Architecture-Agnostic Time-Step Boosting: a Case Study in Short-Term Load Forecasting*

Ioannis Pierros[1[0000−0001−9753−8680]] and Ioannis Vlahavas[1[0000−0003−3477−8825]]

[1] Aristotle University of Thessaloniki, Thessaloniki 54124, Greece
ipierros@csd.auth.gr
[2] Aristotle University of Thessaloniki, Thessaloniki 54124, Greece
vlahavas@csd.auth.gr

Abstract. Time series forecasting is important for short-term operations planning and deciding the long-term growth strategy of a company. High accuracy is clearly the hardest challenge, though fast training is also important because a model can go through thousands of iterations. In this paper, we propose Time-Step Boosting, a streamlined methodology that can be applied to any type of neural network for demand forecasting, that adjusts the model's weights during training to optimize it towards the time steps that are most difficult to predict. First, we calculate the time step error and afterwards train the model anew using the errors as weights when calculating the loss during training. We apply Time-Step Boosting on short-term demand forecasting, a task that is necessary for the smooth operation of all components in the energy sector. Deviations require costly emergency actions to reset the production-demand balance and avoid damaging the substations or even overloading the electrical grid. Even though forecasting systems have advanced in recent years, they oftentimes fail to accurately predict the peaks and lows which admittedly are of utmost importance. Our methodology demonstrates considerable convergence speed and forecasting performance improvements on next-day hourly load forecasting for multiple European countries and 6 states of the U.S. with Multilayer Perceptrons, Long-Short Term Memory networks, Convolutional Neural Networks and state-of-the-art models, showcasing its applicability on more complex architectures.

Keywords: Machine Learning · Time Series · Forecasting · Neural Networks · Energy Demand · Short-Term Load Forecasting.

1 Introduction

Short-term load forecasting (STLF) is necessary for the smooth operation of all components in the energy sector, from the generation and transmission of electricity to its distribution and consumption. Failure to correctly forecast the demand and especially under-forecasting it, requires emergency actions that come

* This research was carried out as part of the project KMP6-0072250 under the framework of the Action "Investment Plans of Innovation" of the Operational Program "Central Macedonia 2014-2020", that is co-funded by the European Regional Development Fund and Greece.

The final authenticated version is available online at https://doi.org/10.1007/978-3-031-15934-3_46.

with a significant cost, while large differentials between forecast and demand can even lead to blackouts. On the other hand, overestimating the energy demand elicits an overproduction of electricity, which, besides the unnecessary costs of producing unused electricity, also risks overloading the electrical grid and damaging its components. Such a scenario results in load shedding and cutting off production from renewable energy sources (RES) because shutting down (or starting up) a conventional power plant is considerably time consuming.

STLF, combined with RES forecasting, are integral to European Union's envisioned fully-integrated internal energy market and the proposed goal of 40% share of RES [5]. From a company's financial perspective, accurate forecasting is essential for its effective participation in the energy markets. It helps inform the bid, avoiding harsh fines for deviations and costs for balancing the supply-demand, thus maximizing profits. This is applicable for both energy/RES energy producers and suppliers, who are under obligation of physical delivery on the next day. Furthermore, the rapidly increasing penetration of RES in the energy mix will introduce considerable fluctuations in the electricity power and frequency, making accurate forecasting of energy production from RES, a necessity.

A common forecasting scheme is the hourly prediction of the electricity demand for the next full day. A certain offset period is inadvertently inserted between the last available historical values that are used as input and the first time step that is forecasted. The offset depends on the closing time of the bidding in the energy market and the delay introduced from the time of the measurement and until the data becomes available. Traditionally, suppliers have to go to the physical location of the measurement box every couple months to get the reading, though smart meters with remote monitoring capabilities can significantly reduce the delay to a few hours or minutes. Another offset that must be considered is introduced from external information providers such as weather forecasting agencies, a common type of exogenous features that is typically used in the energy domain and significantly improves the accuracy of forecasts.

Leaving aside non-linearities and fluctuations caused by sudden changes in weather patterns, another difficult point to forecast are the peaks and dips of a daily signal. Specifically, the daily electricity demand usually includes 2 peaks, one in mid-morning when everyone is working and another one in the afternoon when people are coming back home. Accurately predicting the timing and level of these peaks, which depends on multiple factors such as working habits, weather, and holidays, usually implies an overall increased forecasting performance.

State of the art forecasting models have become quite good at properly modelling interactions between weather conditions and energy demand. Nonetheless, they still struggle at determining when the energy demand peaks and dips and at what levels. These points of change can be more important than the time intervals in-between where the energy signal gradually increases or decreases, yet they are mostly overlooked. Another critical component of training a neural network is the required time to do so. Training time requirements can add up to days or even weeks, therefore it is desirable to keep the convergence speed as high as possible.

In this paper, we propose a new methodology, coined Time-Step Boosting, that optimizes the model's weights by estimating the forecasting error for each time step in the horizon. During subsequent training it uses it as loss weights so that the model will focus on improving the time steps where it struggles the most. When training multiple variations of a neural model during hyperparameter optimization only the error weights of the first iteration are necessary. We evaluated the proposed Time-Step boosting technique using hourly electricity demand data from European countries (ENTSOE data set) and from 6 states of the U.S. (ISONE data set). Similarly, it can be applied to water demand, retail sales, traffic flow, etc., that have a set horizon. Models were trained with/without Time-Step Boosting and compared in terms of convergence speed and accuracy, showcasing that they indeed benefit from the application of the technique.

The remainder of the paper is structured as follows: Section 2 provides a mathematical foundation for the forecasting task and briefly reviews different forecasting schemes and the use of errors as an input to the model in recent works. Section 3 describes Time-Step boosting, how it is calculated and for which forecasting schemes it can be applied. Section 4 defines the framework for the empirical evaluations, the data sets that were used, and presents the final results that showcase improvements in convergence speed and accuracy when Time-Step Boosting is applied. Finally, Section 5 summarizes the derived conclusions and outlines possible future directions.

2 Related Works

The forecasting task can be formulated as a function F that takes N+1 input sequences $\mathbf{x_i}$ and calculates a single output sequence $\mathbf{y}$ ($\mathbf{x_0}$ are historical values of $\mathbf{y}$, the rest $\mathbf{x_i}$ are exogenous features). For predetermined history window W, horizon H and taking possible offsets, p_w history offset and p_h future offset into account, the forecasting task is given by:

$$\mathbf{y} \in \mathbb{R}^{H,p_h} = [y_{T+p_h+1}, y_{T+p_h+2}, \ldots, y_{T+p_h+H}]$$
$$X = \mathbf{x_i} \in \mathbb{R}^{W,p_w} = [y^i_{T-p_w-W}, \ldots, y^i_{T-p_w}]$$
$$\hat{\mathbf{y}}_{T+p_h|H,p_w} = F(X)$$

where $\hat{\mathbf{y}}$ denotes the values of the forecasted time steps. In this context, function F is the neural network which can commonly be a Fully Connected network (MLP), a Convolutional Neural Network (CNN) or a Long Short-Term Memory network (LSTM), though usually a combination is used.

The choice of the history window, horizon and the offsets can guide the selection process of the forecasting scheme. The two typical strategies for multi-step-ahead forecasting are the Recursive and the Multi-Input Multi-Output (MIMO) strategies [6]. In the Recursive strategy, each predicted time step is fed as input for the next time step, until the full horizon is forecasted, thus allowing for forecasts of varying lengths. However, applying a Recursive strategy can complicate the implementation details when offsets are present. On the other hand, models that follow the MIMO strategy can integrate inter-dependencies between each time step in the horizon by producing forecasts for all the time steps at once. An additional advantage is avoiding compounding errors that can arise in the Recursive strategy. Therefore, unless a varying horizon is important for the forecasting task at hand, a MIMO strategy should be considered.

A mixture of forecasting schemes and methodologies can be found in recent literature for STLF. Load data from a distribution network in Cuba were used to train autoregressive integrated moving average (ARIMA) models to forecast the load of the next day in [10]. Daily and weekly seasonalities were removed by decomposing the time series and Particle Swarm Optimization was employed to select the best performing ARIMA model for each hour of the day, creating an ensemble of ARIMA models. White noise was added in [4] to evaluate its impact on ARIMA parameter estimation and their overall robustness, concluding that it remains stable for up to 20% noise to signal ratio.

Similarly, separate MLP models were trained for each hour of the day in [3], using a multi-input scheme with the load and temperature of the past 24 hours and of the same hour for the previous 4 weeks and 6 months, as well as the one hot encoding for season, weekends and holidays. Afterwards, the output of each model was concatenated to a full day and passed to a deep learning model employing residual connections with different skipping lengths. In [13], the authors considered a recursive strategy where a Radial Basis Function (RBF) network used the prediction error as feedback during the forecasting of the next step to increase the accuracy of the predictions. More recently, [2] proposed employing multi-rate input sampling of the input signal and recovering the original output sampling rate using hierarchical interpolation to forecast time series over long horizons. Stacked MLP blocks with residual connections were used for each sampling rate and it was suggested that the sampling rates either increase exponentially or match known seasonality cycles (daily, weekly, etc.).

Renowned for their performance in forecasting are Recurrent Neural Networks (RNNs) and CNNs. An in-depth overview of RNN variants (LSTM being one) and popular architectures was carried out by [1]. Following a notably extensive comparison against exponential smoothing and ARIMA, they concluded that RNNs are competitive alternatives for time series with homogeneous seasonal components. Additionally, CNNs have proved strong candidates for time series forecasting tasks as they are exceptional at extracting temporal patterns and correlations between multivariate time series. Combinations of RNNs and CNNs have also been proposed, such as [9] who used a CNN layer to extract short and long term patterns, an autoregressive component to adjust the signal's scale, and an RNN layer utilizing a Skip Connection to generate the prediction.

3 Time-Step Boosting

Neural networks forecasting electricity consumption routinely struggle with determining the timing and magnitude of peaks and dips. In the case of energy demand, this occurs around the 11 am and 5 pm peaks (shown in the next section). Even though losses such as the Mean Squared Error (MSE) negate this problem to a certain degree by enlarging the largest errors, they still struggle when forecasting slightly longer horizons because error peaks are smoothed out.

In this paper, we propose Time-Step Boosting, a technique that can be used when repeatedly training a model, for example during hyperparameter optimization, or training different models for similar data sets. As an example, European countries generally follow a similar style of living centered around an 8-9 hour working schedule, therefore the error patterns are expected to be similar.

The Time-Step Boosting technique works as follows. First, a model is trained on a forecasting task with a set horizon, in this case 24 hours. Next, the model is evaluated against the validation set and the forecasting error is calculated per time step. Afterwards, the model's loss is adjusted to use the time step errors as weights and multiply the loss per time step. The time step weights can be saved and used repeatedly for subsequent training of different models.

Consider N forecasts $\hat{Y}$ over a 24-hour horizon validation set (offsets are omitted to avoid cluttering the equations):

$$\hat{\mathbf{y}}_{T|24} = \begin{bmatrix} \hat{y}^1_{T+1} & \cdots & \hat{y}^1_{T+24} \\ \vdots & \ddots & \vdots \\ \hat{y}^N_{T+1} & \cdots & \hat{y}^N_{T+24} \end{bmatrix},$$

then, the average error per time step can be calculated as:

$$W^t_{time-step} = \{\frac{1}{N}\sum_{i=1}^{N}(y^i_t - \hat{y}^i_t)^2, t = 1, \ldots, 24\} \tag{1}$$

Afterwards, the Time-Step error is normalized to [a,1], where $a \geq 0$ is a small value close to 0. Finally, the Time-Step errors are used as weights for the respective time step and the MSE loss during training becomes:

$$MSE = \frac{1}{H}\sum_{t=1}^{H}(y_t - \hat{y}_t)^2 * W^t_{time-step} \tag{2}$$

Time-Step Boosting is loss-agnostic, meaning that it can be applied to all commonly used losses that include some sort of function averaging over the forecasted values, regardless of architecture. Though it is suggested that parameter a is selected to be greater than 0, selecting $a = 0$ will still work by association as the model will learn to forecast the other time steps of the time series.

4 Experimental Evaluation

Time-Step Boosting was evaluated on load data from two data sets; the European Network for Transmission System Operators for Electricity (ENTSOE) and the Independent System Operator New England (ISONE) [11,7]. The ENTSOE data set is complemented with weather data from NASA's Modern-Era Retrospective

Analysis for Research and Applications, Version 2 (MERRA-2)[12], which incorporates information regarding ice sheets in the North and South poles and the interactions with other physical processes in the climate system.

Forecasts are produced every day at 6 p.m. for the next full day using data of the previous 24 hours. Time-Step Boosting is tested on MLP, CNN and LSTM models. For each architecture configuration the respective model is trained and evaluated. Afterwards, the average error per time step (Equation 1) is calculated, the model's weights are reset to the initial pre-trained weights and the model is retrained using the time-step-weighted MSE loss (Equation 2). Both models are trained on the same data set using the same training regime and for the same number of epochs. Finally, a comparison of the models' improvement over time as well as their final performance evaluation on the test set is made.

4.1 Data Sets and Preprocessing

The ENTSOE data set includes hourly energy load data from 32 Transmission System Operators of European countries [11], who are required to publish information and data regarding the generation, load, transmission, and balancing. There are 43824 values spanning 5 years, from 2015 and until 2019, for each country. The MERRA-2 data set [12] comprises of hourly temperature, direct radiation energy and diffused radiation energy data for 19 European countries since 1980, though only the overlapping 2015-2019 period was used. The first semester of 2019 was reserved for the validation and the last for the testing.

The distribution of the electricity demand in Greece for 2015-2019 is displayed in Figure 1, where a few hourly, daily, and monthly patterns can be discerned with ease. On an hourly level, demand peaks late in the morning around 11 am when everyone is working, dips for a few hours and quickly increases again late in the afternoon around 5 pm. The highest variance, which is harder to forecast, occurs in the early afternoon. On a monthly level the peak electricity demand coincides with the warmest and coldest months. Sudden peaks in June, July and August are primarily correlated with heatwaves. On the other hand, the daily patterns indicate comparable electricity demand regardless of the day with a minor exception for the weekend days when demand is slightly reduced.

The ISONE data set [7] is comprised of the electricity demand, dry-bulb temperature and dew-point temperature for 6 U.S. states: Maine, New Hampshire, Vermont, Connecticut, Rhode Island, and Massachusetts. All variables are at an hourly resolution. Due to its significant size, Massachusetts is further broken down to Southeast, West/Central, and Northeast. The data set ranges from 2003 to 2017 and has a total of 131496 entries, however only the last 5 years were used in the experiments to cover the same period as the ENTSOE data set.

The daily mean electricity demand in 2015 is displayed in Figure 2, which follows similar patterns for all 6 states. The variance of the signal's amplitude between each state throughout the months can be explained by the geographical location, spatial weather conditions, the population and the general socioeconomic situation, at least to a certain degree.

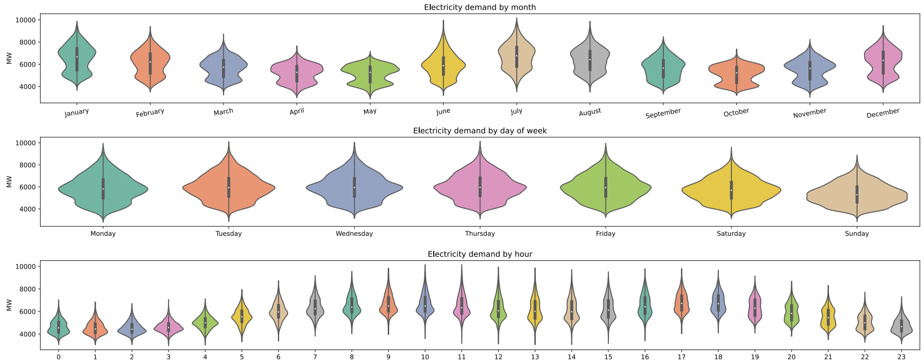

Fig. 1. Distribution of electricity demand in Greece, 2015-2019. Top: electricity demand by month. Middle: electricity demand by week. Bottom: electricity demand by hour

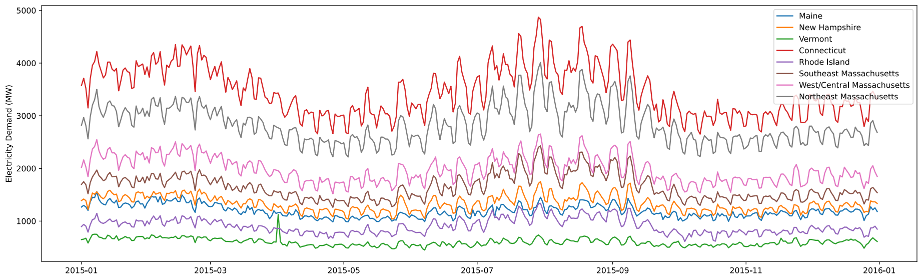

Fig. 2. Daily electricity demand for the New England region (USA) in 2015. States are annotated with different colors. Massachusetts is broken down to 3 areas.

An analysis was conducted to find missing values and determine the appropriate imputation approach that would preserve the periodicities and correctly impute peaks and dips, depending on the number of consecutive missing values. In ISONE there are only a couple missing values, therefore it is trivial to interpolate them using the adjacent hours without introducing significant noise. On the other hand, there are 44261 missing values in the ENTSOE energy data set across all countries, though Cyprus and Ukraine account for 41077 of them.

Countries that have only 1 and 2 missing values respectively, such as Germany and The Netherlands, can be filled in naively using adjacent values. Some countries like France and Greece have periods of more consecutive missing values, so imputation is done using the values of the previous/next day or week, for the same hour that is being filled in. Finally, in cases of even longer periods of consecutive missing values, for example Czech Republic which has 100 consecutive missing values in some cases, attempting to impute them would be extremely hard. In those cases, the time series is split in two parts so that the first part ends where the period of missing values starts, and the second part begins where the period of missing values ends. Further preprocessing is applied separately on each part, and they are concatenated before training.

4.2 Experiments and Results

The basic architecture design was a neural network layer that was repeated one or more times, with added Dropout layers in-between that had a small constant dropout rate of 0.1. The neural network layer was selected from an MLP, a CNN or an LSTM layer. A final Fully Connected (FC) layer with linear activation was used to produce the requested 24-hour forecast. For MLPs and CNNs, hidden layers ranged between 1 to 5 layers, while the hidden units were selected from [24, 100, 300]. Architectures with CNN layers included a GlobalMaxPooling layer before the final FC layer, to reduce the dimensionality of the computed data. Furthermore, kernel size was either 2 or 3, as having a longer kernel size would be impractical due to the 24-hour time step input. On the other hand, architectures with LSTM layers were limited up to 3 hidden layers and 100 hidden units, because more hidden layers or hidden units significantly increased the required training time without improving the model's performance noticeably.

A Stochastic Gradient Descent optimizer was used for calculating the gradients when training the model. The learning rate was configured between 0.1 and 0.000001, while the momentum was set either as the same value as the learning rate or at one tenth (1/10 * learning rate). The Nesterov accelerated gradient variation was chosen over the classical momentum as it often achieved a better convergence rate. Additionally, the choice of SGD over Adam, two of the most common optimizers, was made to take advantage of the greater generalization performance of the first compared to the later and the more stable training [8].

Two evaluation metrics were used in the experiments of this paper to report the results, Root Mean Squared Error (RMSE) and Mean Average Percentage Error (MAPE), both commonly used in time series forecasting tasks. The squared error in RMSE essentially highlights larger differences between forecasted and actual value and the root ensures it remains at the original scale. On the other hand, MAPE is a scale independent metric that allows for easy comparison across different data sets or time series with different scales.

The diagrams in Figure 3 present the gradual reduction of the MSE on the validation set during training on the ENTSOE data set for the MLP, LSTM and CNN models variations. The "weighted" variation is for the model where the Time-Step Boosting technique is applied. Comparing the two evaluation curves, one can observe that the weighted model converges faster than the normal one, quickly reaching a good approximation at around 25 epochs, while the normal model had to be trained for 100 epochs (4x) for the LSTM or 250+ epochs (10x) for the MLP and the CNN until it reached a similar performance.

Generally, at the end of the training scheme the weighted model variation outperformed the other one by a small margin most of the time. However, there were cases, such as the LSTM in the middle of Figure 3, where Time-Step Boosting helped the model escape the local minima and further improve its forecasting accuracy by up to 40%. This could be attributed to technique's capacity to focusing the model's attention on the correct source of forecasting error.

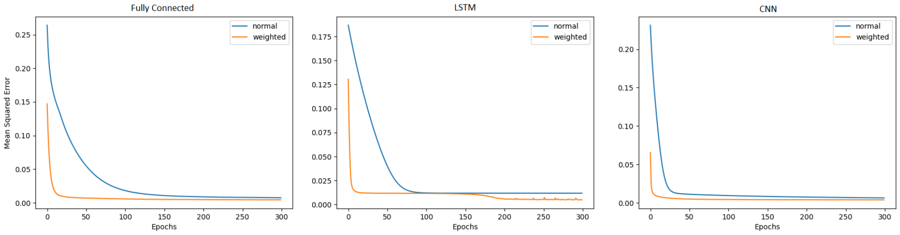

Fig. 3. Mean Squared Error evaluation during training on the validation set. FC: 5 layers, 300 units. LSTM: 5 layers, 100 units. CNN: 2 layers, 24 filters, kernel size 2.

From a hyperparameter search perspective, the best RMSE and MAPE performance scores for each model type are listed in Table 1 for both the normal and weighted variations. Generally, a learning rate of 0.01 was proven to be optimal for training the models, as a smaller learning rate often failed to converge. Overall, the best forecasting accuracy was achieved by a CNN model of 2 layers with 300 filters of kernel size 3. Nonetheless, satisfactory performance was also achieved with LSTM and MLP models.

Of interest to note, are the dissimilarities between normal and weighted models. MLPs and CNNs were slightly improved with the application of Time-Step Boosting, while LSTMs' accuracy almost doubled. Table 2 provides a performance report for the most accurate MLP, CNN, and LSTM models for each country. The accuracy varies between countries, though it seems that is especially true for countries with low electricity consumption on average, which results in the MAPE exploding. Overall, the application of Time-Step Boosting provided significant performance improvements.

The experiments were repeated on the ISONE data set. The hyperparameter search space remained the same as the first study with the same arrangement of MLP, CNN and LSTM neural networks. The findings are indeed similar as before

Table 1. Combinations with highest performing models on the ENTSOE data set. Performance for weighted models is reported on the right side of the slash "/".

			RMSE			MAPE	
Layers	Units	MLP	LSTM	CNN	MLP	LSTM	CNN
2	300	463/**419**	794/**441**	469/**418**	.062/**.057**	.1098/**.059**	.0635/**.056**
1	300	466/**418**	821/**457**	465/**416**	.062/**.056**	.1125/**.062**	.0628/**.055**
2	100	492/**430**	800/**465**	471/**420**	.065/**.058**	.1095/**.062**	.0635/**.056**
5	300	493/**435**	823/**472**	466/**418**	.065/**.059**	.1133/**.063**	.0630/**.056**
2	24	539/**446**	805/**489**	542/**429**	.071/**.060**	.1083/**.066**	.0730/**.057**

Table 2. Best performing models on the ENTSOE data set. Performance for the weighted models is reported on the right side of the slash ”/”.

Country	RMSE			MAPE		
	MLP	LSTM	CNN	MLP	LSTM	CNN
AT	823/**678**	1096/**985**	1055/**755**	.091/**.079**	.121/**.113**	.117/**.090**
BE	778/**653**	1031/**771**	1066/**721**	.065/**.055**	.088/**.067**	.092/**.062**
CH	652/**526**	1243/**641**	890/**636**	.082/**.065**	.155/**.080**	.115/**.082**
DE	6230/**4855**	7017/**6491**	7176/**5865**	.092/**.075**	.107/**.101**	.111/**.093**
DK	362/**304**	571/**423**	483/**407**	.076/**.066**	.118/**.092**	.103/**.090**
EE	83/**71**	169/**113**	155/**80**	.070/**.062**	.150/**.101**	.136/**.072**
ES	2459/**1923**	2776/**2622**	2949/**2271**	.072/**.056**	.081/**.079**	.085/**.069**
FI	532/**480**	1650/**698**	1595/**625**	.043/**.040**	.147/**.054**	.139/**.050**
FR	4683/**3971**	10528/**4799**	10444/**4195**	.071/**.062**	.168/**.076**	.169/**.066**
GR	473/**414**	713/**572**	805/**460**	.063/**.055**	.095/**.076**	.110/**.062**
HR	166/**142**	220/**207**	224/**164**	.065/**.057**	.085/**.081**	.086/**.064**
HU	399/**345**	501/**454**	537/**388**	.065/**.058**	.084/**.077**	.086/**.065**
IT	4438/**3816**	5085/**4960**	5213/**4169**	.111/**.095**	.130/**.127**	.133/**.107**
LV	68/**58**	103/**80**	97/**70**	.064/**.058**	.099/**.081**	.096/**.071**
NL	1276/**1024**	1774/**1490**	1695/**1277**	.083/**.067**	.116/**.101**	.109/**.082**
PL	1981/**1627**	2242/**2134**	2319/**1873**	.085/**.072**	.099/**.097**	.101/**.086**
PT	537/**446**	652/**604**	678/**521**	.078/**.066**	.093/**.090**	.094/**.078**
SI	163/**138**	193/**178**	200/**158**	.094/**.087**	.114/**.109**	.118/**.096**
SK	247/**198**	335/**264**	346/**225**	.060/**.048**	.082/**.068**	.084/**.057**

and confirm the superiority of the Time-Step Boosting technique. Cases with already low evaluation score, such as Vermont (VT), see negligible improvements, as the performance improvement is greater when the evaluation error is initially higher. Training the weighted models was also faster for most combinations.

When comparing performances between the ENTSOE and the ISONE data sets, it becomes apparent that all models that were trained on ISONE achieve worse accuracy. We consider two explanations as most probable. The first is that the dry and wet bulb temperature features are lacking, perhaps due to the size of the areas or maybe because the data collection procedure was subpar. The other possible explanation is that the direct radiation feature that was used in the previous experiments actually carries a lot of important information. Unfortunately, it is not available in the ISONE data set.

Table 3. Forecasting accuracy using the model proposed by [3]. Performance for the weighted models is reported on the right side of the slash ”/”.

RMSE	MAPE	Units
848 / **481**	.121 / **.066**	300
832 / **499**	.117 / **.068**	100
860 / **514**	.122 / **.069**	10

Table 4. Best performing hyperparameters on the ISONE data set. Performance for the weighted models is reported on the right side of the slash ”/”.

Area Code	Model	RMSE	MAPE	No Layers	No Units
ISONE	MLP	1303 / **1058**	.079 / **.064**	5	300
	CNN	1818 / **1649**	.117 / **.104**	5	300
	LSTM	2119 / **1860**	.124 / **.111**	2	100
CT	MLP	327 / **256**	.077 / **.061**	5	300
	CNN	469 / **412**	.120 / **.104**	1	100
	LSTM	580 / **473**	.135 / **.124**	2	24
ME	MLP	145 / **139**	.121 / **.118**	2	300
	CNN	194 / **178**	.167 / **.154**	2	300
	LSTM	379 / **365**	.154 / **.146**	1	300
NH	MLP	137 / **131**	.093 / **.088**	2	300
	CNN	207 / **191**	.148 / **.132**	1	300
	LSTM	244 / **206**	.169 / **.146**	2	24
VT	MLP	67 / **66**	.107 / **.108**	2	300
	CNN	115 / **87**	.188 / **.142**	1	24
	LSTM	116 / **100**	.186 / **.170**	2	24
SEMA	MLP	150 / **117**	.076 / **.058**	5	300
	CNN	218 / **197**	.114 / **.103**	1	300
	LSTM	274 / **214**	.140 / **.114**	2	24
WCMA	MLP	302 / **297**	.119 / **.117**	2	300
	CNN	359 / **351**	.143 / **.140**	2	100
	LSTM	365 / **354**	.145 / **.142**	2	100

The proposed technique was further validated with the model proposed by [3]. The original configuration was used with 10 units for the layers that receive the load and temperature values and 5 units for the season and weekend encodings. Additionally, variations with 100/10 and 300/20 units were also trained because in the previous experiment it was observed that MLPs performed better with wider layers. The results shown in Table 3, indicate that increasing the number of units per layer can indeed provide a small forecasting accuracy increase. Furthermore, it is noted that the use of the Time-Step Boosting technique (reported on the right side of the slash ”/” for each metric) had a significant positive impact on the models' performance.

5 Conclusions and Future Work

In this paper, we introduced Time-Step Boosting, a loss-agnostic technique for time series forecasting that can be applied to any neural network architecture. It adjusts the loss by taking into account the average loss per forecasted time step, thus focusing on the parts of the time series that are the most challenging to forecast. Experimental results showed that it significantly reduces the time needed for the model to converge and it can often help escape local minima. The technique was validated with multiple MLPs, CNNs and LSTMs of varying width and depth on energy load data of 19 European countries and 6 U.S. states, and additionally using a state-of-the-art model.

Overall, the experiments showed the Time-Step Boosting technique to be robust and offer considerable increases both in convergence speed and forecasting accuracy. Employing it during optimization or when training a single model on multiple data sets could be extremely helpful as it would greatly reduce the necessary time for exploring the hyperparameter space. It would also be interesting to see how important a fixed horizon is and to further validate the technique on different timeseries that exhibit non-uniform forecasting errors.

References

1. Benidis, K., Rangapuram, S.S., Flunkert, V., Wang, B., Maddix, D., Turkmen, C., Gasthaus, J., Bohlke-Schneider, M., Salinas, D., Stella, L., Callot, L., Januschowski, T.: Neural forecasting: Introduction and literature overview. arXiv (2020). https://doi.org/10.48550/arXiv.2004.10240
2. Challu, C., Olivares, K.G., Oreshkin, B.N., Garza, F., Mergenthaler, M., Dubrawski, A.: N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting (2022). https://doi.org/10.48550/arXiv.2201.12886
3. Chen, K., Chen, K., Wang, Q., He, Z., Hu, J., He, J.: Short-Term Load Forecasting With Deep Residual Networks. IEEE Transactions on Smart Grid **10**(4), 3943–3952 (2019). https://doi.org/10.1109/TSG.2018.2844307
4. Chodakowska, E., Nazarko, J., Nazarko, Ł.: ARIMA Models in Electrical Load Forecasting and Their Robustness to Noise. Energies **14**(23), 7952 (2021). https://doi.org/10.3390/en14237952
5. European Commission: Questions and answers - making our energy system fit for our climate targets. (14 July 2021), https://ec.europa.eu/commission/presscorner/detail/en/qanda_21_3544, accessed 7 February 2021
6. Hewamalage, H., Bergmeir, C., Bandara, K.: Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. International Journal of Forecasting **37**(1), 388–427 (2021). https://doi.org/10.1016/j.ijforecast.2020.06.008
7. Hong, T., Xie, J., Black, J.: Global energy forecasting competition 2017: Hierarchical probabilistic load forecasting. International Journal of Forecasting **35**(4), 1389–1399 (2019). https://doi.org/10.1016/j.ijforecast.2019.02.006
8. Keskar, N.S., Socher, R.: Improving Generalization Performance by Switching from Adam to SGD. arXiv (dec 2017)
9. Lai, G., Chang, W.C., Yang, Y., Liu, H.: Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 95–104. ACM, New York, NY, USA (jun 2018). https://doi.org/10.1145/3209978.3210006
10. Marrero, L., Garcia-Santander, L., Carrizo, D., Ulloa, F.: An Application of Load Forecasting Based on ARIMA Models and Particle Swarm Optimization. In: 2019 11th International Symposium on Advanced Topics in Electrical Engineering (ATEE). pp. 1–6. IEEE, Bucharest, Romania (mar 2019). https://doi.org/10.1109/ATEE.2019.8724891
11. Open Power System Data: Data Package Time series. Version 2020-10-06 (2020). https://doi.org/10.25832/time_series/2020-10-06

12. Open Power System Data: Data Package Weather Data. Version 2020-09-16 (2020). https://doi.org/10.25832/weather_data/2020-09-16
13. Zemouri, R., Patic, P.C.: Prediction error feedback for time series prediction: A way to improve the accuracy of predictions. In: Proceedings of the 4th Conference on European Computing Conference. p. 58–62. ECC'10, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA (2010)