energies

Article

Justifying Short-Term Load Forecasts Obtained with the Use of Neural Models

Tadeusz A. Grzeszczyk [1,*] and Michal K. Grzeszczyk [2,3]

[1] Faculty of Management, Warsaw University of Technology, ul. Narbutta 85, 02-524 Warsaw, Poland
[2] Faculty of Electronics and Information Technology, Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland; michal.k.grzeszczyk@gmail.com
[3] Sano—Centre for Computational Personalised Medicine—International Research Foundation, ul. Nawojki 11, 30-072 Cracow, Poland
* Correspondence: tadeusz.grzeszczyk@pw.edu.pl

Abstract: There is a lot of research on the neural models used for short-term load forecasting (STLF), which is crucial for improving the sustainable operation of energy systems with increasing technical, economic, and environmental requirements. Neural networks are computationally powerful; however, the lack of clear, readable and trustworthy justification of STLF obtained using such models is a serious problem that needs to be tackled. The article proposes an approach based on the local interpretable model-agnostic explanations (LIME) method that supports reliable premises justifying and explaining the forecasts. The use of the proposed approach makes it possible to improve the reliability of heuristic and experimental neural modeling processes, the results of which are difficult to interpret. Explaining the forecasting may facilitate the justification of the selection and the improvement of neural models for STLF, while contributing to a better understanding of the obtained results and broadening the knowledge and experience supporting the enhancement of energy systems security based on reliable forecasts and simplifying dispatch decisions.

Keywords: time-series forecasting; short-term load forecasting; energy forecasting model; neural networks; explainability; local interpretable model-agnostic explanations

Citation: Grzeszczyk, T.A.; Grzeszczyk, M.K. Justifying Short-Term Load Forecasts Obtained with the Use of Neural Models. *Energies* **2022**, *15*, 1852. https://doi.org/10.3390/en15051852

Academic Editor: Ramiro Barbosa

Received: 10 February 2022
Accepted: 28 February 2022
Published: 2 March 2022

1. Introduction

Power (electric) load forecasting is of increasing importance due to the role of electricity and the daily functioning of the information society in running a business; there is a need to synchronize three processes: power generation, transmission and utilization, difficulties with storing large amounts of electric energy, and inevitable changes in power systems towards highly complex and intelligent solutions [1]. The results of electric load forecasts make it possible to satisfy the electric utilities-related needs of business entities, including those related to planning and operations of power systems, energy trading, rate design, and revenue projection; they are also useful for industrial and big commercial companies, regulatory commissions, trading firms, banks and insurance companies [2].

An essential concept in time series (TS) forecasting is the forecasting horizon, i.e., the farthest moment in the future for which forecasts are made. In general, from the point of view of the forecasting horizon and the dissimilarities of the problems to be solved related to it, load forecasting in power systems can be classified as follows: from a few minutes to hour-ahead scheduling—very short-term load forecasting (VSTLF); from hourly, daily and weekly to yearly TS—short-term load forecasting (STLF); up to 3 years ahead—medium-term load forecasting (MTLF) and up to 10 years ahead—long-term load forecasting (LTLF) [3].

The use of forecasting results with various time horizons solves different problems. Solutions connected with VSTLF are applicable, e.g., for the precise prediction of loads in energy management systems and the selection of demand response strategies for intelligent buildings, which can provide peak load reduction [4]. Models of STLF (typically from 24 to 72 h ahead) are often used to solve short-term unit commitment scheduling problems [5,6]. The improvement of solutions related to short-term forecasting is also conducive to the development of currently necessary research on how to optimize the planning of complex energy storage systems for electric and gas vehicles [7]. The monthly and yearly load forecasting results are helpful, e.g., in renewable-energy integration processes, in medium-term planning power plants or grids, and in generator maintenance scheduling [8]. On the other hand, LTLF is used primarily in long-term power system operation and planning, which can be based on macro-economic indicators (e.g., GDP and population), sectoral decomposition, technological penetration in various market segments and detailed temporal granularity [9].

Among the different types of forecasting, short-term load and electricity price prediction play a key role; research on all kinds of valuable methods is being developed [10,11]. Further considerations are focused on this type of forecasting; theoretical analysis is supplemented by the results of empirical research based on the load time series with a 3-day horizon (hourly granularity).

Many methods that have been used in STLF have been developed and can be classified as follows: traditional statistical methods (too complicated to properly analyze the electric load data, which are often non-linear and are highly variable over time; a significant fluctuation of features and high noise occur), artificial intelligence (AI) methods (accepting noisy, incomplete and non-linear data) and hybrid approaches (eliminating the disadvantages of different methods used separately) [12].

Statistical forecasting models include, e.g., linear regression, generalized linear regression, rule-based, classification and regression tree-based, mixed (multilevel) and ensembles of models. In the context of large datasets and big data (many dimensions and computational complexity) problems related to STLF, AI approaches, machine learning (ML) methods, and the most commonly used artificial neural network (ANN) forecasting are an essential addition to traditional statistical methods. In different variants (e.g., using pattern representations of TS), they work well in the case of STLF, for which complicated and intricate relationships between predictors, outcome variables, and TS with multiple seasonal cycles are required [13].

Recently, many successful STLF models have been based on ML and ANN. Among the STLF methods, there are, e.g., the similar-pattern method (similar day, pattern sequence and sequence learning), the variable selection method (stepwise method, correlation, mutual information, filtering and optimization algorithm), hierarchical forecasting (bottom-up, top-down, ensemble and weighed combination) and weather station selection (average model and optimal-number-of-stations model) [14]. The possibilities of using various types of ANN for STLF were investigated, e.g., convolutional neural network (CNN) [15,16], recurrent neural network (RNN) [1], multi-layer perceptron (MLP) and deep learning methods [17]. Neural networks are also helpful when building hybrid neural models, e.g., integrating CNN and bidirectional long/short-term memory (CBiLSTM) [18]. The results of these studies confirm the high importance of neural models in STLF.

However, using neural models for STLF is associated with certain inconveniences and problems. A clear and reliable selection of an appropriate neural network is often a significant challenge. The type of neural networks used and the structural parameters of the models are, to a large extent, selected heuristically. In addition, these are black-box models that provide results that are difficult or impossible to interpret, on the one hand (most often there is no justification for the forecasts obtained for specific input data), and, on the other, regarding the selection of critical parameters and tasks related to control processes, that work for the safety and planning operation of power systems. Therefore, the demand from practitioners forces the need for a scientific solution to the problem of the lack of a clear, readable and trustworthy justification of the STLF obtained with the use of neural models. This article aims to present an approach based on explaining the forecasting, which can constitute the basis for justifying the selection and improvement of neural models for STLF, building confidence in the results obtained, enhancing the security of energy systems based on forecasts and improving decision-making processes in load planning processes. The local interpretable model-agnostic explanations (LIME) method was used to determine reliable premises that justify and explain the forecasts.

It should be emphasized that the main contributions of the research are related to study findings regarding the possibility of interpretability and justification of neural models for STLF. The use of deep neural networks in forecasting and obtaining the lowest error values is not the objective of the studies. The obtained results are in line with the current research trend of key and growing importance, related to the search for justification and building trust in the many types of existing AI models. The study proposes an approach based on the LIME method that supports reliable premises that justify and explain the forecasts. The use of the proposed approach makes it possible to improve the reliability of heuristic and experimental neural modeling processes, the results of which are difficult to interpret in accordance with existing needs. Explaining the forecasting may facilitate the justification of the selection and the improvement of neural models for STLF, contribute to a better understanding of the obtained results, simplify dispatch decisions and broaden the knowledge and experience supporting the enhancement of energy systems security based on reliable forecasts.

Experimental implementations and verifications of the developed forecasting models were realized; the proposed approach supporting the formulation of reliable justification and explanation of the forecast was practically verified. To improve comparative results, two computational experiments were performed. The first one used the STLF model based on the RNN and long short-term memory (LSTM) architecture. Then, in the second experiment, the first experiment results were compared to two benchmarks using linear regression and the LSTM network. In this experiment, the five-fold cross validation was conducted for linear regression and the LSTM network based on modified data. Then, the five trained networks were used to generate LIME explanations. The use of cross-validation and division of the dataset into several folds was a relatively simple way to enrich empirical research and to improve the comparative results.

The next section discusses the key possibilities of justifying STLF obtained with the application of ANN; Section 3 describes the methodological issues relating to the proposed approach and empirical research. Section 4 presents the research results and indicates the importance of the findings for filling the existing gap in the STLF field. The last section briefly highlights the significant results achieved and outlines possible directions for further research.

2. Possibility of Justifying and Explaining Neural Forecasts

Justifying and explaining neural forecasts consists in getting to know the variables that impact the determined explained variable and examining their significance in the forecasting processes. The forecast justification and explanation approaches differ for the various models used (traditional statistics vs. ANN). Traditional statistics require a thorough understanding of the predicted phenomena and the use of exploratory data analysis (EDA) and hypothesis testing, while ANN allows for building flexible models that are adapted to work with large volumes of data, that have good predictive performance, and that do not require users to have such a significant knowledge domain nor to run a complex EDA [19].

Models of ANN are one of the best-known AI solutions that are used in predicting energy and load TS. The interest of the scientific community in these issues is expressed in the growing number of publications (about half of all publications in the area of energy forecasting are load forecasting papers); the massive development of computing technologies is conducive to the application of various advanced ANN and ML methods [20]. Generally, only the most commonly used regression models are more popular than the TS analysis with explanatory variables using ANN [21].

The idea of building mathematical neural models arose as a result of drawing inspiration from the observed natural neurons and the connections between them (synapses) that occur in the human brain. Biologists conduct comparative analyses of neural models and brains in terms of biological characteristics and study human and animal performance while solving various tasks; neuroscientists are interested in cognitive functions performed by brains; the central area of interest of energy engineers is the possibility of using ANN as a powerful forecasting tool [22]. Neural models used in STLF also more or less refer to information processing of the brain, its structure, and individual areas, e.g., the visual cortex of the brain [23]. Working with these models takes place in two stages: learning and using the models for forecasting. In the first stage, a learning process (usually supervised) takes place, as a result of which the weights assigned to individual synapses present in the selected connection topology (architecture) are modified. The intuitively and heuristically selected architecture of neuron connections and their weights in the learned network structure are the key elements of models that enable forecasting. While the first stage of preparing such models might be complex and time-consuming, the second stage appears to be relatively easy and fast in terms of its application.

Supervised learning (supervised ML) consists of using search algorithms for the most appropriate output signals (network response obtained at the output) corresponding to the input information. Models of ANN learn from datasets that contain learning examples, i.e., pairs of input and the corresponding output information.

For the successful performance of supervised learning, weak supervision sometimes has to be used if it is impossible to provide strong supervision information based on the usually costly research associated with collecting a large amount of learning examples and the data-labeling process [24]. In the learning processes, the ability of ANN is used to generalize the knowledge of the experiences obtained during the learning stage. Thanks to this, it is possible to obtain correct output information also in the case when the input of neural models is provided with information that the ANN did not deal with during the learning process (it did not exist in the training set).

Models of ANN are simple to apply and have powerful application capabilities in the field of load forecasting. They are easily used not only by experts with broad and deep competencies in AI and ML but also by business practitioners. Essential advantages of deep learning models based on ANNs are that they use mathematical tools extracted from empirical data and that they often perform better than physics-based models when it is necessary to conduct multifaceted and multidimensional analyses [25]. Moreover, such models do not require detailed analysis nor learning the description of the relationships between the input (explanatory, independent) variables (features) and the predicted response (explained, dependent) target variable.

The enormous potential of application values of neural models is accompanied by serious difficulties resulting from the problem of the lack of legible and credible justifications for the results obtained at the outputs of such black-box systems, inside which the forecaster cannot see; the knowledge stored in the structures of connections between neurons and the weights assigned to them is somewhat unreadable and incomprehensible. In general, black-box-based approaches are among the most popular data-driven models (used for energy prediction and forecasting), which, in addition to ANN, also include regression, multiple linear regression, Gaussian process regression (GPR), support vector machines (SVM), decision trees and several other optimization methods [26].

For black-box-based approaches and methods, it is challenging to build valuable dependencies and mathematical functions that allow for reflecting the meaning and influence of individual variables on the obtained values of the explained variable. The generation of mathematical descriptions resulting from in-depth forecast phenomena is rather typical of traditional statistical forecasting models. This is positively perceived by business practitioners, who usually do not want to study the forecasted phenomena thoroughly and do not feel the need to use complicated mathematical tools. Satisfaction with the easy application of neural models ends when there is a need to justify and explainneural forecasting models. Neural models are difficult to understand by practitioners due to their increasing complexity, the presence of many dimensions, the unavailability of the contents and their meaning, and the transparency of black-box tools [27]. One of the methods used by practitioners is the ex-post evaluation of models, which consists of a simple comparison of actual and forecasted values, and which, on this basis, determines forecast errors.

Justifying and explaining neural STLF based on ex-post evaluations of neural models is usually of crucial importance in practice. The quality of the obtained results and the size of errors primarily result from the hidden knowledge of the designers of these models, i.e., their experiences and intuition. It is challenging to build solely on this during STLF, as the consequences of incorrect predictions of load in specific time intervals can be severe. Load depends on many factors and is often random in nature, which may cause load forecast errors, inefficient daily system operation, and the following negative economic impacts: if the load is under-forecast, the energy demand may not be satisfied; in the case of an over-forecast occurrence, there may be unnecessary start-ups and excessive spinning reserve (SR) [28].

Appropriate STLF is not only conducive to the creation of a proper SR capacity but may also contribute to the minimization of production costs and to an increase in power system reliability [29]. Minimizing the load forecasting error and determining correct forecasts of this kind makes it easier to solve the unit commitment problem, taking into account the SR of dispatchable units that help to ensure the availability of adequate energy storage and correct operation scheduling under demand estimation uncertainties [30].

The possibility of forecast errors is increased due to the growing complexity of problems that are heuristically and intuitively solved with the help of dynamically developing large deep neural models, which are also successfully used in forecasting. Therefore, it is reasonable to look for solutions that enable the justification and explanation of neural STLF. Among the different directions developed for explaining neural models, some are related to already trained and fixed models; the others are related to self-explanatory neural models with built-in modules for generating forecast explanations [31]. Further considerations and computational experiments are focused on in the first type of model.

Justifying and explaining neural STLF for an already trained and fixed model can be achieved, for example, using Shapley additive explanations (SHAP) [32] or the LIME method that belongs to the model-agnostic methods (MAM) class and consists in using an interpretable model to explain predictions in a credible way using local approximation [33]. Both methods are competitive, e.g., in relation to the feature importance plots method (developed into partial dependence plots) [34] that provides information about the global model behavior but that does not provide a clear interpretation of the relationships between variables, i.e., an explanation of making particular classifications or forecasting; therefore, it is not suitable for justifying STLF obtained with the use of ANN. The results of empirical research related to comparative analyses of the effectiveness and efficiency of basic explanation algorithms and methods are available in the literature [35].

Some argue that high-stakes decisions assisted by neural models should be avoided due to the general difficulties in obtaining justifications and explanations for the results achieved with black-box ML models [36]. That is why it is essential to conduct research in this field and to provide solutions that allow one to justify and explain neural forecasting models. Applying the universal (useful for various types of black-box models) LIME method, one can obtain interpretive explanations that support the understanding and justification of the results with a high likelihood, according to the feature space defined by the user. Then, the local approximation of model behavior is determined, which applies to most items from datasets [37].

In general, the justification of predictions and interpretability of ML and ANN contributes to increasing the acceptance of the deepening integration of machines and soft computing algorithms with the business environment and everyday life. Explanations using MAM effectively support the interaction of people with machines thanks to the model flexibility (cooperation with any ML model), explanation flexibility (different forms of explanation) and representation flexibility (other feature representation compared to the model being explained) [38]. The main stages of the LIME method are as follows [39,40]:

(1) Selecting observation for explaining and justifying,
(2) Generating a new dataset with perturbed samples for a selected observation (randomly around it),
(3) Using the chosen black-box model to calculate the forecast for the permuted data,
(4) Calculating the weights of new samples according to their proximity to the selected observation—the weight values determine the relative importance of each permutated sample,
(5) Identifying features from permuted data enabling the best description of the neural model,
(6) Using the permuted data to train a simple interpretable model,
(7) Explaining the neural models' local behavior by using weights of features regarding the simple model.

The use of the LIME method enables the prediction of the behavior of neural models, building trust in them and the forecasts determined with their help. The results of justifications and explanations of the models also provide a basis for comparing different models and an opportunity for their improvement.

3. Methodology

Empirical research was carried out using an electricity load forecasting dataset that contains an hourly post-dispatch electricity load for Panama, ranging from 3 January 2015 to 27 June 2020 (48048 samples) [41]. Apart from the date, time and load, each sample was associated with weather data from three big cities in Panama (David, Santiago and Panama City). The weather data consisted of wind speed, humidity, air temperature (factors measured at 2 m above ground) and total precipitable liquid water. Finally, all samples were extended with binary features indicating whether the specific day was during the school period and whether holidays were occurring at this time. The last feature of the dataset was the holiday indicator (equal to 0 if there was no holiday). Since the period of the year and the day are important factors regarding STLF, the DateTime feature was divided into three features: month, date and hour, which resulted in the initial dataset containing 19 features.

As part of the research, two computational experiments were implemented to carry out the empirical verification of the proposed models, determine their forecasting metrics, and check the suitability of the approach that supports the justification and explanation of forecasts based on the LIME method. The first forecast experiment was performed as follows. As in the previous studies, the horizon of 72 h (break of 72 h) was assumed [42]. Contrary to previous studies, the forecast concerned specific hours and was estimated based on the values of the 96 h before the 72-h gap (Figure 1). Due to the forecasting horizon of 72 h for pre-dispatch load reports, we decided to use the neural model for predicting the load after the 72-h gap from the values of 19 features during 96 consecutive hours (therefore, each input sample contained 1824 features). The dataset was divided into training (80%) and testing (20%) by splitting the initial set after 80% of consecutive samples.

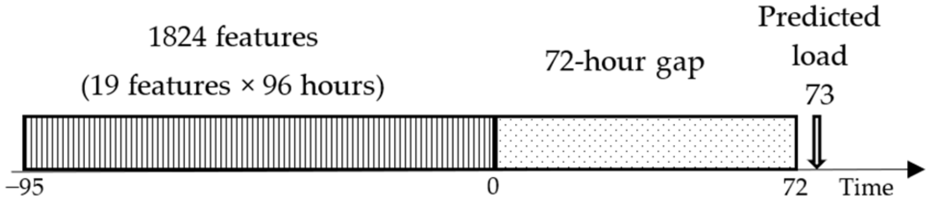

Figure 1. The scheme of making the forecast (experiment 1).

After dividing the dataset, the subsets were preprocessed to form 96 blocks of features (for each hour) and the corresponding load after a 72-h gap, which resulted in the creation of the final training set of shape (38270,96,19) and the final testing set of the form (9442,96,19).

This means that the data in training and testing datasets were collected in arrays with three dimensions corresponding to the numbers of samples, hours and variables. Before training, the input features were normalized to have a mean of 0 and a variance of 1. The testing set was transformed using the same values for normalization as in the training set.

Since the STLF task structured in this way is based on sequential data, the ANN model based on RNN was chosen. The developed model consisted of 2 LSTM layers (both containing two units and rectified linear unit activation). The last layer of the model was a fully connected layer composed of one neuron with linear activation to perform the regression of electricity load.

The model was trained with a batch size of 64 using the mean squared error (MSE) loss function. For the training, the method for stochastic optimization, i.e., the adaptive moment estimation (Adam) method [43], with a learning rate of 0.001 for the first 20 epochs, was used. Then, the model was fine-tuned for the next 60 epochs with Adam optimizer's learning rate of 0.0005. The hyperparameters of the model were chosen empirically after initial experiments. The model had 219 trainable weights; one training epoch took 675 s on average (with the mean training step of 1.13 s) on graphics card NVIDIA GeForce GTX 960M. After the model was trained, the LIME method was applied to explain the model predictions of the electricity load in the next 72 h. This approach allowed us to justify the model's prediction and to clarify what could impact the electricity demand load. For clarity, only the 40 most important features (detected by LIME) were analyzed.

Then, as part of the second computational experiment, the results from the first experiment were compared to two benchmarks using linear regression and the LSTM network. The dataset for training the linear regression was created from the 19 features during the same hour and on the same day as in the previous four weeks (4×19 features in each training sample). The second forecasting experiment used the above-mentioned dataset and the LSTM network with parameters and architecture similar to those described earlier in this article. The second network was trained with analogous parameters with one change (0.001 learning rate was applied for the first 40 epochs of training, and 0.0005 learning rate was used for the next 80 epochs of training). Before training the LSTM network, the number of features per hour was reduced to 10 (4×10 features in the training sample) using the recursive feature elimination (RFE) function and cross-validated metrics. RFE is one of the most crucial feature-engineering techniques, enabling the elimination of features that could negatively affect training processes and model functioning [44]. The 5-fold cross-validation was conducted for linear regression and the second LSTM training. Then, the five trained networks were used to generate LIME explanations. The second LSTM network had 147 trainable weights (the number is lower than in the case of the first network due to the lower number of analyzed days); one training epoch took 42 s on average (with the mean training step of 71 ms).

To perform the necessary calculations, program codes were prepared using the high-level object-oriented interpreted programming language Python [45]. The models were implemented with Tensorflow/Keras frameworks and libraries for deep learning models development [46]. Data processing was conducted using flexible Pandas [47] and Scikit-learn libraries [48]. The LIME algorithm was derived from the original implementation [49].

4. Results and Discussion

Figure 2 and Table A1 present the real and predicted electricity load during 1–4 June 2019. It can be noticed that the ANN managed to learn the main trends in electricity load; the only mistakes were made during high fluctuations between consecutive hours. The figure containing more days of prediction curves is shown in the Appendix C section.

The qualitative analysis of the compliance of the forecasted values with the actual load curve indicates the satisfactory suitability of the constructed model for forecasting. The most popular (well described in the literature) error measurements (key performance criteria, forecasting metrics) for the quantitative assessment of neural models (e.g., used for TS forecasting) include the following: mean absolute percentage error (MAPE), given as a percentage, and following three as absolute values—mean absolute error (MAE), mean square error (MSE) and root mean square error (RMSE) [50,51].

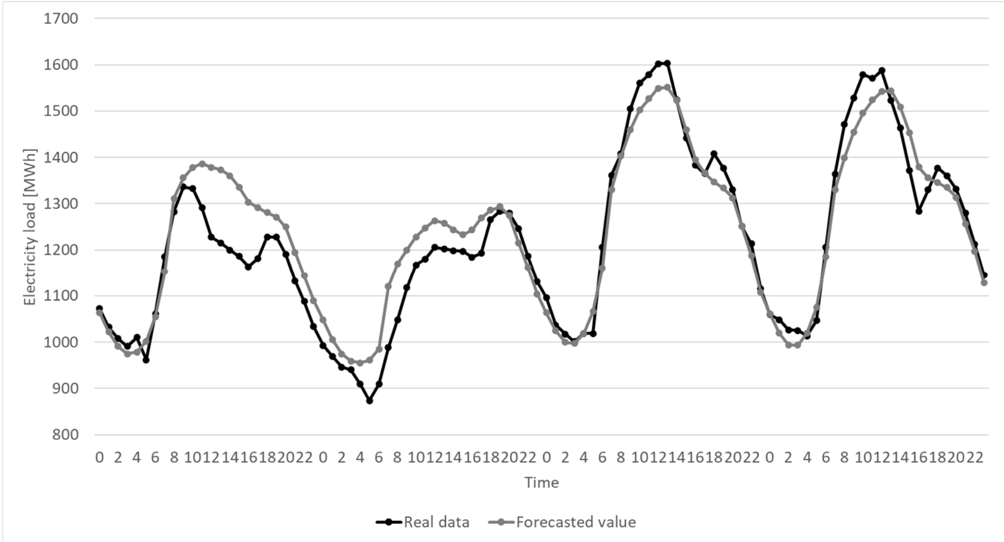

Figure 2. Actual vs. predicted electricity load in experiment 1.

Forecasting metrics determined for the test dataset (presented in Table 1) confirm that the developed STLF model based on the RNN and LSTM architecture is characterized by meaningful predictive ability and forms a reasonable basis for continuing research in the second experiment. The model based on LSTM layers achieved the MSE error rate of 8126.50 and RMSE error rate of 90.15. Those are satisfactory results considering that a significant part of the testing set coincided with the COVID-19 pandemic, during which the electricity load was radically different.

Table 1. Forecasting metrics (experiment 1).

Type of Error	Value
MAPE	5.68
MAE	68.54
MSE	8126.50
RMSE	90.15

Figure 3 and Table A2 present the explanations provided by LIME for the 40 most essential features in the model's prediction (the numbers after feature name and underscore indicate the hour in the 96-h period in the input sample for which the feature was taken and analyzed). The horizontal axis visible in the explanation for the prediction chart clearly shows the impact of individual features; this impact is measured in MWh. Such units allow a clear interpretation of the positive and negative measurable impact defined by the plus or minus signs.

The explanation was generated for the prediction of the electricity load in the first hour of 1 June 2019. The real electricity load for this hour was 1072.2 MWh; the predicted value was 1063.06 MWh. As expected, the features that had the most significant impact on the prediction were the hour values (indicating the time of the day for which the forecast was being made) and the national electricity demand. For instance, the national demand at midnight of 25 May 2019 (nat_demand_0), higher than 1029.32 MWh, impacted the model's prediction by around 22.3 MWh. After the two most essential feature types, the weather variables started to appear. For example, air temperature 2 m above ground in Panama City at 4 a.m. on 25 May 2019, lower than 26.94 degrees Celsius, increased the electricity load prediction by 5.90 MWh. The LIME algorithm offered insight into the previously created black-box model and provided knowledge about the features impacting the national electricity load.

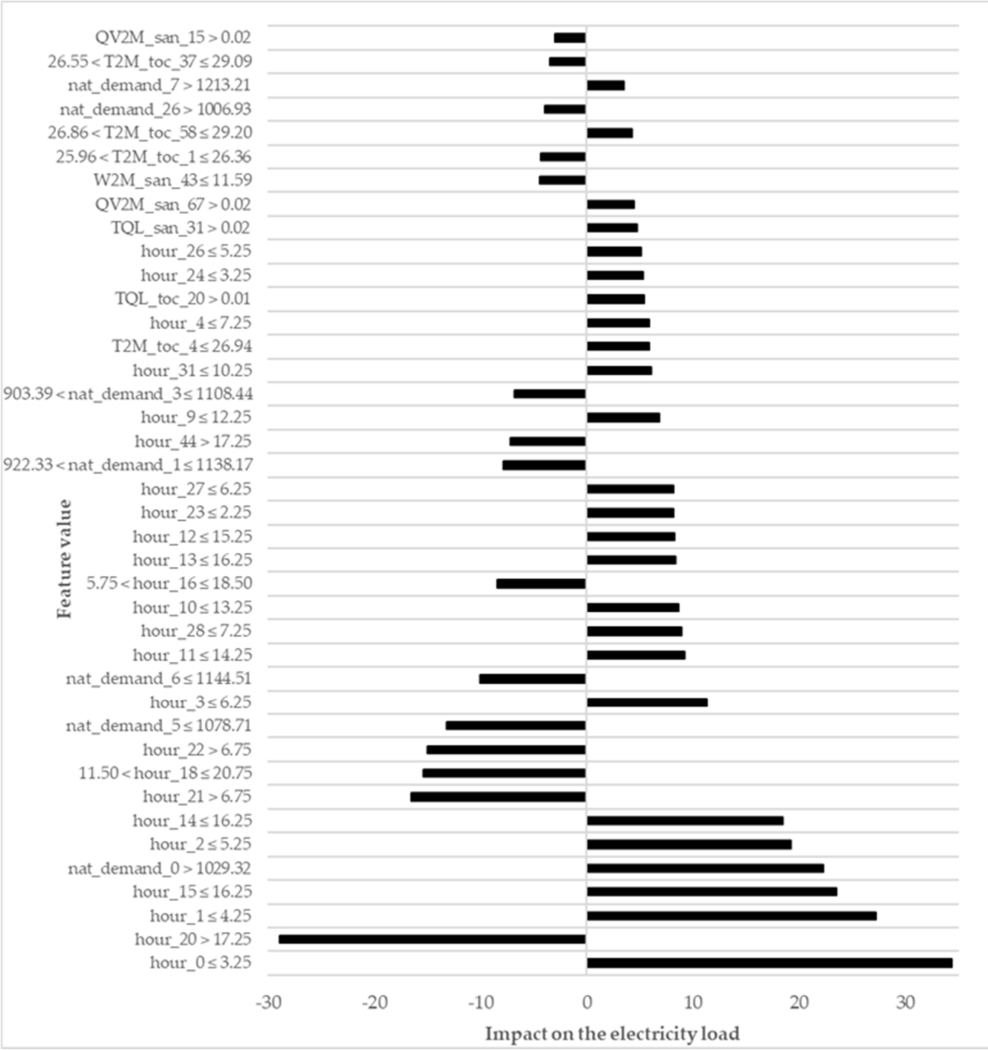

Figure 3. Explanations for the prediction of the electricity load in experiment 1.

The two computational experiments allowed us to examine the quality of the predictive models and to outline the possible results from applying the LIME method. The dataset presented earlier were used in the empirical study of the first experiment; the second experiment expanded the diversity of data spaces and feature dimensions by introducing 5-fold cross-validation in this set. Forecasting metrics for the second experiment are presented in Table 2. The obtained MAPE values were not very good, although sometimes even 6–7% values are considered accurate [52]. Still, they can be acknowledged to be satisfactory at this research stage, focused mainly on examining the possibility of using the LIME method.

The results of the second experiment were related to slightly better MAPE values compared to the first one. It may be noted that the average MAPE for linear regression was a bit lower than for LSTM. This is somewhat surprising, as it could be expected that this deep neural network would have much better results than the simpler model. The applied LSTM model was, however, relatively simple, and the full potential of this type of neural network was not used, which would probably be revealed after using higher complexity deep learning networks (with more layers and units). Moreover, obtaining such results could be caused by the specificity of the dataset, which, to some extent, included data for the testing set related to the COVID-19 pandemic period (Fold 5).

Table 2. Forecasting metrics for the second experiment.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
Errors for linear regression						
MAPE	4.69%	4.30%	4.38%	4.24%	5.25%	4.57%
MAE	52.47	48.47	50.96	48.59	63.2	52.73
MSE	5590.97	4916.31	5260.55	4924.68	7491.08	5636.72
RMSE	74.77	70.12	72.53	70.18	86.55	74.83
Errors for LSTM based on cross-validation splitted data						
MAPE	4.99%	4.09%	4.24%	4.16%	5.48%	4.59%
MAE	55.79	45.8	49.23	47.87	66.55	53.05
MSE	6157.98	4668.14	5115.48	4787.59	8231.21	5792.08
RMSE	78.47	68.32	71.52	69.19	90.73	75.65

Future research may involve using more empirical datasets. Applying the k-fold cross-validation method in these experiments made it possible to better examine the developed models' quality using the single dataset. In the first experiment, a division into a training set and a test set was made. Using the previously mentioned method in the second experiment allowed us to enter five equinumerous subsets (folds) of the available dataset. Consequently, individual subsets were applied for testing; the rest were applied for training. Thanks to this, it was possible to eliminate misinterpretations related to the strict selection of the division into training and test sets.

As a result of applying the RFE algorithm implemented in the Scikit-learn Python environment, the following ten best features were obtained: 'nat_demand', 'T2M_toc', 'W2M_toc', 'T2M_san', 'W2M_san', 'T2M_dav', 'W2M_dav', 'month', 'day' and 'hour'. These features were used for each of the four hours that fell on the four days of the consecutive weeks preceding the forecasts. For this reason, there are 40 items on the vertical axis of the LIME explanation chart.

Figure 4 presents the explanations provided by LIME for LSTM and Fold 1. The results for the remaining four folds are presented in Figure A1. These figures show that justifications and explanations are slightly different due to the data set's introduced divisions. There are, however, clear analogies and repeated dependencies between these LIME results, which confirms the practical usefulness of the proposed approach.

The obtained results indicate the possibility of practical local interpretation (for specific observation), concerning even complex and complicated neural forecasting models; they also offer a deeper understanding and justification of their load predictions, while also generating explanations, thanks to the identification of features that are particularly important for the values obtained on the output of models.

Regardless of these positive features of approaches based on the LIME method, many of their disadvantages should be identified. One of these drawbacks is related to possible problems with stability issues and the generation of different justifications for repeated calculations under similar conditions, which may, for example, result from randomly generated data around selected observation.

For the LIME stability assessment, additional indicators may be helpful; they make it possible to increase confidence in the achieved results of calculations and to avoid cases when different explanations for the same forecasts are obtained [53]. One of the possible ways is to reduce instability in the obtained explanations, e.g., by replacing random perturbation of data with agglomerative hierarchical clustering (AHC) [54]. The robust model interpretability can sometimes be difficult due to the application of local approximation based on linear models, which may be inadequate for many analyzed problems.

One way to overcome this limitation is to use a kernel-based LIME with feature dependency sampling (KLFDS), which can contribute to reducing errors resulting from the use of linear approximation, not taking into account complicated correlations between features and usually-non-linear local decision boundaries [55].

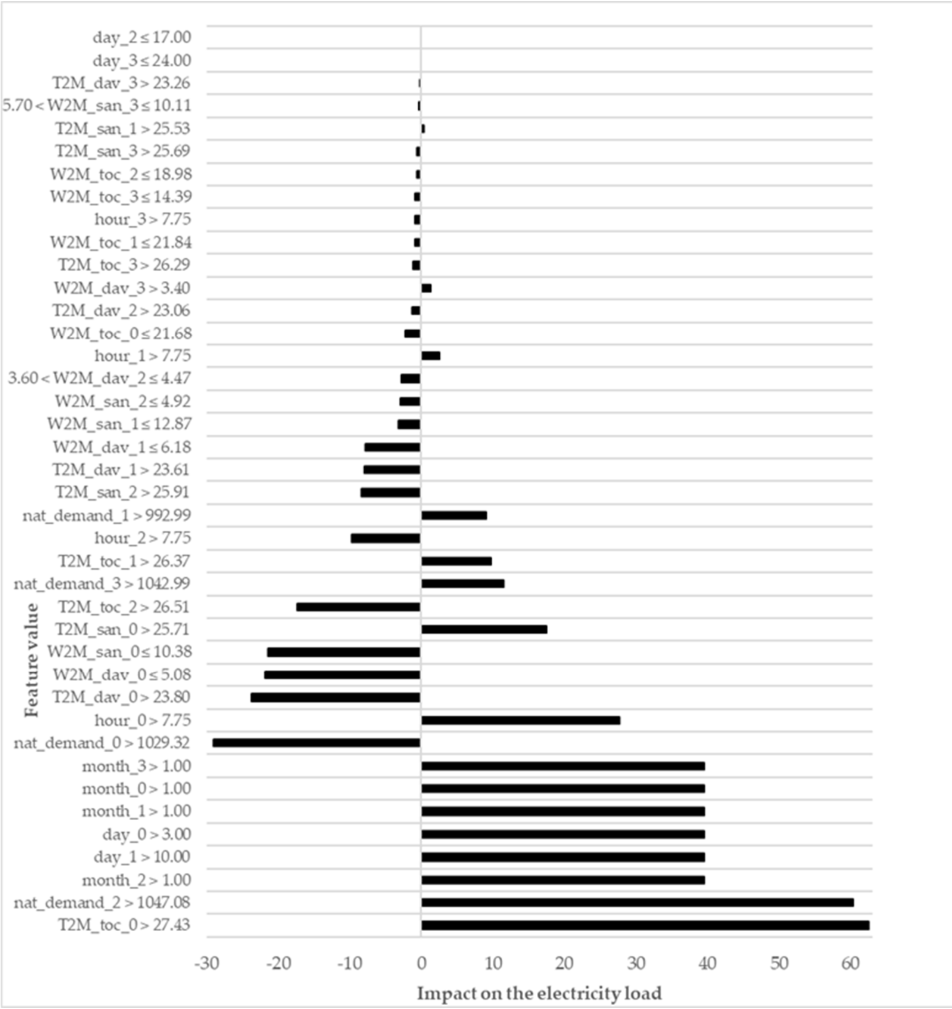

Figure 4. Explanations for LSTM and Fold 1 (experiment 2).

5. Conclusions

Results of short-term load forecasting affect the selection of critical parameters and tasks related to the control processes, production cost, work safety and planning for the operation of reliable power systems. The approach proposed in the article, supporting the determination of credible premises that justify and explain the forecast, opens a vast research area for improving the applied neural short-term load forecasting models and the reliability of the processes of building machine learning and neural network models used for forecasting based on black-box models, the results of which are often not trustworthy and are challenging to interpret. On the one hand, neural networks are characterized by substantial computational capabilities that are useful in forecasting energy and load time series; a significant number of publications in this field have been created. On the other hand, the scientific community and business practitioners notice the critical problem of the lack of a clear, readable and credible short-term load forecasting justification obtained using such models.

The aim of the article, to present an approach based on the explanation of forecasting, was achieved, which can constitute the basis for justifying the selection and improvement of neural models for short-term load forecasting. The short-term load forecasting models using the recurrent neural network architecture and long short-term memory were built in connection with introducing this goal. Then, its experimental implementation and empirical verification were carried out, confirming its meaningful predictive ability. Finally, this approach supported establishing reliable premises, justifying and explaining the forecast based on the local interpretable model-agnostic explanations method.

Taking the abovementioned into consideration, on the one hand, the obtained research results form the basis for the use of a satisfactory accurate neural forecasting model. On the other hand, the analysis using the local interpretable model-agnostic explanations method and justification of the prediction results may contribute to their better understanding, broadening the knowledge and experiences that contribute to increasing the possibilities of improving the quality of subsequent forecasts. The presented research outcomes show that the explanation of forecasting can be the basis for justifying the selection and improvement of neural models for short-term load forecasting, building confidence in the results obtained, increasing the security of energy systems based on forecasts and improving decision-making in load planning processes.

The availability of data used in calculations increases the presented results' credibility and facilitates comparative analyses by other researchers and business practitioners. The achieved results constitute a reasonable basis for further development of research in the field of load forecasting. For example, future research could use different neural models and various methods for explaining black-box models. It is possible to attempt research on reducing explanation algorithms and on defects of methods resulting from stability issues, as well as on randomly generated data around selected predictions and on obtaining different justifications for repeated calculations under similar conditions. Another area of possible research may focus on ways to overcome the limitations that result from the determination in the local interpretable model-agnostic explanations method of local approximations using linear models, which may be inadequate for many analyzed problems. It is also worth developing research on how to justify and explain load forecasting types other than short-term load forecasting, i.e., very short-term, medium-term and long-term.

Author Contributions: Conceptualization, T.A.G. and M.K.G.; methodology, T.A.G.; software, M.K.G.; validation, M.K.G.; formal analysis, M.K.G.; investigation, T.A.G.; resources, M.K.G.; data curation, M.K.G.; writing—original draft preparation, T.A.G.; writing—review and editing, T.A.G. and M.K.G.; visualization, T.A.G. and M.K.G.; supervision, T.A.G.; project administration, T.A.G.; funding acquisition, T.A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research and the APC were funded by the Warsaw University of Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The authors used data from [41].

Acknowledgments: The authors would like to thank the anonymous reviewers for their much-valued comments and suggestions.

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

Appendix A

Table A1 displays the results of the forecasts for two days within experiment 1.

Table A1. Selected forecasting results.

Time	Real Data	Forecasted Value
0	1072.1968	1063.062134
1	1032.3157	1021.810242
2	1007.6383	990.6638794
3	990.8934	974.3623657
4	1010.025	978.1856079
5	961.7391	1001.430359
6	1061.3078	1054.686035
7	1185.7693	1152.481079
8	1282.3128	1310.768188
9	1336.0031	1355.456787
10	1332.4882	1378.082275

Table A1. *Cont.*

Time	Real Data	Forecasted Value
11	1290.6503	1386.229858
12	1227.4551	1378.469116
13	1215.3455	1372.918579
14	1199.5798	1360.308105
15	1185.8578	1335.524292
16	1163.5908	1303.252197
17	1181.7484	1291.111572
18	1228.4201	1280.887939
19	1227.8296	1270.615356
20	1190.3462	1249.526245
21	1132.5383	1194.442871
22	1087.9636	1143.660767
23	1033.2848	1089.948975
0	992.4635	1047.69165
1	969.2115	1005.492493
2	946.2374	974.3960571
3	939.9386	959.0457153
4	909.2571	955.1030273
5	873.436	961.0645142
6	909.0867	985.1030273
7	988.2789	1120.075073
8	1047.6219	1169.357788
9	1117.3657	1199.871338
10	1167.4369	1228.380249
11	1180.0029	1247.645508
12	1205.2567	1262.932129
13	1202.2126	1258.16394
14	1198.2685	1243.799683
15	1197.2401	1233.031982
16	1184.4191	1242.855103
17	1192.5696	1268.72876
18	1266.025	1285.984131
19	1283.757	1293.797607
20	1279.0493	1274.454834
21	1245.9557	1214.341064
22	1186.8137	1162.392456
23	1130.7019	1103.609497

Appendix B

Table A2 shows the explanations for essential features within experiment 1.

Table A2. Results concerning explanations.

No	Feature	Impact
0	hour_0 ≤ 3.25	34.34750015
1	hour_20 > 17.25	−28.84380605
2	hour_1 ≤ 4.25	27.26926009
3	hour_15 ≤ 16.25	23.48685886
4	nat_demand_0 > 1029.32	22.30685105
5	hour_2 ≤ 5.25	19.22766052
6	hour_14 <= 16.25	18.46285147
7	hour_21 > 6.75	−16.52832108
8	11.50 < hour_18 ≤ 20.75	−15.38437866
9	hour_22 > 6.75	−14.97356073
10	nat_demand_5 ≤ 1078.71	−13.16680104
11	hour_3 ≤ 6.25	11.32555623
12	nat_demand_6 ≤ 1144.51	−10.06540743
13	hour_11 ≤ 14.25	9.260677418
14	hour_28 ≤ 7.25	8.95563697
15	hour_10 ≤ 13.25	8.685037066
16	5.75 < hour_16 ≤ 18.50	−8.446567774
17	hour_13 ≤ 16.25	8.376842584
18	hour_12 ≤ 15.25	8.233946128
19	hour_23 ≤ 2.25	8.209306696
20	hour_27 ≤ 6.25	8.184816332

Table A2. *Cont.*

No	Feature	Impact
21	922.33 < nat_demand_1 ≤ 1138.17	−7.865249086
22	hour_44 > 17.25	−7.166772948
23	hour_9 ≤ 12.25	6.855280623
24	903.39 < nat_demand_3 ≤ 1108.44	−6.797931718
25	hour_31 ≤ 10.25	6.05921166
26	T2M_toc_4 ≤ 26.94	5.907768902
27	hour_4 ≤ 7.25	5.862321239
28	TQL_toc_20 > 0.01	5.393426144
29	hour_24 ≤ 3.25	5.310395233
30	hour_26 ≤ 5.25	5.102217001
31	TQL_san_31 > 0.02	4.736387869
32	QV2M_san_67 > 0.02	4.474977329
33	W2M_san_43 ≤ 11.59	−4.388825137
34	25.96 < T2M_toc_1 ≤ 26.36	−4.321585585
35	26.86 < T2M_toc_58 ≤ 29.20	4.257097006
36	nat_demand_26 > 1006.93	−3.967277063
37	nat_demand_7 > 1213.21	3.526961657
38	26.55 < T2M_toc_37 ≤ 29.09	−3.399095666
39	QV2M_san_15 > 0.02	−2.975194042

Appendix C

Figure A1 shows actual vs. predicted electricity load in experiment 1 (the presented period is 9 July 2019–20 July 2019).

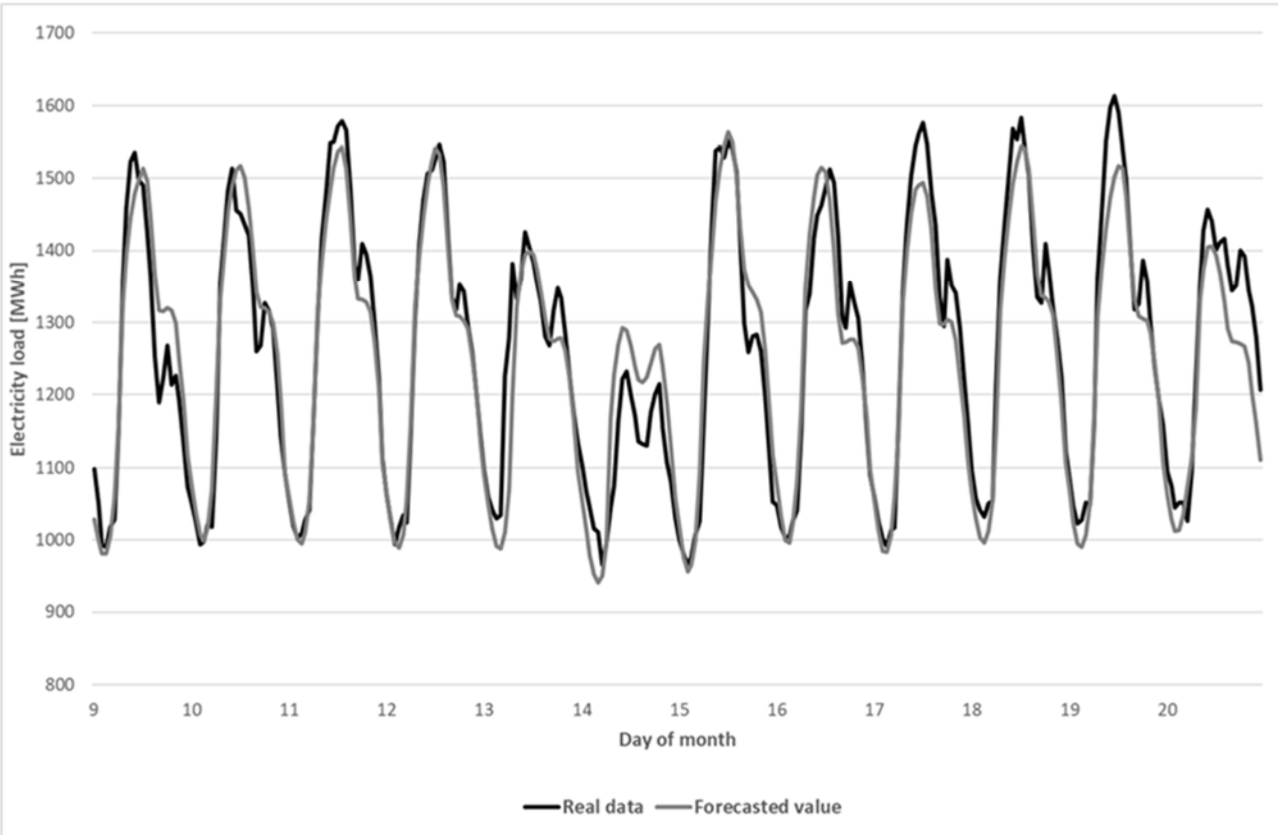

Figure A1. Actual vs predicted electricity load.

Appendix D

Figure A2 presents the explanations for LSTM and Fold 2–5 within experiment 2.

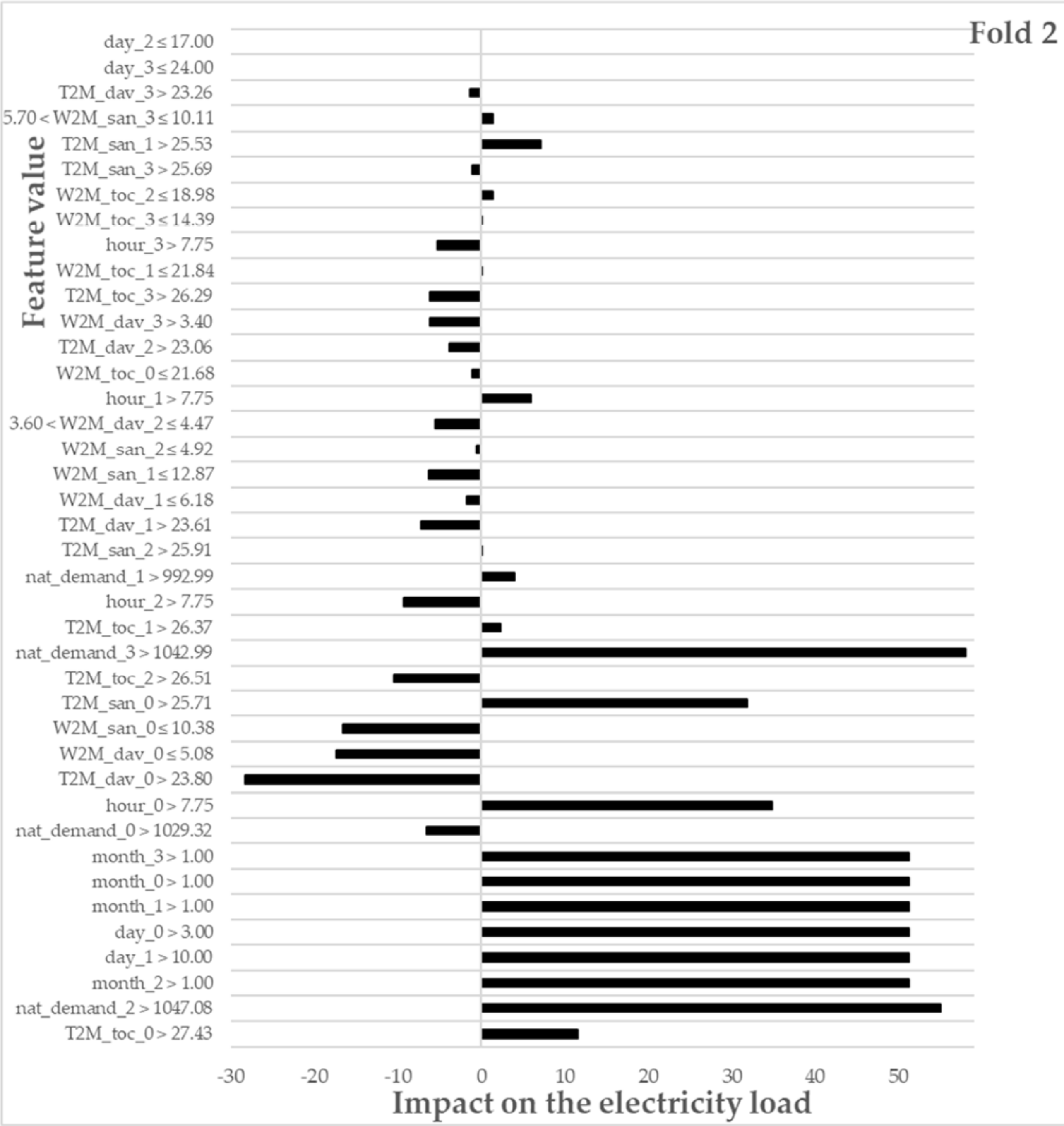

Figure A2. *Cont*.

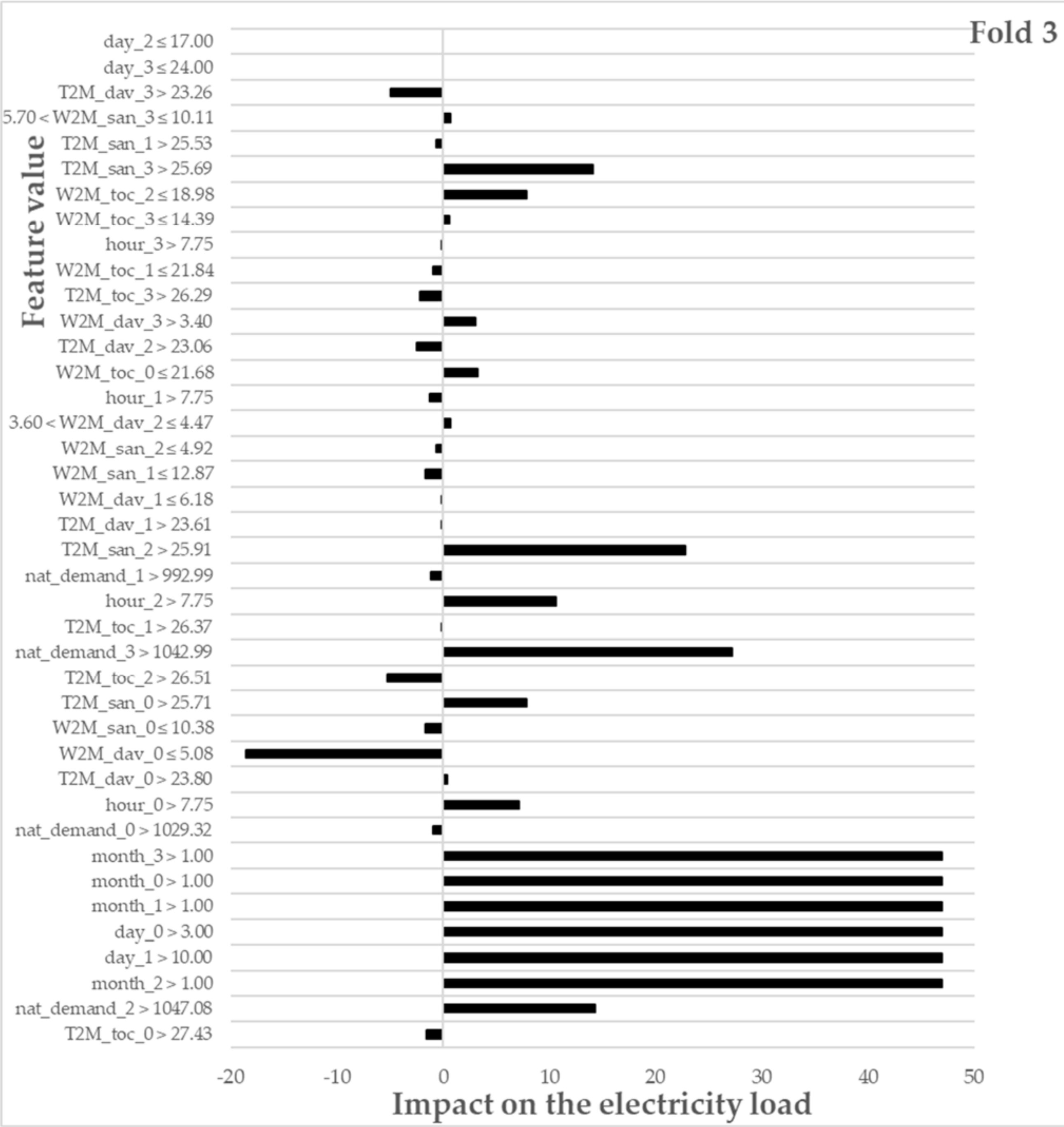

Figure A2. *Cont*.

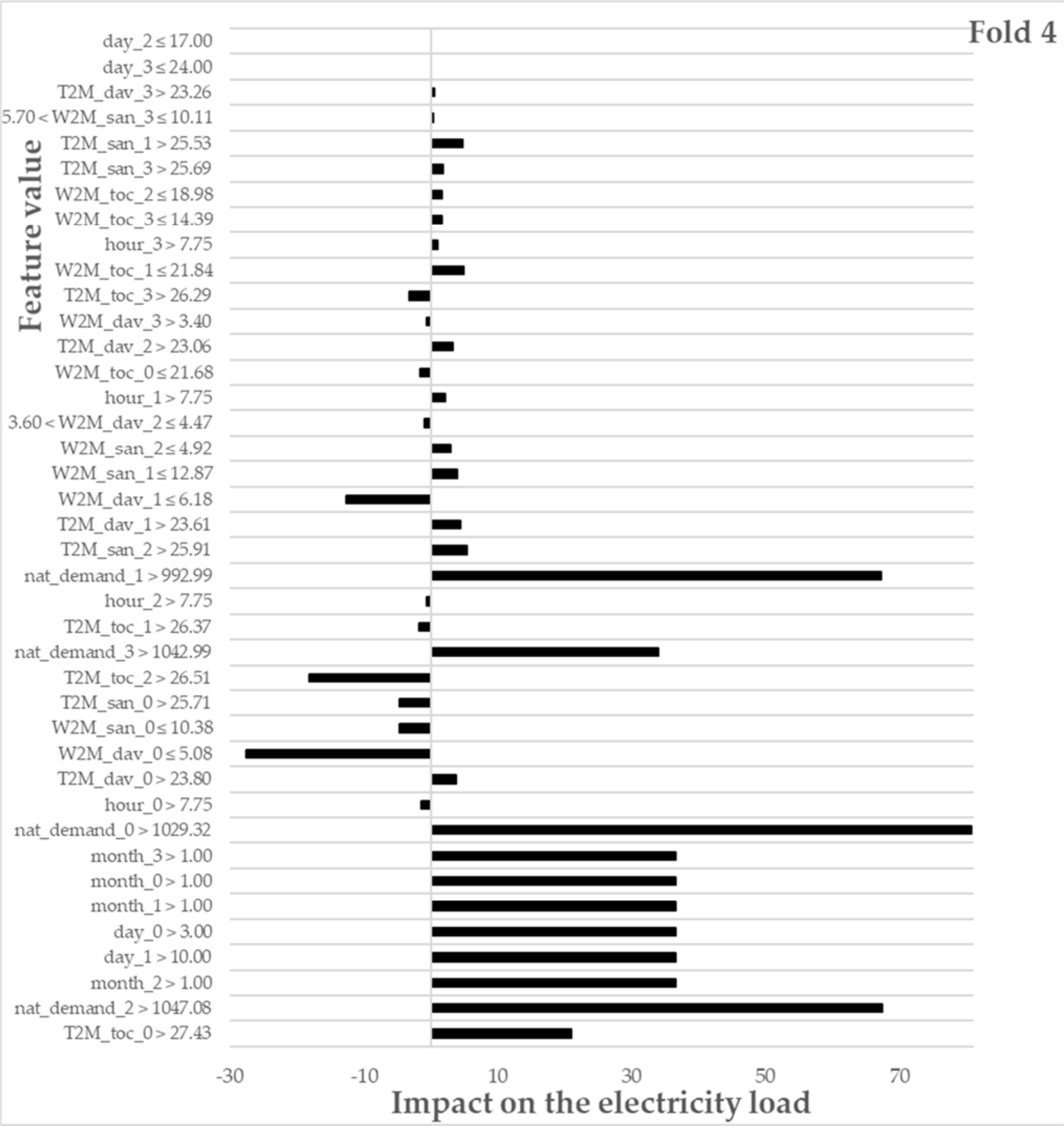

Figure A2. *Cont*.

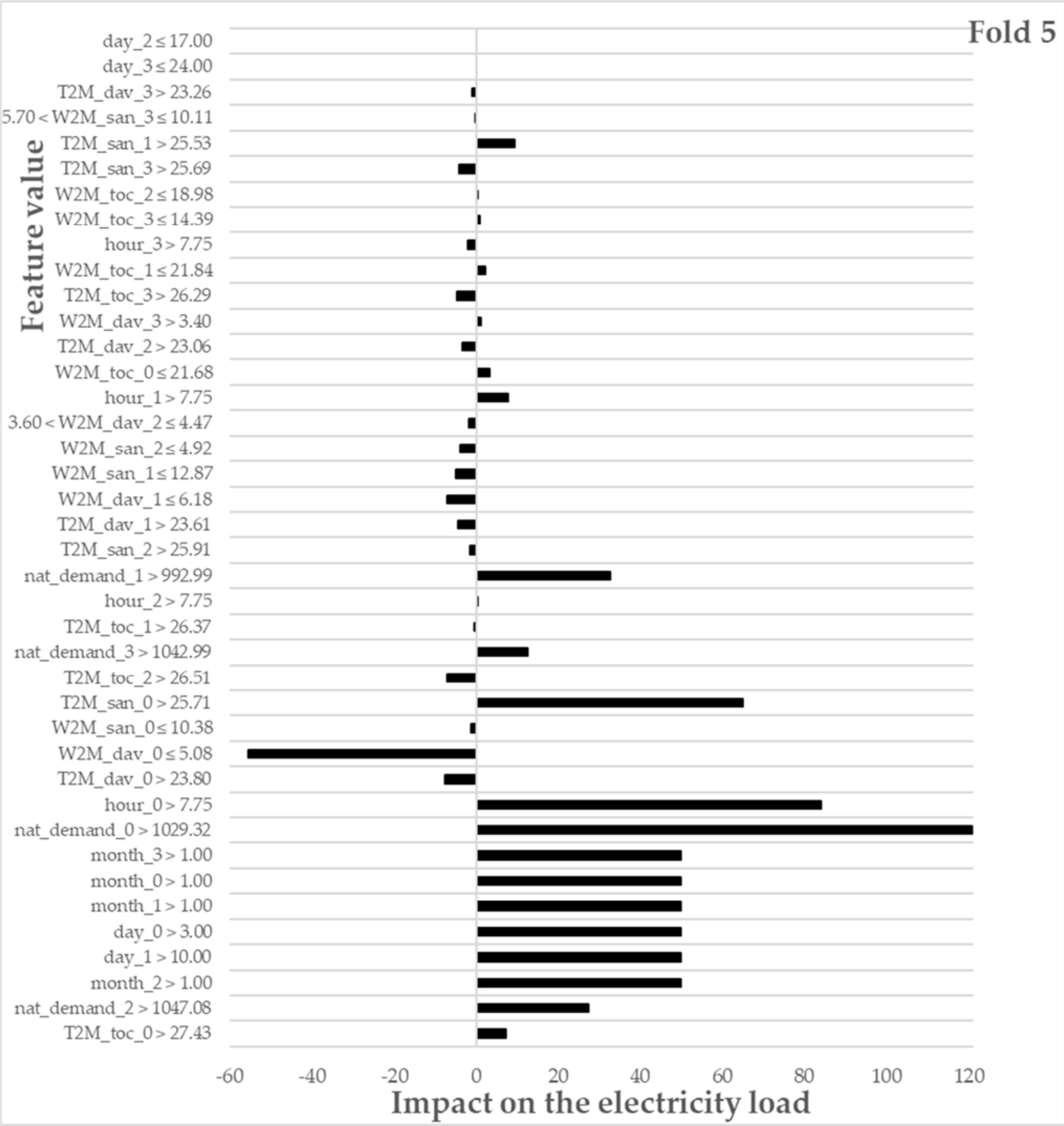

Figure A2. Explanations provided by LIME for LSTM and Fold 2–5 (experiment 2).

References

1. Wang, Y.; Zhang, N.; Chen, X. A short-term residential load forecasting model based on lstm recurrent neural network considering weather features. *Energies* **2020**, *14*, 2737. [CrossRef]
2. Hong, T. Energy Forecasting: Past, Present, and Future. *Foresight Int. J. Forecast.* **2014**, *32*, 43–48.
3. Hong, T.; Fan, S. Probabilistic electric load forecasting: A tutorial review. *Int. J. Forecast.* **2016**, *32*, 914–938. [CrossRef]
4. Dagdougui, H.; Bagheri, F.; Le, H.; Dessaint, L. Neural network model for short-term and very-short-term load forecasting in district buildings. *Energy Build.* **2019**, *203*, 109408. [CrossRef]
5. Seguin, S.; Cote, P.; Audet, C. Self-Scheduling Short-Term Unit Commitment and Loading Problem. *IEEE Trans. Power Syst.* **2016**, *31*, 133–142. [CrossRef]
6. Tovar-Ramírez, C.A.; Fuerte-Esquivel, C.R.; Martínez Mares, A.; Sánchez-Garduño, J.L. A generalized short-term unit commitment approach for analyzing electric power and natural gas integrated systems. *Electr. Power Syst. Res.* **2019**, *172*, 63–76. [CrossRef]

7. Gu, C.; Zhang, Y.; Wang, J.; Li, Q. Joint planning of electrical storage and gas storage in power-gas distribution network considering high-penetration electric vehicle and gas vehicle. *Appl. Energy* **2021**, *301*, 117447. [CrossRef]
8. Liu, D.; Sun, K.; Huang, H.; Tang, P. Monthly load forecasting based on economic data by decomposition integration theory. *Sustainability* **2018**, *10*, 3282. [CrossRef]
9. Lindberg, K.B.; Seljom, P.; Madsen, H.; Fischer, D.; Korpås, M. Long-term electricity load forecasting: Current and future trends. *Util. Policy* **2019**, *58*, 102–119. [CrossRef]
10. Pezzutto, S.; Grilli, G.; Zambotti, S.; Dunjic, S. Forecasting Electricity Market Price for End Users in EU28 until 2020—Main Factors of Influence. *Energies* **2018**, *11*, 1460. [CrossRef]
11. Heydari, A.; Majidi Nezhad, M.; Pirshayan, E.; Astiaso Garcia, D.; Keynia, F.; De Santoli, L. Short-term electricity price and load forecasting in isolated power grids based on composite neural network and gravitational search optimization algorithm. *Appl. Energy* **2020**, *277*, 115503. [CrossRef]
12. Liu, T.; Jin, Y.; Gao, Y. A new hybrid approach for short-term electric load forecasting applying support vector machine with ensemble empirical mode decomposition and whale optimization. *Energies* **2019**, *12*, 1520. [CrossRef]
13. Dudek, G. Short-term load forecasting using neural networks with pattern similarity-based error weights. *Energies* **2021**, *14*, 3224. [CrossRef]
14. Fallah, S.N.; Ganjkhani, M.; Shamshirband, S.; Chau, K. wing Computational intelligence on short-term load forecasting: A methodological overview. *Energies* **2019**, *12*, 393. [CrossRef]
15. Deng, Z.; Wang, B.; Xu, Y.; Xu, T.; Liu, C.; Zhu, Z. Multi-scale convolutional neural network with time-cognition for multi-step short-Term load forecasting. *IEEE Access* **2019**, *7*, 88058–88071. [CrossRef]
16. Huang, Q.; Li, J.; Zhu, M. An improved convolutional neural network with load range discretization for probabilistic load forecasting. *Energy* **2020**, *203*, 117902. [CrossRef]
17. Bak, G.; Bae, Y. Predicting the amount of electric power transaction using deep learning methods. *Energies* **2020**, *13*, 6649. [CrossRef]
18. Massaoudi, M.; Refaat, S.S.; Abu-Rub, H.; Chihi, I.; Oueslati, F.S. PLS-CNN-BiLSTM: An end-to-end algorithm-based savitzky-golay smoothing and evolution strategy for load forecasting. *Energies* **2020**, *13*, 5464. [CrossRef]
19. Biecek, P.; Burzykowski, T. *Explanatory Model Analysis. Explore, Explain and Examine Predictive Models*; Chapman and Hall/CRC: New York, NY, USA, 2021.
20. Hong, T.; Pinson, P.; Wang, Y.; Weron, R.; Yang, D.; Zareipour, H. Energy Forecasting: A Review and Outlook. *IEEE Open Access J. Power Energy* **2020**, *7*, 376–388. [CrossRef]
21. Maçaira, P.M.; Tavares Thomé, A.M.; Cyrino Oliveira, F.L.; Carvalho Ferrer, A.L. Time series analysis with explanatory variables: A systematic literature review. *Environ. Model. Softw.* **2018**, *107*, 199–209. [CrossRef]
22. Kriegeskorte, N.; Golan, T. Neural network models and deep learning. *Curr. Biol.* **2019**, *29*, R231–R236. [CrossRef] [PubMed]
23. Wang, Q.; Zhang, Y.; Zhu, X.; Qiu, Y.; Wang, Y.; Zhang, Z. Short-term load forecasting model based on ridgelet neural network optimized by particle swarm optimization algorithm. In Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, Beijing, China, 24–26 November 2017.
24. Zhou, Z.H. A brief introduction to weakly supervised learning. *Natl. Sci. Rev.* **2018**, *5*, 44–53. [CrossRef]
25. Runge, J.; Zmeureanu, R. A review of deep learning techniques for forecasting energy use in buildings. *Energies* **2021**, *14*, 608. [CrossRef]
26. Rahman, H.; Selvarasan, I.; Jahitha Begum, A. Short-term forecasting of total energy consumption for India-a black box based approach. *Energies* **2018**, *11*, 3442. [CrossRef]
27. Skilton, M.; Hovsepian, F. *The 4th Industrial Revolution: Responding to the Impact of Artificial Intelligence on Business*; Palgrave Macmillan: Cham, Switzerland, 2017.
28. Ortega-Vazquez, M.A.; Kirschen, D.S. Economic impact assessment of load forecast errors considering the cost of interruptions. In Proceedings of the 2006 IEEE Power Engineering Society General Meeting, PES, Montreal, QC, Canada, 18–22 June 2006.
29. Singla, M.K.; Nijhawan, P.; Oberoi, A.S.; Singh, P. Application of levenberg marquardt algorithm for short term load forecasting: A theoretical investigation. *Pertanika J. Sci. Technol.* **2019**, *27*, 1227–1245.
30. Alvarado-Barrios, L.; Rodríguez del Nozal, Á.; Boza Valerino, J.; García Vera, I.; Martínez-Ramos, J.L. Stochastic unit commitment in microgrids: Influence of the load forecasting error and the availability of energy storage. *Renew. Energy* **2020**, *146*, 2060–2069. [CrossRef]
31. Camburu, O.M. Explaining deep neural networks. *arXiv* **2020**, arXiv:2010.01496.
32. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
33. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.
34. Kumar, A.; Saini, P. Effects of partial dependency of features and feature selection procedure over the plant leaf image classification. In *Communications in Computer and Information Science*; Springer: Singapore, 2018; Volume 799.
35. Ramon, Y.; Martens, D.; Provost, F.; Evgeniou, T. A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: SEDC, LIME-C and SHAP-C. *Adv. Data Anal. Classif.* **2020**, *14*, 801–819. [CrossRef]

36. Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. [CrossRef]
37. Hase, P.; Bansal, M. Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior? *arXiv* **2020**, arXiv:2005.01831.
38. Molnar, C. Interpretable Machine Learning. A Guide for Making Black Box Models Explainable. 2019. Available online: https://christophm.github.io/interpretable-ml-book (accessed on 25 August 2021).
39. Ribeiro, M.T.; Singh, S.; Guestrin, C. Anchors: High-precision model-agnostic explanations. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, New Orleans, LA, USA, 2–7 February 2018.
40. Shams Amiri, S.; Mottahedi, S.; Lee, E.R.; Hoque, S. Peeking inside the black-box: Explainable machine learning applied to household transportation energy consumption. *Comput. Environ. Urban Syst.* **2021**, *88*, 101647. [CrossRef]
41. Aguilar Madrid, E. Short-Term Electricity Load Forecasting (Panama Case Study), Mendeley Data, V1. 2021. Available online: https://data.mendeley.com/datasets/byx7sztj59/1 (accessed on 9 July 2021).
42. Madrid, E.A.; Antonio, N. Short-term electricity load forecasting with machine learning. *Information* **2021**, *12*, 50. [CrossRef]
43. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015.
44. Bisong, E. More Supervised Machine Learning Techniques with Scikit-learn. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*; Springer: Berlin/Heidelberg, Germany, 2019.
45. Python Software Foundation. About Python. Available online: http://python.org (accessed on 15 August 2021).
46. Arnold, T.B. kerasR: R Interface to the Keras Deep Learning Library. *J. Open Source Softw.* **2017**, *2*, 296. [CrossRef]
47. Pandas. Available online: https://pandas.pydata.org (accessed on 5 May 2021).
48. Scikit-Learn Scikit-Learn. Available online: https://scikit-learn.org/stable (accessed on 5 May 2021).
49. Ribeiro, M.T. Lime: Explaining the Predictions of Any Machine Learning Classifier. Available online: https://github.com/marcotcr/lime (accessed on 10 May 2021).
50. Butt, F.M.; Hussain, L.; Mahmood, A.; Lone, K.J. Artificial Intelligence based accurately load forecasting system to forecast short and medium-term load demands. *Math. Biosci. Eng.* **2021**, *18*, 400–425. [CrossRef]
51. Naz, A.; Javed, M.U.; Javaid, N.; Saba, T.; Alhussein, M.; Aurangzeb, K. Short-term electric load and price forecasting using enhanced extreme learning machine optimization in smart grids. *Energies* **2019**, *12*, 866. [CrossRef]
52. Webberley, A.; Gao, D.W. Study of artificial neural network based short term load forecasting. In Proceedings of the IEEE Power and Energy Society General Meeting, Vancouver, BC, Canada, 21–25 July 2013.
53. Visani, G.; Bagli, E.; Chesani, F.; Poluzzi, A.; Capuzzo, D. Statistical stability indices for LIME: Obtaining reliable explanations for machine learning models. *J. Oper. Res. Soc.* **2020**, *73*, 91–101. [CrossRef]
54. Zafar, M.R.; Khan, N. Deterministic Local Interpretable Model-Agnostic Explanations for Stable Explainability. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 525–541. [CrossRef]
55. Shi, S.; Du, Y.; Fan, W. Kernel-based LIME with feature dependency sampling. In Proceedings of the International Conference on Pattern Recognition, Milan, Italy, 10–15 January 2021.