Explainable Forecasting Models For Load Forecasting

ANANT TRIVEDI, University of Twente, The Netherlands

Electricity grids face rising demand from population growth and renewable energy integration, creating a crucial need for efficient and explainable forecasting models. The research seeks to evaluate the performance and interpretability of Facebook (FB) Prophet and LSTM (Long-Short-Term Memory) under data constraints compared to Transformer-based solutions such as PatchTST (Patch TS Transformer). This study performed exploratory data analysis (EDA) on a decentralized electricity grid in the Netherlands and predicted the power consumption of a company using the above-stated models. The methodology uses accuracy metrics: (i) Mean Square Error (MSE), (ii) Mean Absolute Error (MAE) and SHapley Additive exPlanations (SHAP) for the explainability of models. FbProphet showed efficiency for short-term forecasting from 15-minute to 1-hour intervals, making it suitable for real-time operational decisions. PatchTST performs consistently for longer horizons (>= 6 hours), benefiting long-term planning and resource allocation. LSTM models require further tuning or additional data to improve accuracy. PatchTST or FbProphet can be selected based on the specific forecasting horizons and availability of training data. In future, these models can forecast load profiles inside a given company to gain insights into the utilization of energy within a company. Additionally, it can expand on explainability tools for Transformer-based models.

Additional Key Words and Phrases: Load forecasting, Explainability, Transformers, FbProphet, LSTM, PatchTST

1 INTRODUCTION

The Dutch electricity grid is becoming increasingly busy, driven by population growth and industrial expansion. Furthermore, the transition from natural gas to electricity and the rise of electric vehicles will increase the electric demand, leading to insufficient grid capacity to transport all the electricity generated in the Netherlands [37]. To develop a flexible energy system that aligns energy supply and demand effectively, the integration of technologies is essential.

Smart grids are electricity networks that use digital technologies, sensors and software to manage the load in real time. They provide an in-depth understanding of electricity usage by identifying key consumption patterns. Smart grids coordinate the needs and capabilities of grid operators and electricity market stakeholders. Advancement in technology motivates researchers to develop machine learning tools that can learn and obtain insights to manage energy infrastructures.

The goal of data-driven techniques is to create models to forecast/predict energy consumption or load demands [8]. Time Series Forecasting (TSF) plays a crucial role in this context [42]. Forecasting energy allows companies to understand future energy demands, infrastructure development, and reducing power losses. It is necessary to predict energy demand for power operators to ensure cost- and energy-efficient decisions about power generation scheduling, system reliability, and power optimization. Finally, producing energy in environmentally friendly ways and meeting the increasing energy demand can be achieved through the insights of forecasting models [43].

Depict these advantages forecast energy demands have limitations. When analyzing Time Series (TS) data, it is crucial to use appropriate methods that account for the temporal dependencies and potential changes in distribution over time. Traditional spatial statistics methods that assume independence and identical distribution are unsuitable for TS data because of inherent restrictions. This research uses solar energy TS data for a specific company where the above assumptions do not hold. For example, power consumption depends on time exhibits non-stationarity and the value at a one-time stamp is often influenced by the value at the previous time point. Additionally, seasonal variations and weather conditions can cause significant fluctuations in energy consumption patterns over time.

Due to these limitations, conventional forecasting methodologies, such as moving averages, trend analysis, and exponential smoothing, are widely used to solve the energy TSF [12]. However, Artificial Neural networks (ANN) are more efficient than statistical models because traditional time series methods rely solely on linear numerical calculations and do not incorporate in-context information [16, 39].

More robust deep learning models such as Recurrent neural network (RNN) [1], LSTM [1, 29, 35, 39] are used to model TS data. The neural networks have played a crucial role in modelling the non-linearity present in the complex load profiles [29]. Deep learning models such as transformers have shown faster training speeds and negligible inference times with comparable accuracies with other SOTA machine learning models [32, 36]. They also provide probabilistic forecasts, which offer insights into the uncertainty of predictions. Understanding the range of possible outcomes is crucial for decision-making in the energy sector [30].

However, these models come with certain limitations. Deep neural networks (DNN) overfit on small datasets and struggle with vanishing/exploding gradient problems. Extensive preprocessing, dealing with missing values, noise reduction, and normalization are also some challenges of DNNs. Usually, these model struggle with explainability because of their complex architecture [19]. The paper by Zeng et al. mentions that transformers suffer from temporal information loss, which is crucial for TSF.

This research compares the performance and explainability of different deep neural networks (DNNs) with other machine learning techniques for load forecasting. It aims to identify the most effective approach for enhancing energy grid efficiency by evaluating these models based on data size constraints, computation efficiency, and explainability.

The first section of the study, related work, discusses various modules and technologies applied for predicting energy demands. The following section formulates the research problem, detailing the main research questions and breaking into focused sub-questions.

TScIT 41, July 5, 2024, Enschede, The Netherlands

Following, the methodology section explains the approaches taken to perform the experiments. The results section then analyzes the outcomes of these experiments. Finally, the paper mentions some improvements that can be made in future work and conclusion of this work.

2 RELATED WORK

Exploratory data analysis (EDA) is a method of evaluating or comprehending sensor data to derive insights and key characteristics [5]. Tyralis et al. (2017) highlight the importance of EDA in understanding TS data. The paper mentions the graphical representation of data showing cyclic and seasonal behaviour in data. The aim is 'to find the unexpected', for instance, identifying misleading patterns or missing values in data. It complements the model building based on the findings of EDA [6]. The book [38] presents advantages like dimensional reduction and regression analysis to identify significant variables and build predictive models via the EDA framework. It enables exploration of the relationship between meteorological variables and power output through correlation analysis. These provide insights into factors influencing power generation, leading to improved prediction accuracy [23].

Innovative and sophisticated techniques such as Transformers networks are emerging in forecasting and predictive modelling [24]. PatchTST is tailored for multivariant TSF and self-supervised representation learning [28, 33]. It exhibits consistently strong predictive performance across various types of TS. Lemishko and Landi (2024) mentions its proficiency in capturing complex patterns, making it superior to other DNNs. Essentially, PatchTST segments the TS into multiple patches to serve as input tokens for transformers. This segmentation reduces the redundancy in data [34]. The model can consider the presence of patches or sudden fluctuations in TS data because of its sophisticated analysis of patterns [15].

While other transformers used point-wise attention, using every time step as an input token, PatchTST aggregates multiple tokens called *patching*. Point-wise attention has a clear disadvantage of loss of local information. For example, a single time or pixel does not inform anything about the TS data or image. Patching has resulted in a reduction in MSE values by 21% and 16.7% on MAE for supervised learning and close to 30% to 50% reduction for similar metrics for unsupervised models [33].

PatchTST uses self-attention mechanisms to focus on relevant patches, enabling it to extract meaningful features and relationships from data [15, 34, 45]. The model also supports *Channel-Independence* (CI). It allows different channel inputs to process independently. For example, if the data includes temperature and humidity columns, there will be two input channels to the Transformer. Predictions from each channel are concatenated for the final predictions. Nie et al. (2023) explains the architecture and working of PatchTST in more detail. The paper [47] mentions that CI proved to work well with Convolutional Neural networks (CNN). Similarly, the paper [33] performed an ablation study on PatchTST, where they combined patching and CI. It showed slightly better results in comparison with other transformers.

Most of the literature regarding PatchTST is long-term forecasting with a horizon of a minimum of 3 days. It is relatively new and has not undergone extensive testing [28]. Hence, it remains to be seen how efficiently it will perform in practical applications and short-horizon forecasting. Building upon this literature, our study aims to compare the predictive abilities of PatchTST for short-term electric load forecasting.

Angelopoulos et al. (2019) mentions the wide use of neural networks for energy forecasting. RNNs are used for quarterly energy demand forecasting of Australia, France, the USA, Spain and Greece [13, 31]. A bidirectional stacked LSTM decomposes TS data to reduce the impact of irregular patterns [4]. The model consists of neural network layers and different memory blocks called cells. The cells are helpful in information retention [27]. The research Cascone et al. (2023) mentions that vanilla LSTM predictions for household energy consumption used an attention mechanism and sequence-to-sequence algorithm. LSTMs have a memory mechanism that stores information learned from past inputs and couples it with the current input to deduce the output [2].

According to Korstanje (2021) in his book, "The LSTM cell adds long-term memory in an even more performant way because it allows even more parameters to be learned. This makes it the most powerful [Recurrent Neural Network] to do forecasting, especially when you have a longer-term trend in your data. LSTMs are one of the state-of-the-art models for forecasting at the moment".

Gupta et al. (2017) mentions using FbProphet for predicting solar energy generation. The model forecasts TS data using an additive approach. Non-linear trends are fitted based on yearly, weekly, and daily seasonality and holiday effects [41]. FbProphet has robust methods for handling missing data and outliers; it works best with TS data having strong seasonal effects and several season-wise historical data [17]. The model uses Seasonal and Trend decomposition using LOESS (STL), which entails identifying and modelling these components using mathematical and statistical methodologies. Stefenon et al. (2023) explains that these methodologies: (i) linear regression used to model the trend component of a TS, (ii) Fourier series used to model the seasonality component of a TS and (iii) binary indication for certain days used to describe the holiday component of a TS.

Prophet results show that it outperforms classical statistical and machine learning approaches [44, 48]. It is designed to handle forecasting on data which has multi-seasonality features. Prophet is a good alternative for forecasting season TS data [4]. This research implements the FbProphet model on electricity data under multiple data constraints to check the performance of the model and its usability in the energy industry.

3 PROBLEM STATEMENT

Power prediction has been extensively researched and has various approaches. It already has been designed, tested and shown to work reasonably well. As mentioned in the previous section, transformers have better accuracy and training times than RNNs and traditional machine learning models for TSF. Nonetheless, transformers have problems like quadratic complexity and inductive bias, which make them less efficient and unsuitable to put the models in production. Additionally, the literature about the explainability of deep learning models is scarce, especially in the field of TS [10]. Explainability

is necessary to characterize accuracy, fairness, transparency and outcomes in AI-powered decision-making [20].

This exploration aims to verify the truthfulness of these claims on deep learning models in the context of TS.

3.1 Research Question

How does the performance and interpretability of transformers and deep learning models for load forecasting change under limited data?

To be able to answer this, we shall further mention the following sub-questions:

3.1.1 ***SQ1:*** How can computational complexity be effectively measured and compared across different machine learning architectures such as FbProphet, PatchTST, and LSTM in the context of load forecasting?

3.1.2 ***SQ2:*** How can popular Explainable AI (XAI) tools be used to improve the interpretability of deep learning models?

4 METHODOLOGY

4.1 Exploratory Data Analysis (EDA)

EDA facilitated understanding of the sensor data and helped to gain insights into a specific company in a solar energy park. Figure 1 represents the workflow in this research. *Power data* was collected from a solar energy business park in the Netherlands. It contains instantaneous and apparent power values with 15-minute time intervals. Literature reviews on solar energy forecasting showed a strong correlation with meteorological data. Therefore, *weather data* was accessed from the Dutch National Weather Service and combined with *power dataset.*

Columns for weekends, holidays, work time, and the year were added to check their correlation with power. Plotting was the most crucial part of EDA, containing *Histograms* and *Box-plots* amongst others. Figure 2 shows the power consumption of a company over a year. It provides a visual of general trends in the times series. For example, the graph shows a discrepancy in power values between 2023 and 2024, as the earlier months of 2024 do not exhibit similar values to those in 2023.

Figure 3 showed seasonality in power values. Therefore, a pattern of peak demand for power occurred at specific intervals; it was future-verified by weekly box plots. Monthly box plots revealed less consumption of energy in the summer season and higher peaks in power during the winter season. As shown in Figure 2, a similar pattern of high power consumption begins in October 2024.

Finally, Dickey-Fuller Tests determined numerically the stationary of the data. The data showed non-stationarity because it failed the null hypothesis test. It proves that the data depends on time and exhibits trends and seasonality. It correlates with the findings above and shown in figure 2 and 3.

4.2 Facebook (Fb) Prophet

FB Prophet uses a decomposed TS model with three main model components: trend, seasonality, and holidays [41].

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t. \quad (1)$$

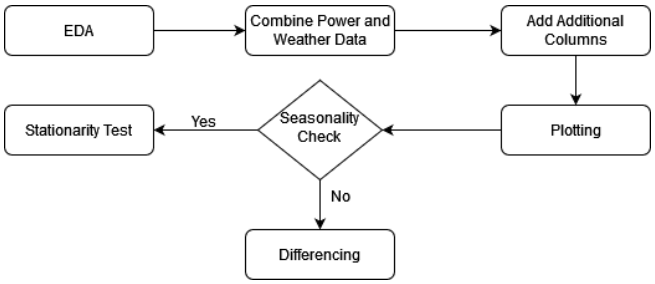

Fig. 1. EDA workflow.

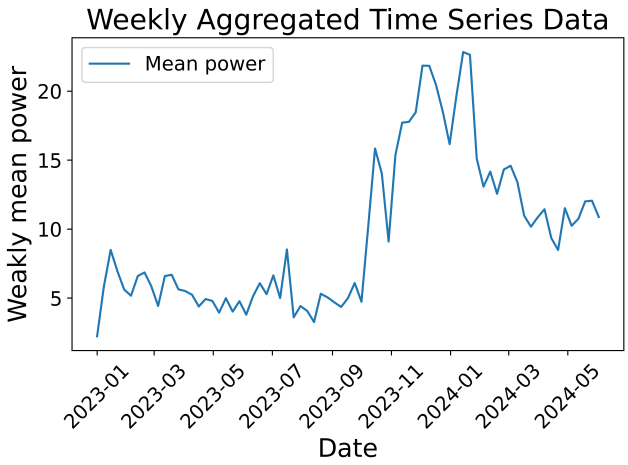

Fig. 2. This figure shows the average weekly power consumption over the entire dataset. Higher peaks are observed starting from October and continue throughout the remainder of 2024.

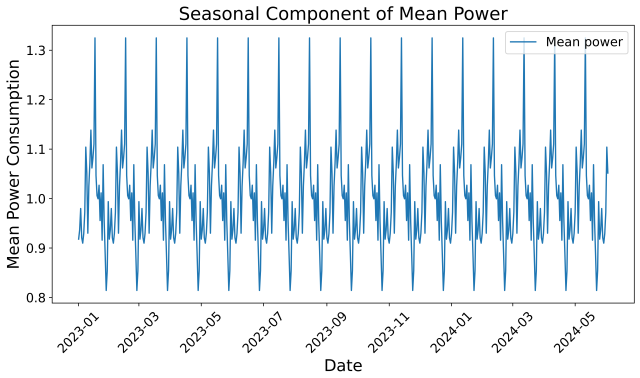

Fig. 3. This figure shows the seasonal composition of the TS data. Monthly trends of peak power consumption can be seen in the data.

The equation (1) includes *g(t)* (trend), *s(t)* (seasonality), and *h(t)* (holidays).

In this study forecasting, the mean power consumption represents the carrying capacity, modelled using the logistic growth model :

$$g(t) = \frac{C}{1 + \exp(-k(t - m))} \tag{2}$$

where C is the carrying capacity, k the growth rate, and m an offset parameter.

Fourier variables handle seasonal patterns, while holidays and events are incorporated as binary indicators, assuming their effects are independent.

The Prophet model requires specific naming conventions for its input. In this study, the data frame columns were renamed to 'ds' for datetime and 'y' for the target variable. The target variable was mean power consumption before being used as inputs to the model. *Adding regressors* in the Prophet model allows it to use external information to improve the accuracy and interpretability of TS forecasts. For instance, including 'work hours' and 'weather data' as regressors before training the model transforms it into a multivariate forecasting approach. After fitting, the model predicts different forecasting horizons and its evaluation metrics were recorded for further analysis.

4.3 Patch TS Transformer (PatchTST)

PatchTST incorporates a framework containing patching and channel-independence. Nie et al. explains the core architecture of PatchTST; they implemented a vanilla Transformer encoder and simple linear head for decoding. Patching splits the original multivariant TS into separate univariant TS. Both use the same model weights during training. So, the model learns the average loss across different TS in the training process. Figure 4 provides an overview of the patching mechanism [25]. In the test channel, the model only makes predictions of the respective channel, so it does not affect the predictions of others. The shared 'Backboard' indirectly learns all the information from different channels.

Fig. 4. By setting patch length(P) = 4 and stride(S) = 2 with sequence length(L) = 8, four patches are generated as shown in this figure.

In this research, a python virtual environment to implement patchTST is created with a *conda* environment and *pip* as our package manager. A bash script was created for installing dependencies, Jupyter kernels, and download IBM git repositories. *HuggingFace* transformer model and *IBM tsfm* package are used for data preprocessing during the implementation of PatchTST. The following steps were taken during the implementation

- First, *load and prepare the data* this includes accessing electricity and weather data. Separate time and forecast columns, determine the amount of historical data for the input as context length and set the forecast horizon and batch size. Split the data into train, valid and test in the 'ForecastDFDataset' class of *IBM tsfm* package.
- Secondly, *configure the PatchTST* and create a model. It sets the weights for the model. Configurations determine the number of input channels based on the forecasting columns and set prediction length equal to the forecasting horizon. The number of attention heads was a constant of 16 and 3 hidden layers were applied in the model.
- Thirdly, for training *transformer* package use *Training Arguments*, *Early Stopping Callback*, and *Trainer* classes.

Overall, the PatchTST model demonstrates a powerful approach to TSF, effectively handling the complexities of multivariate data while maintaining high prediction accuracy and efficiency.

4.4 Long-Short-Term Memory (LSTM)

LSTM uses transfer functions referred to as *activation functions* within the LSTM unit to transform and process input data. There are two types of functions: sigmoid and hyperbolic tangent (tanh).

The sigmoid function forms gates within the unit and is defined as:

$$S(t) = \frac{1}{1 + e^{-t}} \tag{3}$$

The tanh function scales the output of the LSTM, compressing values to the range of -1 to 1. Both functions regulate their outputs within specific bounds: [0, 1] for sigmoid and [-1, 1] for tanh.

Figure 5 illustrates a unit across three-time slices: the current time (t), previous (t-1), and next (t+1).

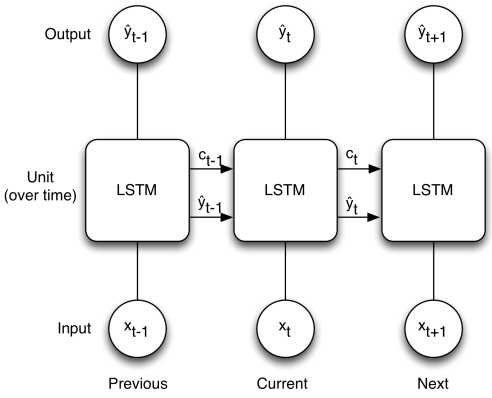

Fig. 5. LSTM unit over three-time slices. The value $\hat{y}$ is the output from the unit; the values ($\mathbf{x}$) are the input to the unit, and the values $\mathbf{c}$ are the context values. The output and context values always feed their output to the next time slice. The context values allow the network to maintain the state between cells.

This research followed the tutorial to construct the LSTM [18]. The implementation involves a windowing mechanism; it takes N values and predicts the future values. A full explanation of windowing can be found in an article by Jaffry (2022).

Forecasting Horizons		15 mins		1 hour		6 hours		12 hours		24 hours		72 hours	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PatchTST	3	0.1365	0.0856	0.3171	0.1532	0.6358	0.2579	0.6704	0.3169	0.8227	0.3661	2.089	0.5157
	6	0.1628	0.0912	0.2789	0.1542	0.3877	0.2395	0.4658	0.2879	0.5282	0.3175	**0.8417**	0.4245
	9	0.1427	0.0833	0.2337	**0.1389**	0.3470	**0.2297**	0.4291	0.2791	0.4728	0.3064	0.8420	**0.4142**
	11	0.0908	**0.0770**	0.1529	0.2555	**0.2103**	0.362	**0.3282**	**0.2605**	**0.3962**	**0.2946**	1.077	0.446
FbProphet	3	17.4846	4.1815	16.7437	4.0913	11.6589	3.1985	16.7605	3.8403	13.8631	3.0502	13.4139	2.9205
	6	13.8677	3.7239	12.9210	3.5937	9.1863	2.8195	10.4903	3.0341	8.6155	2.5659	11.5404	2.4955
	9	**0.0087**	0.0932	**0.0358**	0.1670	2.9682	1.0216	10.6783	2.3569	6.3241	1.8232	55.2313	4.2814
	11	225.7363	15.0245	112.9643	9.7215	70.8524	7.0514	56.5692	6.0387	66.3052	6.5145	85.1121	7.1748
LSTM	3	4.3807	1.1527	4.7026	1.1166	11.8845	2.2535	6.3714	1.4202	8.3304	1.7906	8.1513	1.6080
	6	2.4467	0.8648	4.4097	1.0957	3.9949	1.0639	6.2420	1.3954	5.2666	1.2939	4.0198	1.1483
	9	2.4527	0.8258	2.4398	0.8452	2.3188	0.7928	3.0974	0.9121	5.1845	1.2690	4.4372	1.2655
	11	2.9874	1.1054	2.4820	0.9858	3.6675	1.2458	3.6024	1.1329	3.2171	1.1510	4.0005	1.3012

Table 1. Multivariate long-term forecasting results. The research used training lengths $T \in \{3, 6, 9, 11\}$ months of solar energy dataset. The best results are in **bold** and the second best is underlined.

4.5 XAI (Explainable AI)

For the Fb Prophet model, additional regressors are external variables included in the TSF model to improve its accuracy by providing more explanatory power. The coefficients of these regressors indicate how changes in these variables affect the model's predictions. The *coefficients* are numerical values that quantify the impact of each regressor on the forecasted value. A positive coefficient indicates that as the regressor increases, the model's prediction improves. A negative coefficient indicates that as the regressor increases, the forecasted value decreases. The *credible* for each coefficient identifies whether the regressor is meaningful to the model. If the interval includes zero, the regressor has a statistically significant impact on the forecast [14].

Zhang et al. (2021) uses SHapley Additive exPlanation (SHAP) to measure the explainability of the forecasting models. SHAP value measures the impact of each feature on the model's prediction results; it increases the transparency of the black box forecasting models such as neural networks. This research aims to use the same techniques on LSTM to validate the explainability of the energy TS dataset.

4.6 Performance evaluation index

This research trained the models on different batch sizes and forecast horizons to check the model performance. The training sizes were 3, 6, 9 and 11 months of data and forecasting horizons for 15 mins, 1, 6, 12, 24, and 72 hours. Essentially, it helped to evaluate model performance under different data constraints. It was important to check how much data was needed for a specific model to produce accurate results.

The configuration of the device that this research used is:

(1) **CPU Server**
CPU: 64 cores/128 threads /1024 Gb memory.
(2) **GPU Server**
Nvidia A10 / CPU: 56 cores/112 threads / 256 Gb memory.

The general evaluation metrics of the TS prediction models are as follows:
Mean absolute error (MAE)

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|p_i - \hat{p}_i| \tag{4}$$

Mean square error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{5}$$

5 RESULTS

5.1 Multivariant forecasting

Table 1 provides insight into PatchTST, LSTM and FbProphet models' predictions with the actual power values for each forecasting horizon, using MSE and MAE as evaluation metrics.

5.1.1 *General Observations.*

- **Best Performances:** FbProphet achieves the lowest MSE for the 15-minute and 1-hour horizons, while PatchTST performs better for longer horizons when more training data is available. This result correlates with transformer-based models requiring more information to make forecasts.
- **Training Data Impact:** Across all models, more training data generally leads to better performance, as indicated by lower MSE and MAE values.

5.1.2 Analysis of Forecasting Performance. For a prediction horizon of 15 minutes, FbProphet outperforms PatchTST in terms of MSE (FbProphet MSE = 0.008700, PatchTST MSE = 0.0908), indicating better prediction accuracy. However, PatchTST has a slightly lower MAE (PatchTST MAE = 0.0770, FbProphet MAE = 0.093200), suggesting that the average absolute error is smaller.

Moving to 1-hour predictions, PatchTST achieves a lower MAE when trained on more data (PatchTST MAE = 0.127100 for 11 months, PatchTST MAE = 0.138900 for 9 months), indicating fewer large deviations in predictions. However, FbProphet significantly outperforms PatchTST in terms of MSE (FbProphet MSE = 0.035800), suggesting better accuracy compared to PatchTST (PatchTST MSE = 0.152900 for 11 months, PatchTST MSE = 0.233700 for 9 months).

FbProphet is designed to handle TS data with seasonality and seasonal components are more predictable. The model's assumptions about seasonality and trends might not hold for longer horizons. The lower MAE value of PatchTST indicates that on average the model performs better than FbProphet. Additionally, PatchTST's performance improves with more training data, as seen in 1-hour horizon predictions, with reduced MAE, indicating fewer large deviations.

As the forecasting horizon increases to 6 hours, performance degrades with less training data for PatchTST (PatchTST MSE = 0.2103 for 11 months, PatchTST MSE = 0.347000 for 9 months), yet PatchTST shows consistent performance across different training sizes. Similar patterns were observed in 12-hour predictions (PatchTST MSE = 0.328200 for 11 months, PatchTST MSE = 0.429100 for 9 months).

Day-long forecasting remains relatively stable with increased forecasting horizons but shows a slight increase in error metrics with less training data (PatchTST MSE = 0.396200 for 11 months, PatchTST MSE = 0.472800 for 9 months).

However, the LSTM model shows a significant increase in error metrics with a 72-hour horizon (LSTM MSE = 4.0005 for 11 months, LSTM MSE = 8.1513 for 3 months) and performs worse with less training data, indicating higher sensitivity to the amount of training data. As the prediction horizon increases, the model's error accumulates, leading to higher MSE and MAE values.

5.1.3 Analysis of Training Time. Figures 6, **??** show the training time for the forecasting horizons and different training sizes. The results show that an increase in training data decreases the training time of PatchTST. In contrast to FbProphet the training time increases with larger training size.

The larger training size of PatchTST for smaller horizons can be correlated with *patching*. In this research, the forecasting horizon and patch length remained the same. However, because 15-minute intervals create smaller patches or windows compared to 24 hours, training with 15-minute patches involves more data and consequently takes longer. Longer sequences during patching avoids memory constraints and facilitate quicker training speeds.

Despite PatchTST's accurate results as discussed in previous sections, it is computationally costly. The training at its peak was close to 5 hours to predict a horizon of 15 minutes. However, this training time was reduced by 80 % when a predicting window was increased to 12 or 24 hours.

5.2 XAI study

5.2.1 FbProphet. Figure 8 represents the bar plot of coefficient-credible intervals for Fb Prophet. Each bar represents the coefficient value of a regressor and the length of the bar indicates the magnitude of the coefficient. The black dots and horizontal lines through them represent the credible intervals of the coefficients. The following are the results drawn from the coefficient-credible intervals:

(1) **Significant Regressor:** All regressors have narrow credible intervals that do not include zero, indicating that they all have statistical effects on the predictions.
(2) **Positive Impact:** 'Month', 'Week Number', 'Is Weekend', and 'Is Peak' all have positive coefficients, meaning they increase the forecast value.

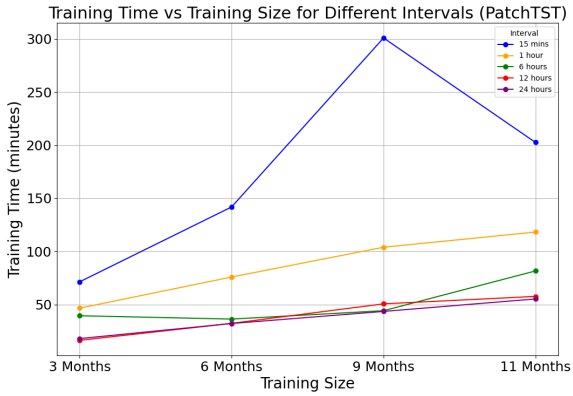

Fig. 6. This figure shows the training time of different forecasting horizons for PatchTST.

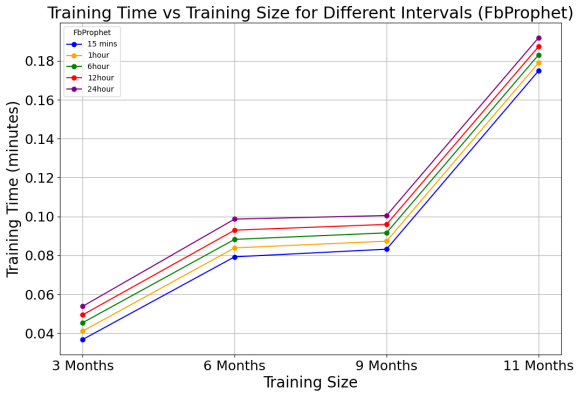

Fig. 7. This figure shows the training time of different forecasting horizons for Facebook Prophet.

(3) **Negative Impact:** 'Quater', 'Is Holiday', 'Is Weekend', 'T' (temperature), and 'SQ' (sun), 'DR' (wind) have negative coefficients, meaning they decrease the forecast value.
(4) **Magnitude of Impact:** The most impactful regressors are 'Is Weekend' and 'Is peak', and given their large coefficients, indicating they significantly influence the forecast values. These regressors have logical inferences. During the weekend no company uses their offices leading to a low/minimal power consumption. Contrarily, it makes sense for 'Is peak' to have the highest significance because a peak in energy is during the morning work hours of an office. Figure 9 shows the increase of consumption from 8:00 am to 9:30 am, reflecting the start of the workday when employees arrive at the office, turn on lights, computers, and other office equipment, leading to a noticeable spike in energy demand.

The model's performance can be improved with *negative impact* regressors removed.

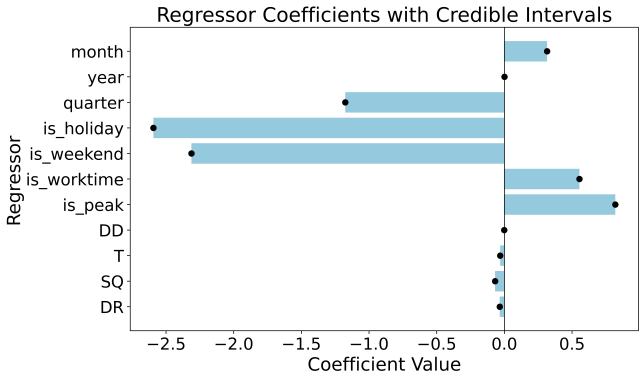

Fig. 8. This figure shows the coefficient with credible intervals for the Facebook Prophet.

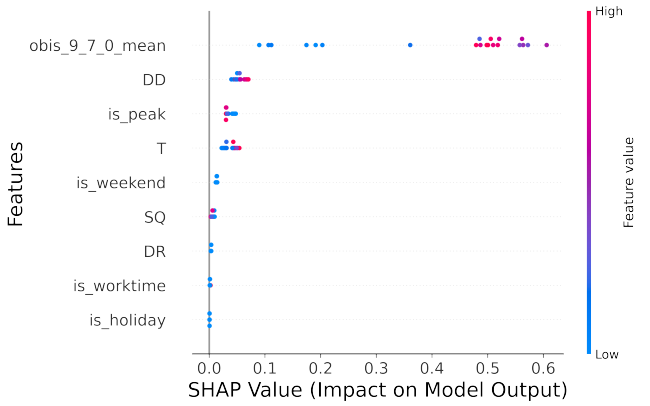

Fig. 10. This figure shows the Shap summary plot for the LSTM.

Weekly and daily seasonality decomposition can be seen in figure 9. Understanding these seasonal patterns helps explain what the model is learning from the data. The model identifies regular patterns in the data. Weakly seasonality shows how the forecasted value is affected by the day of the week, with higher values on Monday and Tuesday and lower on Saturday and Sunday. Significant peaks in the morning around 10:00 AM and lower values in the early morning and late afternoons are shown in daily seasonality.

These patterns are incorporated into the model to improve its accuracy. The model adjusts its predictions based on the learned weekly and daily patterns, leading to more accurate forecasts that reflect real-world behaviour. The prophet model performs better by explicitly modelling seasonality with strong seasonality components.

The model learns the regular fluctuations in the data, leading to reliable forecasts. These results are later helpful in businesses and organizations in making informed decisions, optimizing operations and improving efficiency based on the predicted trends.

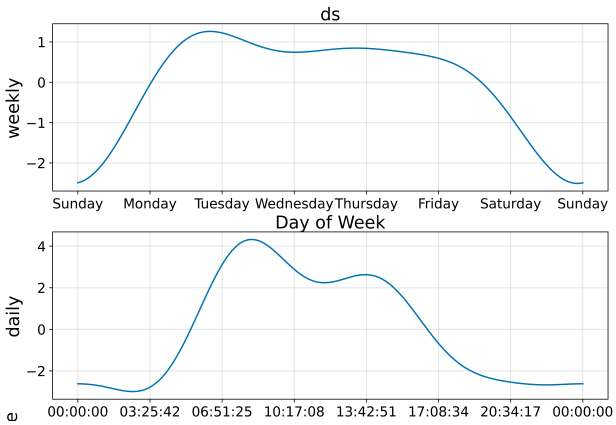

Fig. 9. This figure shows daily and weekly forecast components for the Facebook Prophet.

5.2.2 LSTM. TimeSHAP is a model-agnostic recurrent explainer that builds upon KernelSHAP and extends it to the sequential domain [7]. SHAP value represents the contribution of each feature to a model's prediction for an individual data point. Positive SHAP values push the predictions higher and negative values pull it lower. Every instance of the training dataset appears as its point. Examining how the SHAP values are distributed reveals how a variable may influence the model's predictions [11].

Figure 10 shows a summary of plot SHAP values for the LSTM model. The Y-axis indicates the feature names in order of importance from top to bottom. Mean power consumption (*obis_9_7_0_mean*) has the highest importance for the model's prediction. This result makes sense because LSTM uses windowing which takes mean power as an input in training. Mean power has red rots for for larger SHAP values meaning that higher values of mean power result in a positive SHAP. Higher power values can be corresponded with peak consumption patterns during the day which is a good indication of the model's learning. Positive SHAP values indicate features that push the prediction towards positive outcomes.

The summary plot in figure 10 indicates that rain (*DD*), 'is_worktime', and 'is_holiday' are the least significant features for LSTM model. These features have their lower values close to a SHAP value of 0.0, they do not help in predicting the power consumption values. These results partially counter-checks with Fb Prophets feature importance results, as shown in figure 8.

Wind (*DD*) and Temperature (*T*) showed a slight positive impact on SHAP value with higher values. The fluctuation in temperature and wind has a positive significance in power consumption. Higher wind speeds can lead to increased power consumption. For example, chilly wind leads to higher heating demand in cold climates. Similarly, higher temperatures increase the demand for air conditioning, also driving up power consumption.

6 FUTURE WORK

This research focused on forecasting the power consumption of a specific company in a solar energy park. It performed TSF using PatchTST, LSTM and FbProphet models. In the future, the same models can be used to forecast the load profiles inside a given company

to gain insights into the utilization of energy within a company. For example, if a company has multiple devices such as servers, heaters and charging stations, these models can predict the energy production from these devices. The business can optimize its operation schedules to minimise energy use during peak hours, thus reducing overall energy costs.

Different batch sizes could have been experimented with to fine-tune the models. Especially, with PatchTST it would be interesting to observe if batch sizes affect the inference times of the model for multiple forecasting horizons. For LSTM, multiple-layer architectures can be used to verify the accuracy of the model. Additionally, more data could have been used to find more patterns for the model to predict better.

This research can be expanded by searching and implementing explainability tools for Transformer models specifically in the field of TSF. Islam et al. (2024) proposes a framework for transformers that helps to understand the impact of past observations, but also predict their impact on future values. It showed changes in future importance over past time steps but also predicted the sensitivity of these features in future horizons.

7 CONCLUSION

This research investigated the performance of PatchTST, LSTM, and FbProphet models for multivariate time-series forecasting in the context of energy sector applications. The study evaluated these models across various forecasting horizons, assessing their sensitivity to training data size, and exploring their interpretability through XAI techniques.

FbProphet demonstrated superior accuracy for short-term forecasting intervals ranging from 15 minutes to 1 hour, leveraging its ability to capture daily and weekly seasonal patterns. In contrast, PatchTST excelled in longer forecasting horizons, particularly 6 to 24 hours, benefiting from increased training data. Leveraging FbProphet for short-term forecasting can optimize energy operations, facilitating responsive adjustments to demand fluctuations and market dynamics.

Despite the lack of explainability tools for Transformers, this research successfully discovered and implemented XAI methods for FbProphet and LSTM. Coefficient values for regressors helped to find the positive impact columns for the model. 'Is peaktime' was the most impactful regressor and had the maximum positive impact on the model's accuracy.

Analysis of SHAP values for the LSTM model revealed mean power consumption as the most significant predictor, aligning with the windowing approach. Additionally, the positive influence of wind and temperature fluctuations on SHAP values emphasises their role in driving power consumption, reflecting real-world heating and cooling demands.

This research contributes an understanding of multivariate time-series forecasting models in the energy sector, highlighting their diverse strengths and applications while testing their performance across different forecasting horizons and under varying data conditions.

8 ACKNOWLEDGEMENT

The author extends gratitude to Dr. Faizan Ahmed for his invaluable guidance and support throughout this thesis. Additionally, sincere thanks are due to Saxion University of Applied Sciences, Enschede, for providing the necessary resources that contributed to the successful completion of this research.

A AI USE

During the preparation of this work, the author(s) used [ChatGPT] to correct grammatical errors and structure the LateX file. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the work.

B REFERENCES

[1] Hyung Keun Ahn and Neungsoo Park. 2021. Deep RNN-Based Photovoltaic Power Short-Term Forecast Using Power IoT Sensors. Energies 14, 2 (2021). https://doi.org/10.3390/en14020436

[2] Ibtissam Amalou, Naoual Mouhni, and Abdelmounaim Abdali. 2022. Multivariate time series prediction by RNN architectures for energy consumption forecasting. Energy Reports 8 (2022), 1084–1091. https://doi.org/10.1016/j.egyr.2022.07.139 Technologies and Materials for Renewable Energy, Environment and Sustainability.

[3] Dimitrios Angelopoulos, Yannis Siskos, and John Psarras. 2019. Disaggregating time series on multiple criteria for robust forecasting: The case of long-term electricity demand in Greece. European Journal of Operational Research 275, 1 (2019), 252–265. https://doi.org/10.1016/j.ejor.2018.11.003

[4] S. Arslan. 2022. A hybrid forecasting model using LSTM and Prophet for energy consumption with decomposition of time series data. PeerJ Computer Science 8 (2022), e1001. https://doi.org/10.7717/peerj-cs.1001

[5] Roberto Barriga, Miquel Romero, Houcine Hassan, and David F. Nettleton. 2023. Energy Consumption Optimization of a Fluid Bed Dryer in Pharmaceutical Manufacturing Using EDA (Exploratory Data Analysis). Sensors 23, 8 (2023). https://doi.org/10.3390/s23083994

[6] John T. Behrens. 1997. Principles and procedures of exploratory data analysis. Psychological Methods 2, 2 (1997), 131–160. https://doi.org/10.1037/1082-989X.2.2.131

[7] João Bento, Pedro Saleiro, André F. Cruz, Mário A.T. Figueiredo, and Pedro Bizarro. 2021. TimeSHAP: Explaining Recurrent Models through Sequence Perturbations. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining. ACM. https://doi.org/10.1145/3447548.3467166

[8] Mathieu Bourdeau, Xiao qiang Zhai, Elyes Nefzaoui, Xiaofeng Guo, and Patrice Chatellier. 2019. Modeling and forecasting building energy consumption: A review of data-driven techniques. Sustainable Cities and Society 48 (2019), 101533. https://doi.org/10.1016/j.scs.2019.101533

[9] Lucia Cascone, Saima Sadiq, Saleem Ullah, Seyedali Mirjalili, Hafeez Ur Rehman Siddiqui, and Muhammad Umer. 2023. Predicting Household Electric Power Consumption Using Multi-step Time Series with Convolutional LSTM. Big Data Research 31 (2023), 100360. https://doi.org/10.1016/j.bdr.2022.100360

[10] Hila Chefer, Shir Gur, and Lior Wolf. 2021. Transformer Interpretability Beyond Attention Visualization. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 782–791. https://doi.org/10.1109/CVPR46437.2021.00084

[11] Aidan Cooper. 2024. A Non-Technical Guide to Interpreting SHAP Analyses. https://www.aidancooper.co.uk/a-non-technical-guide-to-interpreting-shap-analyses/

[12] Lars Dannecker. 2015. Energy Time Series Forecasting: Efficient and Accurate Forecasting of Evolving Time Series from the Energy Domain. 1–231 pages. https://doi.org/10.1007/978-3-658-11039-0

[13] L. Ekonomou. 2010. Greek long-term energy consumption prediction using artificial neural networks. Energy 35, 2 (2010), 512–517. https://doi.org/10.1016/j.energy.2009.10.018 ECOS 2008.

[14] Facebook, Inc. 2024. Seasonality, Holiday Effects, and Regressors. https://facebook.github.io/prophet/docs/seasonality,_holiday_effects,_and_regressors.html Accessed: 2024-06-28.

[15] Aslan Feng. 2023. Patch-aware Long-term Weather Forecasting. In Proceedings of the 2023 4th International Conference on Big Data and Social Sciences (ICBDSS 2023). Atlantis Press, 475–484. https://doi.org/10.2991/978-94-6463-276-7_50

[16] Georgios Fotis, Nenad Sijakovic, Mileta Zarkovic, Vladan Ristic, Aleksandar Terzic, Vasiliki Vita, Magda Zafeiropoulou, Emmanouil Zoulias, and Theodoros Maris. 2023. Forecasting Wind and Solar Energy Production in the Greek Power System

using ANN Models. WSEAS TRANSACTIONS ON POWER SYSTEMS 18 (12 2023), 373–391. https://doi.org/10.37394/232016.2023.18.38

[17] Rahul Gupta, Anil Kumar Yadav, SK Jha, and Pawan Kumar Pathak. 2022. Time Series Forecasting of Solar Power Generation Using Facebook Prophet and XG Boost. In 2022 IEEE Delhi Section Conference (DELCON). 1–5. https://doi.org/10.1109/DELCON54057.2022.9752916

[18] Greg Hogg. 2021. LSTM Time Series Forecasting Tutorial in Python. https://www.youtube.com/watch?v=c0k-YLQGKjY Accessed: 2024-06-28.

[19] Huntresselle. 2020. Deep Learning Techniques for Time Series Forecasting: A Comprehensive Guide. https://medium.com/@huntresselle/deep-learning-techniques-for-time-series-forecasting-a-comprehensive-guide-f4c539d19342 Accessed: 2024-06-12.

[20] IBM. 2024. What is Explainable AI (XAI)? https://www.ibm.com/topics/explainable-ai Accessed: 2024-06-27.

[21] Md Khairul Islam, Tyler Valentine, Timothy Joowon Sue, Ayush Karmacharya, Luke Neil Benham, Zhengguang Wang, Kingsley Kim, and Judy Fox. 2024. Interpreting Time Series Transformer Models and Sensitivity Analysis of Population Age Groups to COVID-19 Infections. arXiv preprint arXiv:2401.15119 (2024).

[22] Shan Jaffry. 2022. Time Series Forecasting Using Windowing Method, with LSTM. https://syedshan85.medium.com/window-based-time-series-forecasting-with-keras-lstm-6b664e7c54c4 Accessed: 2024-06-28.

[23] Sergii Kavun and Alina Zamula. 2023. Exploratory Data Analysis in Wind Energy Datasets. In 2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek). 1–6. https://doi.org/10.1109/KhPIWeek61412.2023.10312958

[24] Damian Kisiel and Denise Gorse. 2022. Portfolio Transformer for Attention-Based Asset Allocation. arXiv:2206.03246 [q-fin.PM] https://arxiv.org/abs/2206.03246

[25] Lalf Klein. 2024. PatchTST for Time Series Forecasting: Original Results and New Single-Channel Experiments. https://medium.com/@lalf_klein/patchtst-for-time-series-forecasting-original-results-and-new-single-channel-experiments-f375699f7b91 Accessed: 2024-06-27.

[26] Joos Korstanje. 2023. Advanced Forecasting with Python. Springer, Maisons Alfort, France. https://link.springer.com/book/10.1007/978-1-4842-7150-6

[27] Ashutosh Kumar Dubey, Abhishek Kumar, Vicente García-Díaz, Arpit Kumar Sharma, and Kishan Kanhaiya. 2021. Study and analysis of SARIMA and LSTM in forecasting time series data. Sustainable Energy Technologies and Assessments 47 (2021), 101474. https://doi.org/10.1016/j.seta.2021.101474

[28] Tetiana Lemishko and Alexandre Landi. 2024. A Comparative Analysis of LSTM versus PatchTST in Predictive Modeling of Asset Prices. SSRN (2024). https://doi.org/10.2139/ssrn.4432183 [1]tetiana.lemishko@balanced-research.com, [2]alexandre.landi@balanced-research.com.

[29] Feng Li, Xiaowei Yu, Xin Tian, and Zilong Zhao. 2021. Short-Term Load Forecasting for an Industrial Park Using LSTM-RNN Considering Energy Storage. In 2021 3rd Asia Energy and Electrical Engineering Symposium (AEEES). 684–689. https://doi.org/10.1109/AEEES51875.2021.9403118

[30] Bryan Lim and Stefan Zohren. 2021. Time-series forecasting with deep learning: a survey. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 379, 2194 (Feb. 2021), 20200209. https://doi.org/10.1098/rsta.2020.0209

[31] Tao Liu, Zehan Tan, Chengliang Xu, Huanxin Chen, and Zhengfei Li. 2020. Study on deep reinforcement learning techniques for building energy consumption forecasting. Energy and Buildings 208 (2020), 109675. https://doi.org/10.1016/j.enbuild.2019.109675

[32] Erick Giovani Sperandio Nascimento, Talison A.C. de Melo, and Davidson M. Moreira. 2023. A transformer-based deep neural network with wavelet transform for forecasting wind speed and wind energy. Energy 278 (2023), 127678. https://doi.org/10.1016/j.energy.2023.127678

[33] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. arXiv:2211.14730 [cs.LG]

[34] PeiSong Niu, Tian Zhou, Xue Wang, Liang Sun, and Rong Jin. 2024. Attention as Robust Representation for Time Series Forecasting. arXiv:2402.05370 [cs.LG]

[35] Shaoqian Pei, Hui Qin, Liqiang Yao, Yongqi Liu, Chao Wang, and Jianzhong Zhou. 2020. Multi-Step Ahead Short-Term Load Forecasting Using Hybrid Feature Selection and Improved Long Short-Term Memory Network. Energies 13, 16 (2020). https://doi.org/10.3390/en13164121

[36] Zhuyi Rao and Yunxiang Zhang. 2020. Transformer-based power system energy prediction model. 913–917. https://doi.org/10.1109/ITOEC49072.2020.9141649

[37] Ebbe Rogge. 2024. The European energy crisis, the Dutch TTF, and the market correction mechanism: a financial markets perspective. The Journal of World Energy Law Business 17, 3 (04 2024), 184–200. https://doi.org/10.1093/jwelb/jwae004 arXiv:https://academic.oup.com/jwelb/article-pdf/17/3/184/57840172/jwae004.pdf

[38] T. Santhoshkumar and S. Vanila. 2024. Exploratory Data Analysis and Energy Predictions With Advanced AI and ML Techniques. 336–370. https://doi.org/10.4018/979-8-3693-1586-6.ch018

[39] Ashish Sedai, Rabin Dhakal, Shishir Gautam, Anibesh Dhamala, Argenis Bilbao, Qin Wang, Adam Wigington, and Suhas Pol. 2023. Performance Analysis of Statistical, Machine Learning and Deep Learning Models in Long-Term Forecasting of Solar Power Production. Forecasting 5 (02 2023), 256–284. https://doi.org/10.3390/forecast5010014

[40] Stefano Frizzo Stefenon, Laio Oriel Seman, Viviana Cocco Mariani, and Leandro dos Santos Coelho. 2023. Aggregating Prophet and Seasonal Trend Decomposition for Time Series Forecasting of Italian Electricity Spot Prices. Energies 16, 3 (2023). https://doi.org/10.3390/en16031371

[41] Sean Taylor and Benjamin Letham. 2017. Forecasting at scale. https://doi.org/10.7287/peerj.preprints.3190v2

[42] J. F. Torres, M. J. Jiménez-Navarro, F. Martínez-Álvarez, and A. Troncoso. 2021. Electricity Consumption Time Series Forecasting Using Temporal Convolutional Networks. In Advances in Artificial Intelligence, Enrique Alba, Gabriel Luque, Francisco Chicano, Carlos Cotta, David Camacho, Manuel Ojeda-Aciego, Susana Montes, Alicia Troncoso, José Riquelme, and Rodrigo Gil-Merino (Eds.). Springer International Publishing, Cham, 216–225.

[43] Typeset. n.d.. Why is Energy Load Forecasting Important? https://typeset.io/questions/why-is-energy-load-forecasting-important-o8jl4su2i7

[44] Argyrios Vartholomaios, Stamatis Karlos, Eleftherios Kouloumpris, and Grigorios Tsoumakas. 2021. Short-Term Renewable Energy Forecasting in Greece Using Prophet Decomposition and Tree-Based Ensembles. In Database and Expert Systems Applications - DEXA 2021 Workshops, Gabriele Kotsis, A. Min Tjoa, Ismail Khalil, Bernhard Moser, Atif Mashkoor, Johannes Sametinger, Anna Fensel, Jorge Martinez-Gil, Lukas Fischer, Gerald Czech, Florian Sobieczky, and Sohail Khan (Eds.). Springer International Publishing, Cham, 227–238.

[45] Huiju Yi, Shengcai Zhang, Dezhi An, and Zhenyu Liu. 2024. PatchesNet: PatchTST-based multi-scale network security situation prediction. Knowledge-Based Systems 299 (2024), 112037. https://doi.org/10.1016/j.knosys.2024.112037

[46] Yuyi Zhang, Ruimin Ma, Jing Liu, Xiuxiu Liu, Ovanes Petrosian, and Kirill Krinkin. 2021. Comparison and Explanation of Forecasting Algorithms for Energy Time Series. Mathematics 9 (11 2021), 2794. https://doi.org/10.3390/math9212794

[47] Y. Zheng, Q. Liu, Enhong Chen, Y. Ge, and J.L. Zhao. 2014. Time series classification using multi-channels deep convolutional neural networks. WAIM 2014. LNCS 8485 (01 2014), 298–310.

[48] Landi Zhoul, Ming Chenl, and Qingjian Ni. 2020. A hybrid prophet-LSTM model for prediction of air quality index. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 595–601.