in electrical power generation, such as the intermittency of wind intensity and solar radiation energy, results in a mismatch between the electricity demand and supply. Furthermore, the arrangement of installed renewable energy systems is heavily influenced by geographical and meteorological factors, as highlighted in the literature [3]. Hence, it is crucial to develop effective solutions to address these challenges [4].

According to the literature [2], which is our primary reference, regional analysis plays a vital role in addressing the aforementioned challenges, particularly in the context of weather-dependent renewable energies. This is because it accounts for the regional variations in power generation and consumption. A regional power grid model can facilitate

the analysis of how, where, and when to convert and store excess electrical energy in comparison to local demand. The efficacy of integrating grids into large-scale energy system models depends on the accuracy and precision of the models and their calculations. However, the current methods for assessing region-specific renewable potentials and grid reductions are primarily limited to political administration boundaries or national scales. In addition, there is a lack of local clustering methods for high-voltage substations. Therefore, reference [2] aims to contribute to the development of a postcode and high-voltage distribution network-based cluster model and analyze the regional power generation and power flow parameters.

The development of a regional grid network cluster approach is imperative for reducing the complexity of the overall power grid system and for analyzing the impact of decentralized power generation from renewable energy sources (RESs). It is incumbent upon the grid operator to monitor the power flow in the grid network cluster, given the volatility of power generation from RESs and the diverse behavior of power loads, which result in fluctuating line power flow characteristics. Additionally, the dynamic grid topologies and newly installed power generation assets, such as PV and wind systems, can lead to a significant amount of bidirectional power flow via transformers to/from the overlaid grid system. Moreover, the addition of new commercial and industrial loads to the distribution grid system contributes to a high-power transport through feedlines. Consequently, forecasting can be a proposed solution for monitoring the power flow through the network cluster. Through forecasting, we can capture critical generation-load information [5] about the network cluster, such as how much power is imported when the regional load is high and how much surplus power generation is exported from the investigated region. The detailed explanation of grid network clusters can be found in Section 3 below.

1.2. Related Work and Contribution

Accurate forecasting can be achieved through several methodologies. Notably, with the advancements in the field of computer science, the development of machine learning (ML) and deep learning (DL) has gained momentum [6–8]. Consequently, it is unsurprising that ML and DL techniques are currently being employed for all prediction and forecasting tasks in modern power systems, as evidenced by existing literature [9–12] that utilizes deep learning models for load forecasting. Furthermore, references [13–15] employ deep learning models to solve solar power forecasting, while references [4,16–18] utilize deep learning models for wind power forecasting.

Forecasting bidirectional power flow is still a topic with limited scientific papers. However, there are some research papers on power flow forecasting that are pertinent to this study. For instance, in reference [19], an extreme learning machine was used by the authors to predict vertical power flow, and a postprocessing technique was introduced to improve the forecast based on recent changes in the power flow. Accurate results were achieved by their approach, and the forecast quality was further improved by the postprocessing step. In reference [20], a long short-term memory (LSTM) network was employed by the authors to forecast vertical power flow in the interconnection between medium- and high-voltage networks. A model updating process was integrated to enhance the precision of the predictions, which was demonstrated to have superior performance compared to the non-updating approach. In another study [21], the challenge of short-term power flow forecasting on transmission inerties was effectively resolved by employing Box–Jenkins SARIMA methodology with acceptable data requirements. The authors noted that the forecasting accuracy declines over medium- to long-term periods due to low-quality input data. They discovered that SARIMA models were more precise for power flow forecasting issues with a historical pattern. The study emphasizes the significance of SARIMA methodology in power flow forecasting applications.

The analysis of power flow in an integrated power system with other power resources can offer valuable information regarding bidirectional power flow. The knowledge of bidirectional power flow is critical for power system operators in planning, maintaining,

and modifying circuits to accommodate facilities or loads that may require power during peak load or feed surplus power to the main power supply.

In this study, there is a research gap due to the existing literature focusing on forecasting power flow in power systems but lacking specific research on multistep forecasting bidirectional power flow in regional grid cluster study cases. Meanwhile, there are already some studies addressing power flow forecasting. For instance, reference [22] introduces a novel deep learning model, called multichannel long short-term memory (TL-MCLSTM) with time location, for multistep short-term power consumption forecasting in the smart grid. Similarly, reference [23] proposes a method for multistep time series forecasting utilizing LSTM-recurrent neural network (RNN). The proposed method provides several advantages such as better data pattern fitting, less manual effort, and higher predictive accuracy. A further investigation on multistep forecasting has been conducted and reported in reference [24], where the authors have employed a residual convolutional neural network (R-CNN) with multilayered long short-term memory (ML-LSTM) architecture. The proposed methodology has exhibited a substantial reduction in error rates when compared with baseline models.

Another study, presented in reference [25], provides a 2D convolutional neural network (CNN) for multistep short-term electric load forecasting. The authors found that using this model can significantly reduce the number of trainable parameters, including training time, model size, and computation requirements. Additionally, a similar study mentioned in reference [14] also utilized a CNN and combined it with a chaotic optimization algorithm for multistep short-term solar radiation forecasting. The authors claim that this model can achieve accuracy and robustness, thereby improving the guidance for power grid dispatching. In [26], the multistep forecasting task on electricity load was solved by using a hybrid gated recurrent unit (GRU) with a feedforward neural network. The authors mentioned that the proposed approach can achieve better results compared to other methods in predicting the demand for charging stations in the short-term horizon.

Previous literature reviews have mentioned that deep learning is a powerful method that can be used to overcome multistep forecasting task, however, there is still a need for more research in this domain. Additionally, there is a gap in comparative analysis between different deep learning model variants for multistep forecasting of power flow in regional grid clusters. To address the research gap mentioned above, this study has the research objective of developing a multistep forecasting approach for power flow within a regional grid cluster, specifically dedicated to bidirectional power flow. The multistep forecasting approach in this study is designed for four steps with a 15 min interval, which corresponds to forecasting 1 h ahead. The reason for choosing this interval is that we are working with a power measurement dataset that has a 15 min interval. Forecasting 1 h ahead is intended to capture critical generation-load information [5] about the network cluster, such as the amount of power imported during high regional load and the surplus power generation exported from the investigated region.

The objective of this research is to conduct a multistep forecasting of power flow within a regional grid cluster through the utilization of LSTM autoencoder, which is a variation of the LSTM family of models. Several studies have employed the deep learning model for multistep forecasting. Table 1 provides a comprehensive overview of the literature on power flow forecasting and the utilization of deep learning models for multistep forecasting. The existing literature suggests that the LSTM model is highly effective in short-term forecasting. Nevertheless, there are numerous LSTM model architectures that can be implemented for the same purpose. In the current study, we introduce LSTM autoencoder as an appropriate model for multistep forecasting of power flow. This paper provides some technical contributions as follows:

- This study performs a comparative analysis between LSTM autoencoder and four distinct LSTM family architectures for multistep forecasting, which, as far as the authors are aware, have not been subject to a comparative analysis in prior literature.

- This study presents a 1-h-ahead (four steps of 15 min intervals) forecasting approach for power flow specifically tailored for a regional grid cluster application.

Table 1. Summary of related work.

Topic	Reference	Methodology/Output
Power flow forecasting	Jost et al. [19]	Using an extreme learning machine postprocessing technique to forecast the vertical power flow.
	Brauns et al. [20]	Using LSTM model with updating process for vertical power flow forecasting.
	Paretkar et al. [21]	Implementing Box and Jenkins ARIMA for predicting power flow in the short term on significant transmission interconnections.
Multistep forecasting-based LSTM	Shao et al. [22]	Using TL-MCLSTM for multistep short-term power consumption forecasting.
	Liu et al. [23]	Utilizing LSTM RNN for multistep time series forecasting.
	Alsharekh et al. [24]	Employing R-CNN with ML-LSTM for multistep forecasting.
	Sing et al. [25]	Using 2D CNN for multistep short-term electric load forecasting.
	Duan et al. [14]	Proposing CNN with chaotic aquila optimization algorithm for multistep short-term solar radiation forecasting.
Our approach (multistep forecasting of power flow)	Cheng et al. [26]	Combining GRU model and feedforward neural network for multistep electricity load forecasting.
		Proposing LSTM autoencoder for multistep forecasting of power flow.

The subsequent sections of this paper are organized as follows: Section 2 presents a succinct summary of the deep learning model architectures employed. Section 3 elaborates on the case studies pertaining to grid network cluster and the dataset utilized, while Section 4 delineates the proposed methodology. Section 5 showcases the outcomes and corresponding discussions, and the Section 6 concludes the paper with closing remarks.

2. Deep Learning Model

2.1. Long Short-Term Memory (LSTM) Structure

A recurrent neural network (RNN) is a type of deep learning model that is particularly well suited for processing sequential or time series data [27]. Due to its capacity for learning from training data, the RNN is frequently employed in solving ordinal or temporal problems. The RNN distinguishes itself from other deep learning models by incorporating a memory mechanism that allows it to leverage information from past inputs to influence the present input and output, in contrast to other models that assume independence between inputs and outputs. The recurrent neural network is notorious for its susceptibility to the issues of exploding and vanishing gradients [28], which arise due to the backpropagation through time (BPTT) algorithm employed by the RNN to compute gradients during the training process. These problems can cause suboptimal performance and slow training times for RNNs. To mitigate these issues, alternative models such as the long short-term memory (LSTM) and gated recurrent unit (GRU) models have been developed.

Hochreiter and Schmid Huber [29] initially introduced the long short-term memory model to address the issue of long-term dependence and alleviate the vanishing gradient problem, which is not feasible with the standard RNN model. The LSTM (see Figure 1) model is designed with memory cells and gates to effectively manage information flow and retain information over extended periods. As a result, the LSTM has become a popular deep learning model that is applied in a wide range of prediction and forecasting tasks.

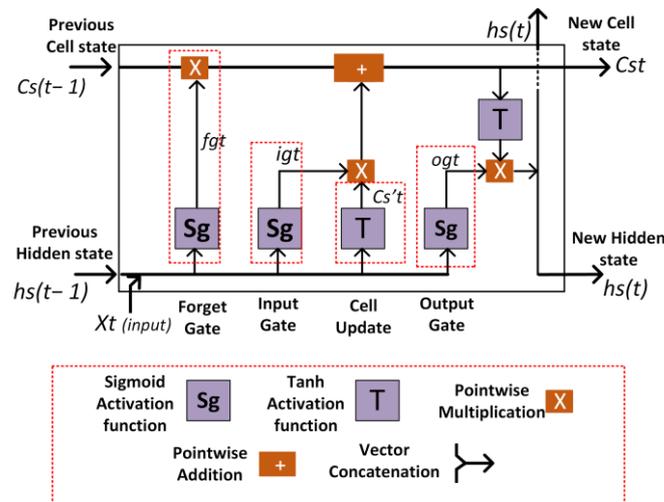


Figure 1. The architecture of LSTM Network.

Broadly speaking, an LSTM network comprises memory blocks called cells, each having two states: the cell state and the hidden state. The LSTM network utilizes these cells to make critical decisions by selectively retaining or discarding information about significant components [7]. These components, called gates, are structured into forget gates, input gates, and output gates. As depicted in Figure 1, the LSTM model operates in three stages: during the first stage, the network employs the forget gate to determine which information is to be retained or discarded for the cell state. This process involves the input at the current time step (x_t) and the previous hidden state value ($hs_{(t-1)}$), both of which are subjected to the sigmoid function (Sg). The calculation for the forget gate (fg_t) is expressed as follows.

$$fg_t = Sg\left(wf \cdot [hs_{(t-1)}, x_t] + b_f\right) \quad (1)$$

During the second phase, the network's calculation persists by transforming the previous cell state, ($Cs_{(t-1)}$), to a new cell state, (Cs_t). This operation involves the selection of updated information that needs to be incorporated in the long-term memory (cell state). The updated cell state is obtained by considering the input gate (igt), forget gate, and cell update gate values (Cs'_t). The mathematical equations for determining the output values of these gates are illustrated below.

$$igt = Sg\left(wi \cdot [hs_{(t-1)}, x_t] + b_i\right) \quad (2)$$

$$Cs'_t = T\left(wc \cdot [hs_{(t-1)}, x_t] + b_c\right) \quad (3)$$

$$Cs_t = \left(Cs_{(t-1)} \cdot fg_t\right) + \left(igt \cdot Cs'_t\right) \quad (4)$$

Upon the completion of cell state updating, the final step entails ascertaining the value of the hidden state, (hs_t), which acts as the network's memory by retaining the previous data and facilitating predictions. To achieve this, the calculation process must incorporate the reference value of the updated cell state and the output gate (og_t). The formula that characterizes this process is presented below.

$$og_t = Sg\left(wo \cdot [hs_{(t-1)}, x_t] + b_o\right) \quad (5)$$

$$hs_t = og_t \cdot T(Cs_t) \quad (6)$$

The foregoing equations pertain solely to a discrete time interval. Consequently, these formulas necessitate recalculation for the ensuing time increment. Accordingly, in the event of a 24-step series, the aforementioned equations must be recomputed 24 times for each temporal phase, respectively.

The weight matrices (w_f, w_i, w_c, w_o) and biases (b_f, b_i, b_c, b_o) are stationary parameters, lacking temporal dependence. Hence, these matrices remain unaltered across successive time increments, that is, they persist as constants throughout the computation of output sequences for varying timesteps.

2.2. LSTM Autoencoder

The LSTM autoencoder is a specific type of autoencoder that is designed to handle sequential data by incorporating LSTM layers [13]. This architecture, as depicted in Figure 2, is widely used in sequence-to-sequence tasks, such as time series forecasting. The input sequence is encoded by the first LSTM layer, which learns a compressed representation of the data. A dense layer can be added after the LSTM layer to extract essential features from the encoded representation before passing it to the repeat vector layer. The repeat vector layer repeats the encoded representation multiple times, enabling it to be decoded back into the original sequence format.

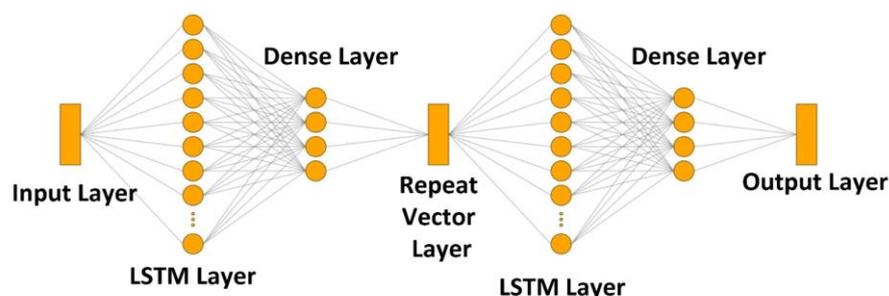


Figure 2. The LSTM Autoencoder Layers.

The second LSTM layer decodes the repeated vector and reconstructs the original sequence. Refining the reconstructed sequence and improving its fidelity to the input can be achieved by adding another dense layer after the LSTM layer. It is worth noting that the number of units and layers used in each LSTM and dense layer can vary depending on the specific task and data under consideration. Moreover, it is critical to train the model with an appropriate loss function because the loss function is a part of the optimization algorithms. It is used to estimate the loss of the model, allowing the weights to be updated and reducing the loss in subsequent evaluations. Additionally, different loss functions can have varying impacts on deep learning models as they capture different aspects of the optimization problem. Therefore, the choice of loss functions depends on the specific task and behavior of the model [30].

3. Grid Network Cluster and Power Flow Dataset

3.1. Grid Network Cluster

The organization of power grids into distinct voltage levels enables the efficient transmission and distribution of electrical energy across various equipment, such as transformers and transmission lines. However, the recent proliferation of dynamic grid topologies and the installation of renewable power generation systems, such as photovoltaic (PV) and wind systems, within distribution systems have introduced bidirectional power flow through transformers and posed significant challenges to the overlaid grid system. This has been further exacerbated by the increasing usage of feedlines by new commercial and industrial loads in distribution grids, contributing to high power transport. The inherent variability of power generation from renewable energy resources and the diverse behavior of power loads have made power flow forecasting a formidable task.

To address these challenges, a regional grid network cluster has been developed to simplify the power grid system and facilitate the analysis of decentralized power generation from renewable energy sources, referring to our preliminary result in the literature [2]. This cluster is designed to be located at the connection point between the transmission system operator (TSO) and the distribution system operator (DSO) through a grid reduction procedure, enabling a comprehensive analysis of power generation and consumption patterns and the loads on the power lines. This analytical tool offers valuable insights into the behavior of power systems under different scenarios and conditions and provides a basis for designing, optimizing, and predicting local power systems while integrating different generation and consumption sources. Similarly reference [31] proposed a clustering of power networks to decompose a large interconnected power network into smaller loosely coupled groups to facilitate easy and flexible management of the power transmission systems by allowing secondary voltage control at regional levels and controlled islanding that aims to prevent the spreading of large-area blackouts. Another study [32] proposed power grid network partitioning and clusters for a splitting a power grid system into separate parts with self-sufficient power generation. Internal connectivity is maximized within the individual clusters and they minimize the power deficiency or surplus.

The importance of grid network clusters extends beyond the analysis of existing power systems, as they can also aid in the design and optimization of power systems and the prediction of power exchange between external grid systems [2]. Figure 3 illustrates an example of a regional electrical grid topology that encompasses low-voltage (LV), overlaid medium-voltage (MV), and high-voltage (HV) levels, under the distribution grid system. The region receives power supply from two connected substations, and the circle area delineates one network cluster. Within this network cluster, a multitude of power generations and loads are aggregated from different voltage levels. Our study focuses on the feedlines from both sides of the network cluster, which consists of six feedlines supplied by two connected substations, as detailed in the literature [2]. By measuring the power flow in the feedlines, researchers and grid system operators can gain a better understanding of the system's behavior and identify the potential power balance between local power generation and consumption.

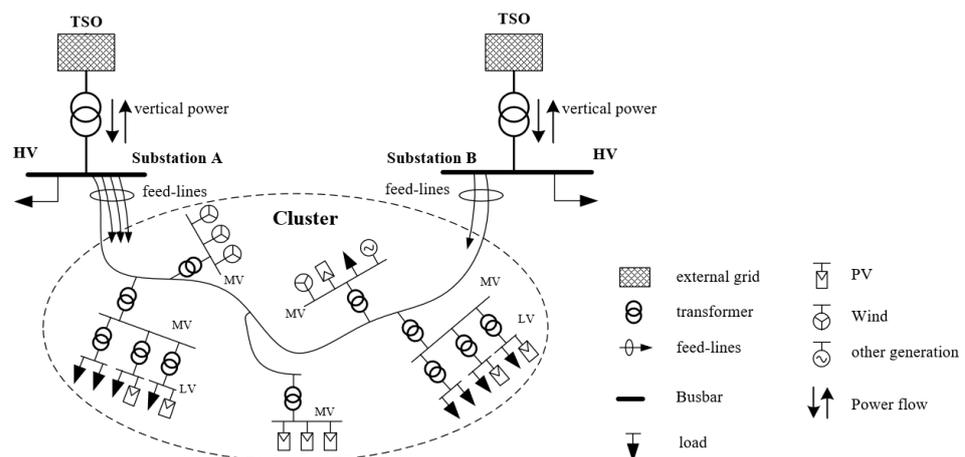


Figure 3. Network Topology.

3.2. Bidirectional Power Flow Dataset

This study focuses on a regional high-voltage subnet situated in the north-east region of Germany, which has already been documented in the literature [2]. For our investigation, we utilized a simplified grid, depicted in Figure 4, which is a visual representation of a network cluster comprising six feedlines that supply power to and receive power from two interconnected substations, namely Sub_A and Sub_B. Four feedlines (Line 3, Line 4, Line 5, Line 6) are connected to Sub_A, while two feed lines (Line 1, Line 2) are connected to Sub_B. Based on the simplified grid, the actual implementation involves the interconnection of

lines in the grid in a parallel manner. Specifically, Line 1 and Line 2 are parallel, Line 3 and Line 4 are parallel, and Line 5 and Line 6 are parallel. Consequently, based on observations from data measurements, it has been inferred that the parallel lines in the cluster exhibit similar power flow patterns, which are distinct from the remaining lines, as illustrated in Figure 5.

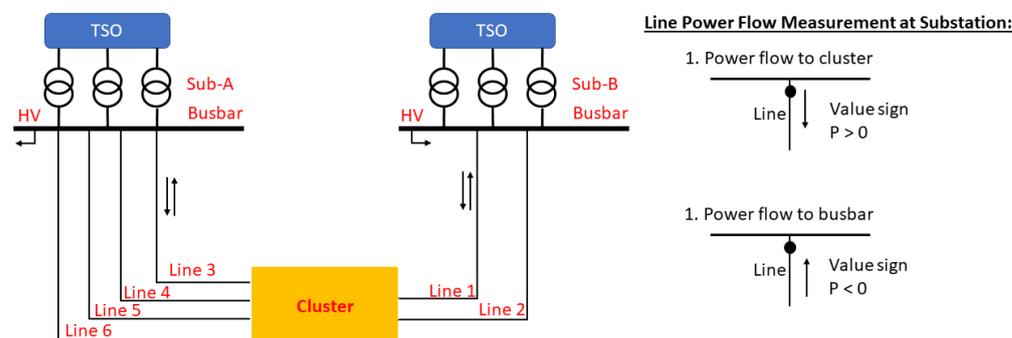


Figure 4. Simplified Network Cluster.

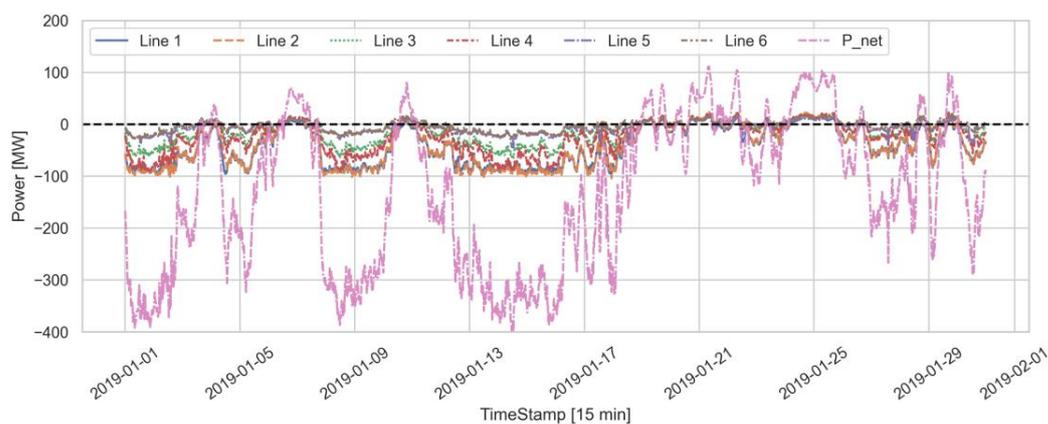


Figure 5. Power flow in all lines of the network cluster.

The power measurement of the feeder lines enables us to acquire vital information about the generation and load of the grid cluster, such as the amount of power imported during periods of high regional load and the quantity of surplus generation exported from the cluster under investigation [2,7]. In this study real power measurement data are utilized to analyze and predict the regional power balance. To this end, we acquire directional feedline power measurement data with a 15 min temporal resolution from the local distribution system operators. These directional power flow data span from 1 January 2019 to 31 December 2019, and an instance of power flow in the network cluster studied in January 2019 is presented in Figure 5. The sign of the active power indicates the direction of power flow between the busbar and the cluster since the power measurement is bidirectional. A positive value signifies that power flows from the busbar to the cluster, while a negative value indicates power flow from the cluster to the busbar. Conversely, negative active power values indicate the exported power from the cluster to the busbar, while positive values indicate the import of power from the busbar to the cluster.

In this study, the primary objective is to predict the power net 1 h in advance, which refers to the total power flow of all feedlines in the investigated network cluster. The power net denotes the power flowing either from the busbar to the cluster or from the cluster to the busbar. As illustrated in Table 2 which shows an example of the dataset used, the dataset contains the power values from all feedlines and the power net in this study. Mathematically, the power net is calculated at a specific point in time (i) by summing up

the power flowing at a specific time (i) through Line 1, Line 2, Line 3, Line 4, Line 5, and Line 6. The corresponding equation is presented below.

$$P_{net_i} = \sum (Line1_i + Line2_i + Line3_i + Line4_i + Line5_i + Line6_i) \tag{7}$$

Table 2. Example of bidirectional dataset used.

Timestamp	Line 1	Line 2	Line 3	Line 4	Line 5	Line 6	P_Net
2019-01-01 00:00:00	-58.285	-56.291	-16.162	-21.027	-7.297	-7.743	-166.805
2019-01-01 00:15:00	-60.758	-59.467	-19.703	-27.23	-7.297	-9.297	-183.752
2019-01-01 00:30:00	-65.043	-62.977	-20.811	-31.649	-6.851	-9.96	-197.291
2019-01-01 00:45:00	-68.495	-65.522	-20.365	-28.77	-9.068	-11.068	-203.288
2019-01-01 01:00:00	-68.531	-67.661	-27.446	-36.527	-10.405	-12.608	-223.178

4. Proposed Methodology

In this study, relevant primary data on bidirectional power flow were gathered from an examined power grid, and data cleansing and filtration were conducted prior to their application. The resulting high-quality data facilitated the training and testing of the proposed deep learning model for power flow forecasting in a simplified network cluster. The proposed methodology comprises three main categories after the data collection stage: data preprocessing, model construction, and model evaluation. An overview of the proposed methodology is illustrated in Figure 6.

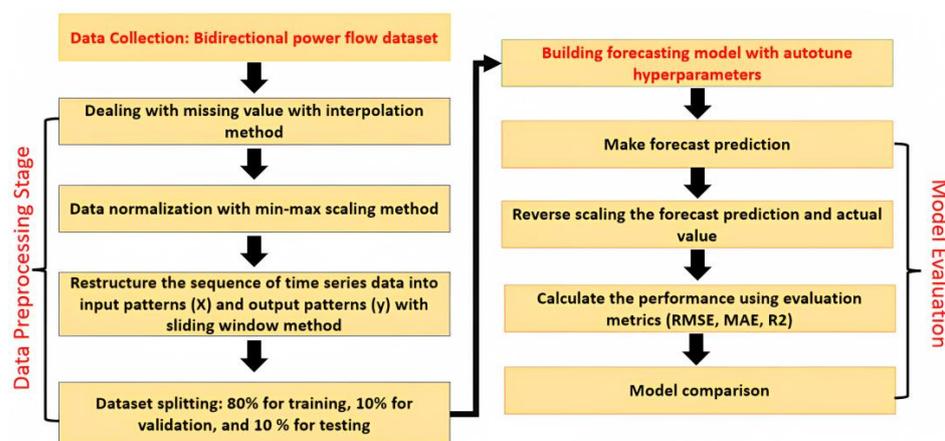


Figure 6. Proposed Methodology Overview.

4.1. Data Collection and Data Preprocessing

Data collection is a crucial step because all further steps depend on the availability of the data. It involves gathering all the necessary data from available sources. In this study, we solely utilized univariate time series data of the total bidirectional power flow of all feedlines in the investigated network cluster (power flow net). The reason for this is the lack of data availability for other external inputs, such as weather variables. Moreover, our study indicates that weather data have no strong correlation with the power flow net. This is due to the regional grid network including a combination of inherent variability in power generation from renewable energy resources and the diverse behavior of power loads.

After the data collection step, data preprocessing plays a pivotal role in transforming raw data into a compatible format for deep learning models by incorporating various techniques. In the present study, diverse methodologies were employed, including handling missing values, data normalization, sliding window, and dataset partitioning.

4.1.1. Step 1: Dealing with Missing Values

As the measurement data collected may contain missing values, which may result from device measurement malfunctions or errors in data collection, it is essential to address them to prevent potential sampling bias. Moreover, forecasting models typically require continuous and complete time series data [33], making it necessary to handle missing values appropriately. In this study, we employed the interpolation method to fill in missing values in the time series dataset by estimating them based on neighboring data points.

4.1.2. Step 2: Data Normalization

The dataset used in this study comprises bidirectional power flow data with varying scales. This difference in scale can have an impact on the performance of deep learning models during the learning process [34]. Therefore, it is necessary to normalize the dataset to mitigate this issue. In this study, we employed the numerical scaling method of min–max normalization. The formula for converting the original values to normalized values is shown in the following Equation (8).

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (8)$$

The equation for normalizing a value, denoted as x' , is based on the original value, x , as well as the maximum value of x ($\max(x)$) and the minimum value of x ($\min(x)$).

4.1.3. Step 3: Sliding Window

Following the normalization of the dataset, a sliding window technique was employed to convert the structured time series data into a supervised learning format comprising multiple subsequences [35]. This approach was necessary as the forecasting model aimed to address a supervised learning problem, where the dataset must include input patterns (x) and output patterns (y). The sliding window approach leveraged the previous time step as the input variable and the value of the following time step as the output variable. This process involved sliding a window of a fixed size along the time series dataset to generate multiple subsequences.

The primary objective of this study was to forecast power flow 1 h ahead. As a result, the time series data were transformed into the necessary format for multistep forecasting, which involves predicting multiple future time steps in a sequence. The optimal length of the input and output variables when utilizing a sliding window approach for the forecasting task depends on several factors, such as the specific time series data, patterns, and dependencies in the data. Therefore, there is no specific answer to the significant optimal length of input and output variables. However, there are considerations that can be implemented to select the lengths of the input and output variables.

One such consideration is the data granularity factor. The granularity factor can impact the window size. If the data have fine-grained observations (such as hourly or daily), a smaller window size may be needed to capture relevant patterns. On the other hand, if the data are aggregated at a higher level (e.g., monthly or yearly), a larger window size may be necessary. In this study, the length of the input and output variables was determined based on the granularity factor and considering computational time. It was recognized that a larger or smaller window size can introduce different computational time requirements during the model training stage. Based on our observation, the last 6 h (24 steps of 15 min intervals) of the time series data following the current time were used as input data. The value 1 h ahead (four steps of 15 min) of the current time was used as the output. The sliding window approach adopted is illustrated in Figure 7, where the yellow bar represents the length of the input variable and the red one represents the output variable. While blue bar represents current time.

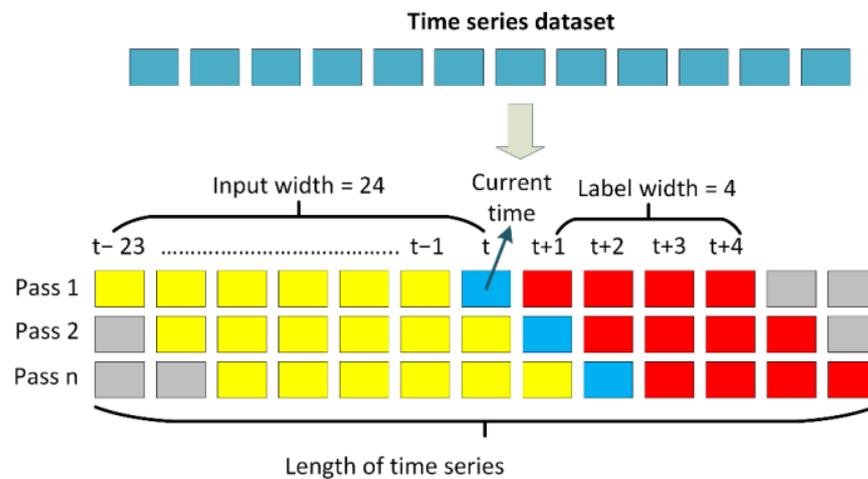


Figure 7. Sliding window approach.

4.1.4. Step 4: Dataset Splitting

Data splitting is a crucial stage that involves dividing a dataset into training, validation, and testing sets. The training dataset is used to train a deep learning model, while the validation dataset is used to evaluate the performance of the model during the training process. Moreover, the testing dataset is used to assess the final performance and generalization capabilities of the trained model. In this study, we performed data splitting after reorganizing the structure of the time series dataset into a supervised learning format. There is no optimal percentage ratio for splitting the dataset. However, existing references indicate several ways to divide the dataset. For example, in references [15,36], the studies provide information on splitting the dataset with a ratio of 90% for training and 10% for testing. In [7,37–41], a scenario of 70% for the training dataset, 15% for the validation dataset, and 15% for the test dataset is used. Based on these references, our research study specifically allocated 80% of the total dataset for training, 10% for validation, and 10% of the data for testing purposes.

In the splitting process, our study does not recommend dividing the dataset randomly for training, validation, and testing when performing forecasting tasks. This is because the time series data used in this study have a temporal order, and the goal is to make predictions on future data based on past observations. Based on this reasoning, randomly shuffling and splitting the dataset can lead to invalidating the forecasting task due to future information leakage and performance estimation bias.

4.2. Model Construction with Autotune Hyperparameter

During the model construction stage, we developed several baseline models to assess the proposed model's performance. The common baseline models used for forecasting tasks have included simple RNN [8,42], LSTM [43,44], GRU [45,46], and bidirectional LSTM [13,47], whereas the proposed model was an LSTM autoencoder model. All models used in this study were developed based on the TensorFlow [48] and Keras libraries [49]. The designs of all structures and layers of the models can be observed in Table 3. During the development of training, all deep learning models were built with autotune hyperparameters. The main reason for this was to automatically search for the optimal values of hyperparameters, providing benefits to deep learning models such as improved performance, time efficiency, and resource efficiency.

Table 3. Deep Learning Model structures.

DL Model	Structure Layers of Model
Simple RNN	Simple RNN layer + Dense layer
LSTM	LSTM layer + Dropout layer + Dense layer
GRU	GRU layer + Dropout layer + Dense layer
Bidirectional LSTM	Bidirectional LSTM layer + Dropout layer + Dense layer
LSTM Autoencoder	LSTM layer + Dense layer + Repeat vector layer + LSTM layer + Dense layer

4.3. Model Evaluation

The process of model evaluation is a crucial step in assessing the precision and performance of all compared models using metric scores. In this study, prior to implementing model evaluation, the prediction results from the models and testing dataset were transformed into their original values, since their prior form was in a normalized state.

In this study, the selection of evaluation metrics was based on recommendations derived from previous research and reports in the domain of predictive modeling. These metrics encompassed the root mean square error (RMSE) [7,13], which is used to calculate the square root of the average of the squared differences between the predicted and actual values, the mean absolute error (MAE) [3,50], which measures the average magnitude of the errors without considering their direction, and the coefficient of determination (R^2) [51,52], which measures the proportion of the variance in the dependent variable that is explained by the independent variables in the model.

When evaluating forecasting models, RMSE and MAE are metrics typically used to assess the accuracy of model predictions, and a lower value of these metrics indicates better performance of the trained model. In contrast, R^2 is used to evaluate the overall quality of the model and assess how well it explains the variation in the data. A higher score of the R^2 metric indicates a better fit of the model. The formulas for computing these metrics used in this study are illustrated in the following equations.

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^N (O_t - \hat{O}_t)^2}{N}} \quad (9)$$

$$\text{MAE} = \frac{\sum_{t=1}^N |O_t - \hat{O}_t|}{N} \quad (10)$$

$$R^2 = \frac{\sum_t (\hat{O}_t - \bar{O})^2}{\sum_t (O_t - \bar{O})^2} \quad (11)$$

The actual value O at time t is denoted as O_t and the predicted value as \hat{O}_t , where \bar{O} is the mean value of O , and N is the total number of observations. In this study, all metric evaluations used were based on the scikit-learn library [53].

5. Results and Discussion

5.1. Comparison of Deep Learning Models in Training Stage

This section presents the results of our proposed LSTM autoencoder model compared with the baseline models during the training stage. All models were designed to forecast the net power flow value of a network cluster 1 h ahead. The structures of all the models compared in this study are based on the information provided in Table 3.

This study employed the autotune hyperparameter technique with the hyperband algorithm from the Keras tuner to optimize the deep learning models. The application of this technique encompasses both the baseline models and the proposed LSTM autoencoder throughout the model training stage. During the model development and training stages in this study, the autotuning hyperparameter technique was used to search for optimal values of key hyperparameters, such as the number of neurons in the hidden layers, preferred activation function, and appropriate learning rate value for the optimization method. This allowed us to automatically determine the most suitable configurations for these hyperparameters and optimize the performance of the deep learning models.

The utilization of the autotune hyperparameter technique yields significant advantages, including improved efficiency in the model training process and enhanced model quality [54]. Moreover, it mitigated the need for laborious manual exploration of numerous hyperparameter combinations, which are often unavailable and require meticulous selection for deep learning models. The hyperband algorithm as a tuner was adopted in this study to optimize the hyperparameters in our deep learning models. This algorithm utilizes a successive halving method to iteratively eliminate poorly performing configurations [55]. By employing the hyperband algorithm with appropriate settings during the model training phase, the hyperparameter space was efficiently explored, resulting in identification of the optimal configuration for our deep learning models. In this study, a hyperband tuner was configured to minimize the validation loss. In addition, the maximum number of epochs for training each model configuration was set to ten, and the algorithm employed a factor of three to determine the number of configurations in each bracket.

After identifying the optimal hyperparameter configurations for all the models, the models were trained using the training and validation datasets. The original structure of these datasets consists of time series data representing the total power flow of all feedlines (power net value) in the investigated network cluster. The training dataset comprised 80% of the total time series data, covering a period from 1 January 2019, with a 15 min interval, to 20 October 2019, at midnight. The validation dataset comprised 10% of the data, spanning 20 October 2019, at 12:15 a.m. with a 15 min interval, to 25 November 2019, at 11:15 a.m. These percentages indicate the ratio of the time series data used in this study. Furthermore, the structure of the datasets was converted into a supervised learning format by employing the sliding window technique, enabling them to be fed into the deep learning model. In terms of dataset size after their size was reorganized, the training dataset comprised 28,010 samples, each consisting of 24 time steps and 1 feature. Similarly, the validation dataset contains 3501 samples, with each sample encompassing 24 time steps and 1 feature.

During the model training stage, all models with configured hyperparameters were trained on the computer listed in Table 4. The models were executed and fitted with a configuration in which the number of epochs was set to 100, and the batch size was set to 32. Furthermore, during model compilation, all models were set with the Adam optimizer and mean squared error (MSE) as the loss function. After the training process, the loss value was recorded to provide an indication of how well the model learned from the training data. In this section, the MSE loss function was used to calculate the average squared difference between the predicted and actual values. Figure 8 was constructed to monitor the performance of the model on both the training and validation data, where the x-axis represents the number of training epochs and the y-axis represents the loss value.

Table 4. Machine Specification.

Parameter	Specification
CPU	12th Gen Intel® core™ i7-12650h
GPU	NVIDIA GeForce RTX 3060 6 GB
HDD/SDD	500 GB
RAM	16 GB
OS	Windows 11 Home 64 bit

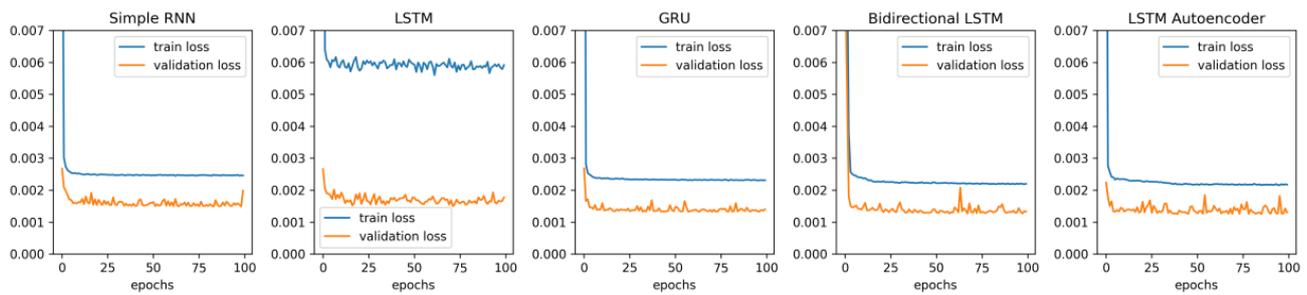


Figure 8. Learning Curve in Training and Validation dataset.

The training loss curve displays the loss function evaluated for the training data during each training epoch, whereas the validation loss curve displays the loss function evaluated for the validation data during each epoch. As depicted in Figure 8, all models generally exhibit a decrease in loss values over several epochs in the training dataset but a fluctuation in the validation dataset. The LSTM model had the highest loss value during training, whereas the other models were similar and tended to have small loss values. In the validation dataset, it is challenging to determine which model performs well with a small loss value, as all models tend to have fluctuating loss values throughout all epochs. This learning curve can diagnose the presence of an insufficient representation of the validation dataset, which implies that the data provided are inadequate for evaluating the model's generalization capability. This scenario can be identified by observing the learning curve, where the training loss curve appears to be a suitable fit, while the validation loss curve displays erratic fluctuations around the training loss [56].

Monitoring the duration of the training process is vital for assessing the efficiency of models. It allows us to gain insights into the duration of model training and identify potential issues that may arise, such as the need for adjustments in batch size to improve training efficiency. Valuable insights can also be obtained regarding the performance and behavior of the models during the training process. Figure 9 shows that the simple RNN model had a prolonged training duration, which may be attributed to the inherent vanishing gradient problem, as discussed in Section 2. Conversely, bidirectional LSTM and LSTM autoencoder models have similarly extended training durations due to their complex architecture, as compared to simple LSTM and GRU models.

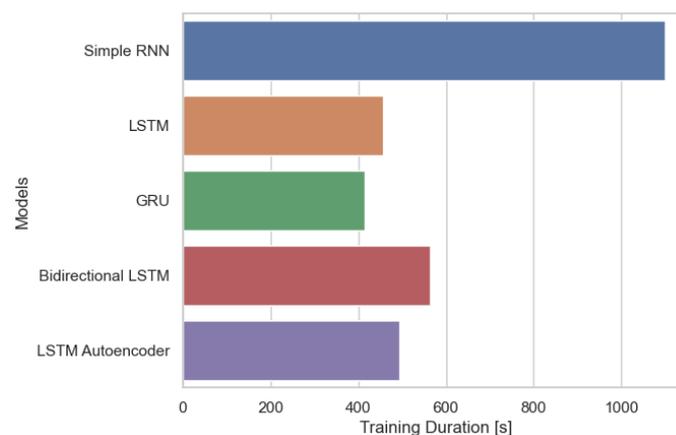


Figure 9. Training Duration of Forecasting Model.

5.2. Performance Comparison of Deep Learning Models

In this section, we present an evaluation of the performance and generalization ability of the trained model on a new dataset, referred to as the testing dataset, which was not used during the training or validation phases. The main objective of this stage is to assess the expected performance of the model in real-world scenarios. To achieve this, we employed several metrics to evaluate the models, including the RMSE, MAE, and R^2 .

In this study, we developed a function to evaluate the performance of a deep learning model. This function considers the input features of the test dataset (x_{test}) and the corresponding output features, labeled y_{test} , which serve as the ground truth. This function allows the trained forecasting model to make predictions based on the input features. The prediction results, along with the ground truth, were converted to their original scales. Subsequently, the function iterates a certain number of times in a loop, allowing for the individual evaluation of each element using relevant evaluation metrics.

In Table 5, we present the evaluation results of each trained deep learning model using the testing dataset. It is evident that the proposed LSTM autoencoder model outperforms the other models in terms of the RMSE and MAE metrics, achieving the lowest scores of 32.243 MW and 24.154 MW, respectively. Additionally, the R^2 score indicated that this model demonstrated a higher value, further confirming its superior performance. However, it is important to note that among all the compared models, the GRU model exhibits similarities to the proposed model, as it obtains the second-lowest scores in terms of RMSE and MAE, while also having the same score in the R^2 metric as the proposed model. Nevertheless, when comparing the training time, it can be observed that the LSTM autoencoder model requires a slightly longer training process compared to the GRU model.

Table 5. Performance Evaluation of Forecasting Model.

Model Name	RMSE	MAE	R^2
Simple RNN	36.238	28.127	0.912
LSTM	38.646	29.398	0.9
GRU	32.377	24.352	0.93
Bidirectional LSTM	32.486	24.552	0.929
LSTM Autoencoder	32.243	24.154	0.93

In Figure 10, we display an example of the multistep power forecasting results of the bidirectional power flow from all trained models, including the proposed model and baseline models. The testing input dataset used in this section covers the last 6 h, consisting of 24 steps with a 15 min interval, following the current time. The time span of the input data ranges from ‘2019-11-29T10:45:00’ to ‘2019-11-29T16:30:00’. The input data provided to all trained models are expected to predict the bidirectional power flow 1 h ahead (four steps of 15 min) after the current time in the investigated network cluster. The expected output data cover the time span from ‘2019-11-29T16:45:00’ to ‘2019-11-29T17:30:00’.

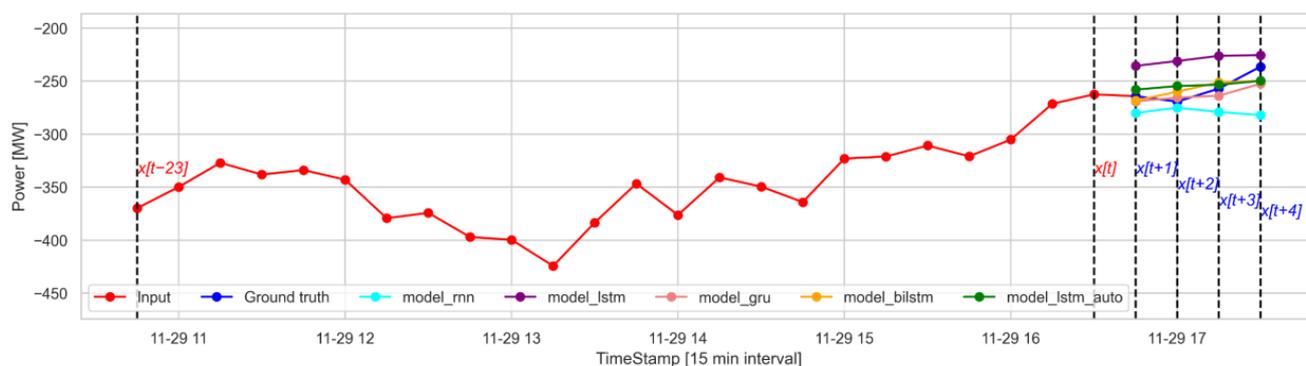


Figure 10. Multistep power flow forecast of all trained models.

Given the superior performance of our proposed model compared with other baseline models, as demonstrated in the model evaluation results, we present an extended forecast using our LSTM autoencoder. This extension involves expanding the test dataset to capture an additional four steps of 15 min interval forecast results from a moving window of the

input dataset. The primary objective of this process is to provide enhanced insights and establish greater credibility in the forecasted results.

In Figure 11, we depict the continuation of the output forecast results for power flow in the network cluster. Specifically, we examine the scenario in which the input values transition every hour (consisting of four steps of 15 min intervals) following the starting time of ‘2019-11-29T10:45:00’ as illustrated in Figure 10.

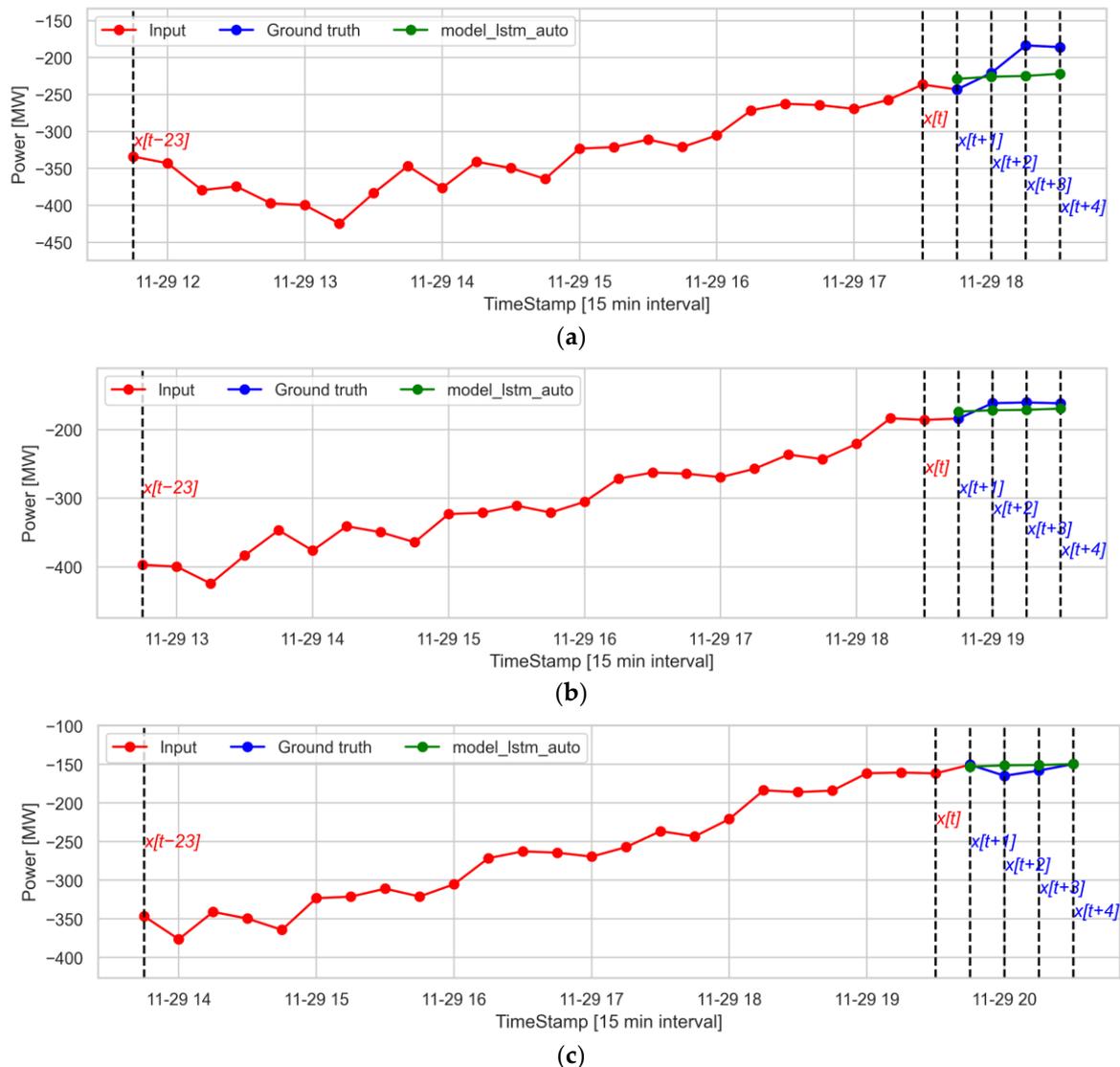


Figure 11. Multistep power flow forecasts generated by an LSTM autoencoder on 29 November 2019, for specific time intervals: (a) 17:45:00–18:30:00, (b) 18:45:00–19:30:00, and (c) 19:45:00–20:30:00.

Figure 11a showcases the utilization of the proposed model with input data ranging from ‘2019-11-29T11:45:00’ to ‘2019-11-29T17:30:00’ to predict the four values encompassing the period from ‘2019-11-29T17:45:00’ to ‘2019-11-29T18:30:00’.

In Figure 11b, the LSTM autoencoder generates predictions for power flow 1 h ahead, spanning from ‘2019-11-29T18:45:00’ to ‘2019-11-29T19:30:00’. The input data used for this prediction correspond to the interval from ‘2019-11-29T12:45:00’ to ‘2019-11-29T18:30:00’.

Furthermore, Figure 11c exhibits the forecasted results of the LSTM autoencoder for predicting power flow 1 h ahead, covering the time range of ‘2019-11-29T19:45:00’ to ‘2019-11-29T20:30:00’. To accomplish this prediction, the input data employed encompass the interval from ‘2019-11-29T13:45:00’ to ‘2019-11-29T19:30:00’.

According to the model evaluation results, our proposed model showed good performance, as indicated by lower scores on evaluation metrics, such as RMSE and MAE, and a high score of R^2 . Although the LSTM autoencoder is not a novel model, it has been employed in various studies and has consistently demonstrated a good performance. For example, in [13], the LSTM autoencoder was used to forecast 1 h ahead for solar power for participants in the intraday electricity market. The model achieved impressive performance with average RMSE and MAE values of 12.87 kW and 6.91 kW, respectively. Another study [57] also demonstrated the superiority of the LSTM autoencoder for power load forecasting. This model integrated long-term and short-term features of the samples and exhibited better performance with an MAE score of less than 52 MW when comparing the prediction results to the actual load values in the results.

6. Conclusions

The proposed regional grid cluster simplifies the power grid and facilitates the analysis of decentralized power generation. It is placed between the TSO and DSO via grid reduction, providing insights into power systems under different scenarios and aiding in the design, optimization, and prediction of local power systems. However, forecasting power flow is difficult because of the variability in renewable energy and diverse power loads.

Within the confines of this research endeavor, our proposition entails the development and application of an LSTM autoencoder with the explicit aim of forecasting power flow patterns, encompassing the estimation of both exported and imported power, for a horizon spanning 1 h. Notably, our approach involves predicting multiple future steps at 15 min intervals. To compare the performance of the proposed model, we developed several baseline models, such as a simple RNN, LSTM, GRU, and bidirectional LSTM. In the training stage, all models were trained using an autotune hyperparameter approach to optimize the selected hyperparameters for each model. The training learning curve indicates that only the LSTM model has a higher score, whereas our proposed model and other baseline models tend to have similar performance. However, in the validation learning curve, all models exhibited fluctuations, indicating issues related to the insufficient representation of the validation dataset. Therefore, future studies should aim to address this issue. Regarding training duration, the simple RNN model has the longest duration owing to the exploding and vanishing gradients issue, as mentioned in some studies. Similarly, our proposed model and bidirectional LSTM tend to have longer durations compared to LSTM and GRU because of complexity in the model structure.

During the model performance evaluation stage, we assessed all models using three metrics: RMSE, MAE, and R^2 . Our findings indicate that the proposed deep learning model, the LSTM autoencoder, outperforms the other models with lower RMSE and MAE scores, as well as a good score in R^2 , demonstrating superior performance. In addition to its performance, our proposed model has an acceptable training duration and performs well on the learning curve. Therefore, our proposed model can be a viable solution as a forecasting model for the challenging task of monitoring exported and imported power in the regional grid cluster proposal.

Author Contributions: Conceptualization, F.A. and Y.L.; proposed methodology development, F.A. and Y.L.; software development, F.A. validation, F.A., Y.L., P.J. and V.S.; investigation, Y.L.; resources, P.J.; data curation, Y.L. and P.J.; writing—original draft preparation, F.A.; writing—review and editing, F.A. and Y.L.; visualization, F.A. and Y.L.; supervision, V.S. and P.J.; project administration, P.J. and V.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data are not publicly available due to the policy of the associate company.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sen, S.; Ganguly, S. Opportunities, barriers and issues with renewable energy development—A discussion. *Renew. Sustain. Energy Rev.* **2017**, *69*, 1170–1181. [[CrossRef](#)]
2. Li, Y.; Janik, P.; Schwarz, H.; Pfeiffer, K. Proposal of a regional grid cluster model for analysis of electrical power network performance. *Arch. Electr. Eng.* **2022**, *71*, 601–613. [[CrossRef](#)]
3. Kumari, P.; Toshniwal, D. Deep learning models for solar irradiance forecasting: A comprehensive review. *J. Clean. Prod.* **2021**, *318*, 128566. [[CrossRef](#)]
4. Soman, S.S.; Zareipour, H.; Member, S.; Malik, O.; Fellow, L. A Review of Wind Power and Wind Speed Forecasting Methods With Different Time Horizons. In Proceedings of the North-American Power Symposium (NAPS) 2010, Arlington, TX, USA, 26–28 September 2010; pp. 1–7. [[CrossRef](#)]
5. Das, U.K.; Tey, K.S.; Seyedmahmoudian, M.; Mekhilef, S.; Idris, M.Y.I.; Van Deventer, W.; Horan, B.; Stojcevski, A. Forecasting of photovoltaic power generation and model optimization: A review. *Renew. Sustain. Energy Rev.* **2018**, *81*, 912–928. [[CrossRef](#)]
6. Agga, A.; Abbou, A.; Labbadi, M.; El Houm, Y.; Ou Ali, I.H. CNN-LSTM: An efficient hybrid deep learning architecture for predicting short-term photovoltaic power production. *Electr. Power Syst. Res.* **2022**, *208*, 107908. [[CrossRef](#)]
7. Aksan, F.; Li, Y.; Suresh, V.; Janik, P. CNN-LSTM vs. LSTM-CNN to Predict Power Flow Direction: A Case Study of the High-Voltage Subnet of Northeast Germany. *Sensors* **2023**, *23*, 901. [[CrossRef](#)] [[PubMed](#)]
8. Rajagukguk, R.A.; Ramadhan, R.A.A.; Lee, H.J. A review on deep learning models for forecasting time series data of solar irradiance and photovoltaic power. *Energies* **2020**, *13*, 6623. [[CrossRef](#)]
9. Li, D.; Sun, G.; Miao, S.; Gu, Y.; Zhang, Y.; He, S. A short-term electric load forecast method based on improved sequence-to-sequence GRU with adaptive temporal dependence. *Int. J. Electr. Power Energy Syst.* **2022**, *137*, 107627. [[CrossRef](#)]
10. Lin, J.; Ma, J.; Zhu, J.; Cui, Y. Short-term load forecasting based on LSTM networks considering attention mechanism. *Int. J. Electr. Power Energy Syst.* **2022**, *137*, 107818. [[CrossRef](#)]
11. Bashir, T.; Haoyong, C.; Tahir, M.F.; Liqiang, Z. Short term electricity load forecasting using hybrid prophet-LSTM model optimized by BPNN. *Energy Rep.* **2022**, *8*, 1678–1686. [[CrossRef](#)]
12. Yazici, I.; Beyca, O.F.; Delen, D. Deep-learning-based short-term electricity load forecasting: A real case application. *Eng. Appl. Artif. Intell.* **2022**, *109*, 104645. [[CrossRef](#)]
13. Suresh, V.; Aksan, F.; Janik, P.; Sikorski, T.; Sri Revathi, B. Probabilistic LSTM-Autoencoder based hour-ahead solar power forecasting model for intra-day electricity market participation: A Polish case study. *IEEE Access* **2022**, *10*, 110628–110638. [[CrossRef](#)]
14. Duan, J.; Zuo, H.; Bai, Y.; Chang, M.; Chen, X.; Wang, W.; Ma, L.; Chen, B. A multistep short-term solar radiation forecasting model using fully convolutional neural networks and chaotic aquila optimization combining WRF-Solar model results. *Energy* **2023**, *271*, 126980. [[CrossRef](#)]
15. Wang, K.; Qi, X.; Liu, H. A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network. *Appl. Energy* **2019**, *251*, 113315. [[CrossRef](#)]
16. Wang, D.; Cui, X.; Niu, D. Wind Power Forecasting Based on LSTM Improved by EMD-PCA-RF. *Sustainability* **2022**, *14*, 7307. [[CrossRef](#)]
17. Liu, H.; Mi, X.; Li, Y. Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM. *Energy Convers. Manag.* **2018**, *159*, 54–64. [[CrossRef](#)]
18. Niu, Z.; Yu, Z.; Tang, W.; Wu, Q.; Reformat, M. Wind power forecasting using attention-based gated recurrent unit network. *Energy* **2020**, *196*, 117081. [[CrossRef](#)]
19. Jost, D.; Braun, A.; Brauns, K.; Dobschinski, J. Forecasting Vertical Power Flows at Transmission Grid Nodes characterized by high Penetration of Renewable Generation and Consumption. In Proceedings of the Wind Integration Workshop (WIW), Dublin, Ireland, 16–18 October 2019. Available online: <https://publica.fraunhofer.de/entities/publication/> (accessed on 28 June 2022). [[CrossRef](#)]
20. Brauns, K.; Scholz, C.; Schultz, A.; Baier, A.; Jost, D. Vertical power flow forecast with LSTMs using regular training update strategies. *Energy AI* **2022**, *8*, 100143. [[CrossRef](#)]
21. Paretkar, P.S.; Mili, L.; Centeno, V.; Jin, K.; Miller, C. Short-term forecasting of power flows over major transmission interties: Using Box and Jenkins ARIMA methodology. In Proceedings of the IEEE PES General Meeting, Minneapolis, MN, USA, 25–29 July 2010; pp. 1–8. [[CrossRef](#)]
22. Shao, X.; Kim, C.S. Multi-Step Short-Term Power Consumption Forecasting Using Multi-Channel LSTM with Time Location Considering Customer Behavior. *IEEE Access* **2020**, *8*, 125263–125273. [[CrossRef](#)]
23. Liu, Y.; Hou, D.; Bao, J.; Qi, Y. Multi-step ahead time series forecasting for different data patterns based on LSTM recurrent neural network. In Proceedings of the 2017 14th Web Information Systems and Applications Conference (WISA), Liuzhou, China, 11–12 November 2017; pp. 305–310. [[CrossRef](#)]
24. Alsharekh, M.F.; Habib, S.; Dewi, D.A.; Albattah, W.; Islam, M.; Albahli, S. Improving the Efficiency of Multistep Short-Term Electricity Load Forecasting via R-CNN with ML-LSTM. *Sensors* **2022**, *22*, 6913. [[CrossRef](#)]
25. Singh, N.; Vyjayanthi, C.; Modi, C. Multi-step Short-term Electric Load Forecasting using 2D Convolutional Neural Networks. In Proceedings of the 2020 IEEE-HYDICON, Hyderabad, India, 11–12 September 2020; pp. 1–5. [[CrossRef](#)]

26. Cheng, Y.; Chang, H.; Tang, K.; Zou, J.; Zhuo, J.; Cai, Y. Multistep electricity load forecasting method based on the hybrid GRU neural network. In Proceedings of the 2022 International Applied Computational Electromagnetics Society Symposium (ACES-China), Xuzhou, China, 9–12 December 2022; pp. 1–3. [\[CrossRef\]](#)
27. Hüsken, M.; Stagge, P. Recurrent neural networks for time series classification. *Neurocomputing* **2003**, *50*, 223–235. [\[CrossRef\]](#)
28. Smagulova, K.; James, A.P. Overview of long short-term memory neural networks. *Model. Optimal Sci. Technol.* **2020**, *14*, 139–153. [\[CrossRef\]](#)
29. Hochreiter, S. Long Short-term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#)
30. Janocha, K.; Czarnecki, W.M. On loss functions for deep neural networks in classification. *Schedae Inform.* **2016**, *25*, 49–59. [\[CrossRef\]](#)
31. Baranwal, M.; Salapaka, S.M. Clustering of Power Networks: An Information-Theoretic Perspective. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017.
32. Abou Hamad, I.; Arne Rikvold, P.; Poroseva, S.V. Floridian high-voltage power-grid network partitioning and cluster optimization using simulated annealing. *Phys. Procedia* **2011**, *15*, 2–6. [\[CrossRef\]](#)
33. Elhassan, A.; Abu-soud, S.M.; Alghanim, F.; Salameh, W. ILA4: Overcoming missing values in machine learning datasets—An inductive learning approach. *J. King Saud Univ.—Comput. Inf. Sci.* **2021**, *34*, 4284–4295. [\[CrossRef\]](#)
34. Aksan, F.; Janik, P.; Suresh, V.; Leonowicz, Z. Review of the application of deep learning for fault detection in wind turbine. In Proceedings of the 2022 IEEE International Conference on Environment and Electrical Engineering and 2022 IEEE Industrial and Commercial Power Systems Europe (IEEEIC/IEEICEP Europe), Prague, Czech Republic, 28 June–1 July 2022. [\[CrossRef\]](#)
35. Suresh, V.; Janik, P.; Rezmer, J.; Leonowicz, Z. Forecasting solar PV output using convolutional neural networks with a sliding window algorithm. *Energies* **2020**, *13*, 723. [\[CrossRef\]](#)
36. Wang, K.; Qi, X.; Liu, H. Photovoltaic power forecasting based LSTM-Convolutional Network. *Energy* **2019**, *189*, 116225. [\[CrossRef\]](#)
37. Lu, X.; Lin, P.; Cheng, S.; Lin, Y.; Chen, Z.; Wu, L.; Zheng, Q. Fault diagnosis for photovoltaic array based on convolutional neural network and electrical time series graph. *Energy Convers. Manag.* **2019**, *196*, 950–965. [\[CrossRef\]](#)
38. Hwang, H.P.C.; Ku, C.C.Y.; Chan, J.C.C. Detection of malfunctioning photovoltaic modules based on machine learning algorithms. *IEEE Access* **2021**, *9*, 37210–37219. [\[CrossRef\]](#)
39. Wang, F.; Xuan, Z.; Zhen, Z.; Li, K.; Wang, T.; Shi, M. A day-ahead PV power forecasting method based on LSTM-RNN model and time correlation modification under partial daily pattern prediction framework. *Energy Convers. Manag.* **2020**, *212*, 112766. [\[CrossRef\]](#)
40. Jahangir, H.; Aliakbar, M.; Alhameli, F.; Mazouz, A.; Ahmadian, A.; Elkamel, A. Short-term wind speed forecasting framework based on stacked denoising auto-encoders with rough ANN. *Sustain. Energy Technol. Assess.* **2020**, *38*, 100601. [\[CrossRef\]](#)
41. Hong, Y.; Lian, C.; Rioflorido, P.P. A hybrid deep learning-based neural network for 24-h ahead wind power forecasting. *Appl. Energy* **2019**, *250*, 530–539. [\[CrossRef\]](#)
42. Fekri, M.N.; Patel, H.; Grolinger, K.; Sharma, V. Deep learning for load forecasting with smart meter data: Online Adaptive Recurrent Neural Network. *Appl. Energy* **2021**, *282*, 116177. [\[CrossRef\]](#)
43. Le, X.; Ho, H.V.; Lee, G.; Jung, S. Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. *Water* **2019**, *11*, 1387. [\[CrossRef\]](#)
44. Wu, Y.; Wu, Q.; Zhu, J. Data-driven wind speed forecasting using deep feature extraction and LSTM. *IET Renew. Power Gener.* **2019**, *13*, 2062–2069. [\[CrossRef\]](#)
45. Xiuyun, G.; Ying, W.; Yang, G.; Chengzhi, S.; Wen, X.; Yimiao, Y. Short-term Load Forecasting Model of GRU Network Based on Deep Learning Framework. In Proceedings of the 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, China; 2018; pp. 1–4. [\[CrossRef\]](#)
46. Inteha, A.; Nahid-Al-Masood. A GRU-GA Hybrid Model Based Technique for Short Term Electrical Load Forecasting. In Proceedings of the 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST) 2021, Dhaka, Bangladesh, 5–7 January 2021; pp. 515–519. [\[CrossRef\]](#)
47. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. The Performance of LSTM and BiLSTM in Forecasting Time Series. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 3285–3292. [\[CrossRef\]](#)
48. TensorFlow. Available online: <https://www.tensorflow.org/> (accessed on 16 April 2023).
49. Keras: Deep Learning for Humans. Available online: <https://keras.io/> (accessed on 16 April 2023).
50. Wang, Y.; Zou, R.; Liu, F.; Zhang, L.; Liu, Q. A review of wind speed and wind power forecasting with deep neural networks. *Appl. Energy* **2021**, *304*, 117766. [\[CrossRef\]](#)
51. Badran, I.; El-Zayyat, H.; Halasa, G. Short-term and medium-term load forecasting for Jordan’s power system. *Am. J. Appl. Sci.* **2008**, *5*, 763–768. [\[CrossRef\]](#)
52. Chen, Y.; Wang, Y.; Dong, Z.; Su, J.; Han, Z.; Zhou, D.; Zhao, Y.; Bao, Y. 2-D regional short-term wind speed forecast based on CNN-LSTM deep learning model. *Energy Convers. Manag.* **2021**, *244*, 114451. [\[CrossRef\]](#)
53. Scikit-Learn: Machine Learning in Python—Scikit-Learn 1.0.2 Documentation. Available online: <https://scikit-learn.org/stable/> (accessed on 13 March 2022).

54. Koch, P.; Golovidov, O.; Gardner, S.; Wujek, B.; Griffin, J.; Xu, Y. Autotune: A Derivative-free Optimization Framework for Hyperparameter Tuning. In Proceedings of the KDD'18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018. [CrossRef]
55. Li, L.; Jamieson, K.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *J. Mach. Learn. Res.* **2018**, *18*, 1–52. Available online: <http://jmlr.org/papers/v18/16-558.html> (accessed on 8 May 2023).
56. Anzanello, M.J.; Fogliatto, F.S. Learning curve models and applications: Literature review and research directions. *Int. J. Ind. Ergon.* **2011**, *41*, 573–583. [CrossRef]
57. Tong, X.; Wang, J.; Zhang, C.; Wu, T.; Wang, H.; Wang, Y. LS-LSTM-AE: Power load forecasting via Long-Short series features and LSTM-Autoencoder. *Energy Rep.* **2022**, *8*, 596–603. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.