Acta Cybernetica **27** (2025) 221–239.

Multi Model Recursion for Hungarian Electricity Load Forecasting

Mátyás Sebők[ab]

Abstract

Time series analysis and prediction is a difficult and complex problem. Many Machine and Deep Learning methods exist with better and better results. This paper proposes a strategy called Multi Model Recursion. It uses separate Deep Learning models per feature that needs predicting. Another improvement is not predicting features which are easily calculated. Having extra models per feature helps in "simulating" a future environment since it predicts external variables otherwise unknown. The Multi Model Recursion developed is an improvement of the commonly used Recursive strategy. The paper compares this method with models and strategies frequently used in the field. The testing dataset is put together from publicly available Hungarian electricity load and weather data. The task was to predict the country's net electricity load for the next 3 hours.

Keywords: time series, deep learning, Multi Model Recursion, electricity load forecasting

1 Introduction & Related works

Short term electricity load forecasting is useful since the forecasting models can adapt better to the given situation and give more accurate predictions. The better predictions give the opportunity for participants to better exploit their resources and minimize their costs. A 3-hour forecast comparison of Hungary's net electricity load shows the different strengths of models at single step forecasting and also describes their longer range performance.

The difficulty is that while weather data is available at a large resolution, forecasts are not always available the same way. The focus of Multi Model Recursion is to create a simulated environment with the given exogenous variables and their respective models to further enhance the predictions of the target variable. Compared to the regular Recursive strategy, this architecture can optimize better since it doesn't have to directly take into account the exogenous and time-series variables when calculating the cost function.

[a]Eötvös Loránd University, Budapest, Hungary
[b]E-mail: sebokmatyas01@gmail.com, ORCID: 0009-0000-0725-4212

DOI: 10.14232/actacyb.311880

This paper primarily compares Multi Model Recursion with the regular Recursive strategy using recurrent deep-learning algorithms. It also compares it with the Multi Input Multi Output (*MIMO*) strategy using Convolutional, Temporal Convolutional and LSTM networks. Compares it with the very powerful Sequence to Sequence (*Seq2Seq*) strategy, which uses an encoder-decoder architecture. To justify the usage of such complex algorithms it also looks at the performance of a machine-learning algorithm known as Random Forests and looks at the advantages compared to a Statistical method known as Seasonal Autoregressive Integrated Moving Average (*SARIMA*). The comparison happens based on a dataset created from public data from OMSZ (*Országos Meterológiai Szolgálat*) for weather data and data from MAVIR (*Magyar Villamosenergia-ipari Átviteli Rendszerirányító Zrt.*) for electricity load data.

1.1 Electricity load forecasting

Nti et al. provides a review on electricity load forecasting [11]. The authors provide a comprehensive study on the used forecasting methods and evaluation metrics. This motivates the use of MAE, RMSE, MPE and MAPE metrics and the evaluation of ANNs as these are the most used algorithms in the field.

Azeem et al. explains the application of electricity load forecasting techniques including short term electricity load forecasting [1]. The forecast horizon of a couple of hours can be critical in the operation and financial decision-making of energy management systems. Such forecasts can be used to decide which resources to utilize, for e.g. gas, coal, solar or wind. Another decision may be to import electricity at a lower cost than the utilization of non-renewable resources. The authors also explain the optimization techniques where the forecasts are used.

While the paper focuses on national load forecasting all methods can be utilized at a lower resolution such as Virtual Power Plants. Ghavidel et al. explain that such VPPs aggregate many physical entities such as renewable and non-renewable power plants, batteries and pump storage [5]. Accurate forecasts help the operation of such VPPs.

Yazici et al. provide a case study for electricity load prediction for Istanbul [14]. The authors achieve an impressive 1% MAPE metric for one-hour-ahead predictions and 2.2% for 24-hour-ahead predictions. While not matching this paper's 3-hour-ahead forecast horizon they provide a baseline to verify the results of this paper.

1.2 Problem description

Gasparin et al. [4] describes the task of time series forecasting for the electricity load case where there is a given uniform resolution $s = [s[0], s[1]..., s[T]]|s \subset \mathbb{R}$ time series data vector. It is ordered by time and has an hourly resolution in this case. For machine and deep learning purposes it is helpful to work with equal n_T length time windows. The prediction window's length is specified as n_O. The paper explores supervised learning solutions which require input-output pairs. Let's specify at time step t the input vector as $x_t = [s[t - n_T + 1], ..., s[t]]$ and the output

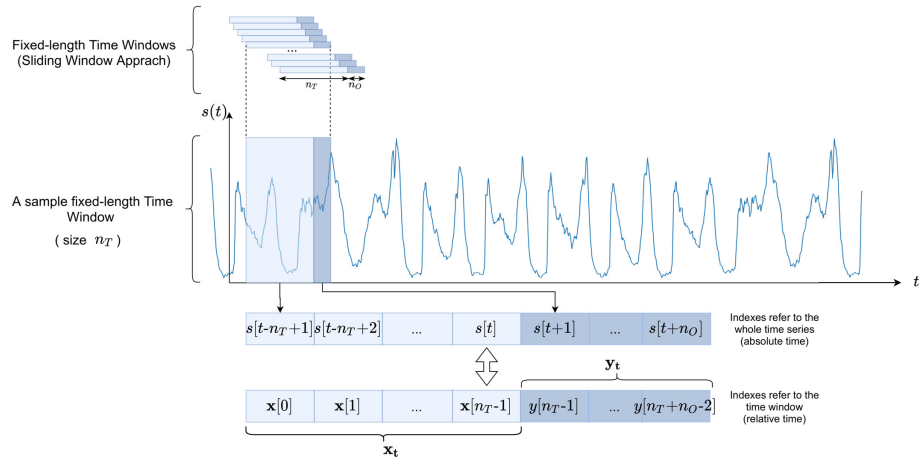


Figure 1: The sliding window approach [4]

vector as $y_t = [s[t+1], ..., [t+n_O]]$. This concludes in a sliding window type approach shown in Figure 1.

A model can be described as a parametric function f and its parameter vector θ, approximated by $\hat{\theta}$. With the above notation at time step t the model's output is $\hat{y_t} = f(x_t, \hat{\theta})$ which is an approximation of $y_t \in \mathbb{R}^{n_O}$. It is important to mention that in the practical application of such approaches the model is split into Machine or Deep Learning models and forecasting strategies where a strategy describes the steps of forecasting. These are discussed separately.

In the extension of the problem description $s \subset \mathbb{R}^d$ where $d-1$ is the number of exogenous or external features. $x_t[i][k]$ notates the kth feature of the ith element in the sequence where $i \in [0..n_T)$ and $k \in [0..d)$. In layman's terms this means that the forecast is helped by including additional features such as the weather, time of day or year.

1.3 Models

This section discusses the commonly used neural network and machine learning architectures for time series forecasting. These models are used in conjunction with the following forecasting strategies to provide predictions. The models are used later for the new Multi-Model Recursion strategy.

The Random Forest model is a classical machine learning method that uses the splitting rule for optimization. Probst et al. explains the optimization and training of such models [12]. This model may be used in any but the Sequence-to-Sequence strategy.

Gu et al. discusses the advancements made in convolutional neural network development [7]. By applying the architecture to 1D data like a time series the model can learn patterns in time that impact the prediction of the next time step.

[4] show the usage of causal convolution which applies the padding from only 1 side. This can further be expanded with dilated causal convolution ensuring a large receptive field for each layer.

Bai et al. shows the architecture and advantages of a temporal convolutional network [2]. This architecture, shown in Figure 2, employs dilated causal convolutions in addition to residual connections. Applying residual connections is beneficial to combat the vanishing gradient problem where the gradient gets progressively smaller as the optimization reaches the early layers. Through residual connections the gradient doesn't get affected by the weights ensuring a more stable descent.

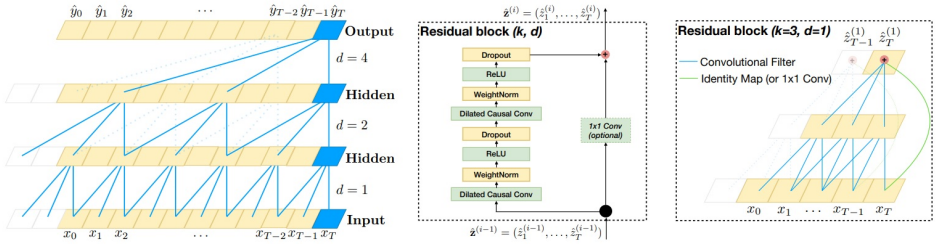


Figure 2: Temporal Convolutional Network Architecture: dilated convolutional layers, residual block and example for a residual block [2]

Masum et al. describes the LSTM model's architecture and its application to time series forecasting [10]. The advantage of RNNs (*Recurrent Neural Network*) is that they can process inputs of different lengths. This architecture can be applied to the Sequence-to-Sequence strategy that is mentioned and evaluated later in this paper. Shen et al. describes the workings of GRU based networks [13]. It is a newer approach compared to the LSTM aiming to resolve the same problem. It is generally not obvious as to which will perform better for a given task out of LSTM and GRU based networks. This is the reason both are evaluated in this paper.

1.4 Forecasting Strategies

Forecasting strategies describe how AI models are used for time series prediction. Strategies have to describe how many models are used and what the output dimensions are. It is possible to have strategies that complete predictions in a single or multiple steps. The following sections describe the forecasting strategies used and compared against Multi Model Recursion.

1.4.1 Multi-Input Multi-Output

Taieb et al. finds that multi-output strategies have good performance on time series forecasting tasks [3]. MIMO (*Multi-Input Multi-Output*) uses the entire input in one step at time t to produce the entire output vector y_t. The strategy can simply be described by the equation below if the forecasting model is f.

$$\hat{y}_t = f(x_t), \text{ simple multi-output prediction} \tag{1}$$

1.4.2 Sequence-to-Sequence

Seq2Seq (*Sequence-to-Sequence*) architectures were designed originally because it can be difficult to provide arbitrary length outputs with RNNs. The encoder-decoder architecture this strategy follows is the basis of modern LLMs (although the SoTA models are decoder only at the time of writing).

It consists of 2 models of the same type of RNNs, usually LSTMs or GRUs. The encoder produces a hidden state (and a cell state in the case of an LSTM). The decoder then uses this hidden state and its own outputs to produce the output. This can go until a certain stop sequence or iteration count. Zaki et al. [9] uses an LSTM based Seq2Seq model for household electricity load prediction but in this paper more success was found using a GRU based approach. Algorithm 1 describes the strategy.

Algorithm 1 Seq2seq strategy

$h_t := f_{enc}(x_t, h_0)$ *only need the hidden state from the encoder*
$\hat{y}_t[-1] := SELECT(x_t[n_T])$ *value that corresponds to the target feature*
for $i = 0, \ldots, n_O - 1$ **do**
  $\hat{y}_t[i] := f_{dec}(\hat{y}_t, h_t)$ *y_t is extended step-by-step*
  if $random(0 \ldots 1) < teacher\_forcing$ **then**
    $\hat{y}_t[i] := y_t[i]$ *teacher forcing, only while training*
  end if
end for
return $\hat{y}_t[0 \ldots]$

Teacher forcing for training Seq2Seq architectures helps with generalization over longer sequences. Since previous predictions affect the new ones they are substituted at a random probability with the real values. The probability gets decreased as training goes on. This technique helps the model train for longer forecast horizons as the compounding effect of incorrect predictions is removed. If $\hat{y}_t[i]$ is swapped for $y_t[i]$ then when optimizing based on the ith prediction step $\hat{y}_t[i]$ is used since $y_t[i]$ would provide a gradient of zero even if the prediction is incorrect. This approach is also applicable to the Recursive and Multi Model Recursive strategy.

1.4.3 Recursive

Taieb et al. explains that the recursive strategy uses a single model that is trained for forecasting only 1 step [3]. Here the model forecasts all external features for the next time step. This approach is interesting as the model isn't necessarily trained for multi step forecasting, but with the strategy it is applicable as such. Algorithm 2 describes the strategy.

Algorithm 2 Recursive strategy

1: **for** $i = 0, \ldots, n_O - 1$ **do**
2: $\hat{y}_t[i] := f(x_t)$ *forecast 1 step*
3: $x_t := x_t[1..n_T)$ *removing the first element of the input window*
4: $x_t[n_T] := \hat{y}_t[i]$ *extend the window by 1 at the end*
5: **end for**
6: **return** $\hat{y}_t$ *this includes the external feature forecasts*

For each t time step the forecast goes for step $t + 1$. This is then viewed as the "truth" and the model forecasts step $t+2$. Iterating this approach gives the output vector. The disadvantage of this method is that external feature forecasting requires larger weight matrices increasing processing requirements. Usually the optimization is also not efficient since the model is optimized for features not relevant for an application.

2 Methodology

2.1 Multi-Model Recursion

This paper presents the MMRec (*Multi-Model Recursion*) strategy which is an enhancement of the previously mentioned Recursive strategy. It aims to keep the advantages of the Recursive approach such as the single step prediction which generally gives more accurate predictions at that step. Being able to predict any length regardless of the training specifications is also an advantage, although performance may not be desirable if the training and inference output lengths are different. It incorporates the advantage of the MIMO and Seq2Seq approaches which only predict and optimize for the target variable.

There are 4 major changes from the Recursive method. The first one is regarding loss calculation for the network. Instead of training for all features directly the loss is calculated at the target variable at each step. This way the optimization focuses on electricity load in this case. The backpropagation will make sure that external features aren't left out. This is especially important in the next steps where multiple models are introduced.

Following the example of Seq2Seq teacher forcing is also applicable for the strategy. The same principles apply with the only difference being that each step's output vector is larger than 1. The random probability shown in Seq2Seq is calculated per member instead of once for the vector. It also decreases over the training period just like the mentioned strategy.

The next difference is the calculation of external features that are simple to predict. Features like time, or the lag of electricity load are easy to calculate via equations. These features are either pre-calculated or implemented into the strategy and used as is. This way no processing power is wasted on features that we know the exact values of even for the future.

Algorithm 3 Multi-Model Recursion

for $i = 0 \dots n_O - 1$ **do**
  for $j = 0 \dots m - 1$ **do**
    $\hat{y}_t[i,j] := f_j(x_t)$ *forecast given feature*
    $x_t := x_t[1..n_T)$ *sliding the window*
    $x_t[n_T, j] := \hat{y}_t[i,j]$ *substitute in the forecasted features*
    if $random(0 \dots 1) < teacher\_forcing$ **then**
      $x_t[n_T, j] := y_t[i,j]$ *teacher-forcing*
    end if
  end for
  $x_t[n_T] := g(x_t)$ *g calculates the obvious variables*
end for
return $\hat{y}_t$

The last change is the Multi-Model part of MMRec. Each feature that isn't calculated with the previous change gets its own neural network. Each model forecasts 1 specific feature making the individual models smaller. It is also possible to vary the architectures of them. Algorithm 3 and Figure 3 describe the strategy where m is the number of features forecasted by Neural Networks.

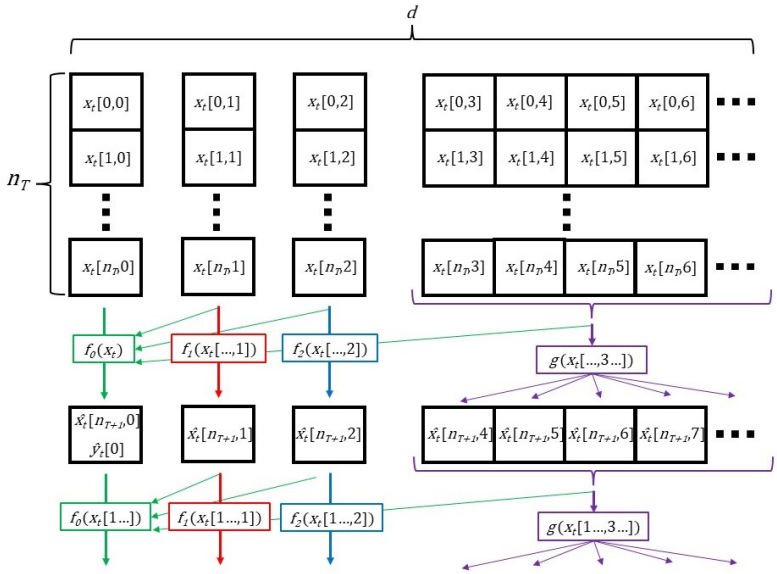
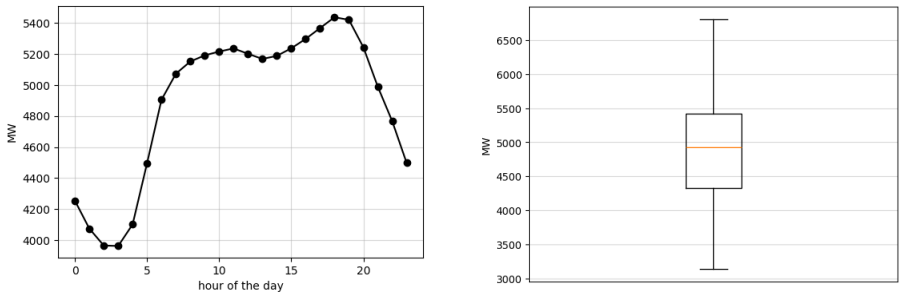


Figure 3: Multi-Model Recursion diagram, f_k are the different models, g calculates the obvious features, note that teacher forcing is not indicated here

2.2 Data

The dataset used for evaluating Multi Model Recursion against the mentioned methods consists of weather data downloaded from OMSZ's data publication [1] and electricity/system load data downloaded from MAVIR's dataset [2]. The observed time is from 01/01/2015 until 31/08/2023 resulting in an approximately 9 year long dataset describing Hungary. The timeframe was chosen due to OMSZ establishing many new weather stations in the year 2014.



(a) Hourly grouping (b) Box plot diagram

Figure 4: Net electricity load graphs

The source for the electricity load part of the dataset contains many fields from which "net electricity load (MW)" was chosen as the target and only feature describing electricity load. This is due to another feature existing in the original dataset named "MAVIR forecast" predicting net electricity load at the time step. This gives a baseline to justify the usage of machine and deep learning methods. Figure 4 (a) shows the hourly grouping of net electricity load for the observed timeframe. These type of graphs vary heavily by country. The box plot diagram in 4 (b) shows the high standard deviation of the dataset making forecasting tasks difficult.

OMSZ's weather stations all measure many different weather features like precipitation, temperature, relative humidity, global radiation and wind speed just to name a few. These are measured at over 100 stations giving a resolution that is too large for country scale predictions. So every weather feature was averaged over all stations resulting in 1 feature for each that describes the entire country. For example instead of 100+ temperature measurements there is only 1 describing Hungary.

Intuitively, many of these weather features don't make a difference for electricity load forecasting. By applying automatic sequential feature selection using Random Forests precipitation and global radiation proved to be the most descriptive. The feature selection algorithm was applied to time describing features at the same

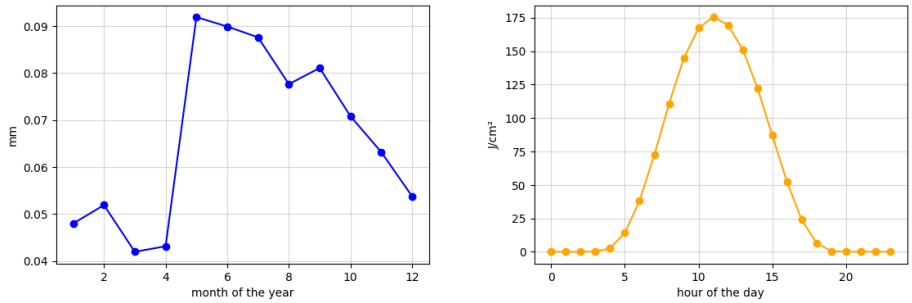
[1] https://odp.met.hu/
[2] https://www.mavir.hu/web/mavir/rendszerterheles

time. This means the process not only gave the useful weather features but time features as well. The best performance was observed at 11 features (the algorithm ran until 13 but after 11 the performance decreased). The chosen features are:

- electricity load and its 24-hour lag
- precipitation and global radiation
- holiday and weekend indicators
- hour, day of the week, day of the year, month, year

The feature selection chose global radiation over temperature for the best results. This is likely due to the fact that global radiation refers to the solar radiation that falls on a horizontal surface. This is supported by a correlation factor of 0.55 when observing temperature and global radiation.

The main point of reducing feature count in this way is that Multi Model Recursion is best applied in cases where only a couple external features are present since they all require separate models. Graphs for precipitation and global radiation from the dataset can be found in Figure 5.



(a) Precipitation monthly grouping (b) Global radiation hourly grouping

Figure 5: Weather feature graphs

The chosen features were re-evaluated at a later stage while training the MIMO LSTM approach and the 11 features chosen performed better. At this stage it's important to mention the choice of $n_T = 24$ for most strategy model pairs. This was made after choosing features and testing 12-, 24-, 36- and 48-hour lookbacks where 24 performed the best. This was re-evaluated for certain models, changes are mentioned where they were made.

2.3 Training and Evaluation method

When evaluating strategy model pairs it is important to find close to the best hyperparameter configurations. In this paper this is done using the Grid Search

algorithm where each hyperparameter gets a specified set of values. All combinations are then tried and the one with the best metrics and/or loss is chosen. Here RMSE was used since its strong reaction to outliers provides a clear picture of performance. RMSE is always calculated on the test set.

For machine and deep learning approaches it's usually important to use some form of cross validation technique. This means that multiple training loops are run using different parts of the data. Time series forecasting differs from other tasks since it wouldn't make sense to use future data in training while predicting the past. Due to this when evaluating in this paper, time series k-fold cross validation (Figure 6) is used. While finding hyperparameters the splits were limited to $k = 6$ (to save computational resources) and for final evaluation it was limited to $k = 9$. The validation set is always separated from the training set, taking up 1/8th of a single fold. For example if observing the 1st fold in Figure 6, the validation set is the last 1/8th of the training set. Furthermore, it is the same length for all other splits but always taken from the end of the training set.

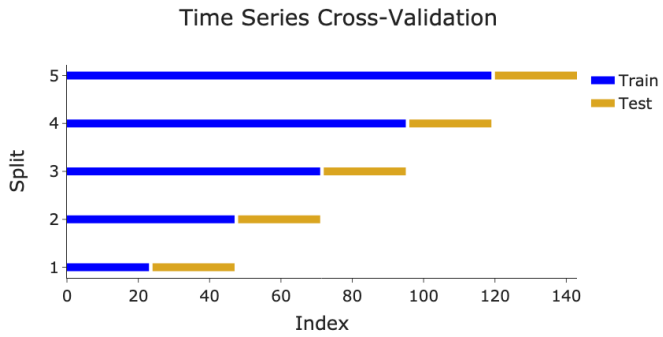


Figure 6: Time series cross validation [8]

After the hyperparameters are found each strategy model pair is trained and evaluated 6 times. These are done with the same hyperparameters but since the weights were initialized randomly the results vary. This is taken into account and the standard deviation of results through each fold and training cycle is displayed in the final table. Observed metrics are the following: MAE (Mean Absolute Error), MSE (Mean Squared Error), RMSE (Root Mean Squared Error), MAPE (Mean Absolute Percentage Error), and MPE (Mean Percentage Error).

Some technical details for the Grid Search algorithm are listed here. For each strategy pair the size and number of layers were searched for. For convolutional networks different lookback lengths were also observed such as $n_t = 48$. Learning rates, batch sizes and dropout ratios were also searched for. For most approaches not all combinations were observed at once but 2–3 hyperparameters were searched for once. This was iterated until a satisfying result was reached. After reaching a good point small changes were tested and if they didn't yield better results the parameters were chosen.

2.4 Chosen strategy model pairs

This section lists the chosen approaches for evaluation to compare with Multi Model Recursion. Other than SARIMA all of them are listed as strategy – model with abbreviations that are present in the final evaluation.

SARIMA (*Seasonal Autoregressive Integrated Moving Average*) This statistical method proved to be ineffective for this dataset since its parameters optimized at 1 time step didn't mean it was good for other time steps. At any one point the forecasts were comparable to MAVIR's predictions but it required a new parameter search to be effective. Due to this it isn't listed in the final evaluation. Parameter search may take over an hour which is not acceptable for this application where the neural network based models show no degradation of performance up to a year after training.

MIMO – RF (*Random Forest*) A classical machine learning approach to compare neural networks with. Decent performance on certain folds but heavily degrades at others.

MIMO – CNN (*Convolutional Neural Network*) A 1D Causal Convolution approach which didn't perform well on the dataset.

MIMO – TCN (*Temporal Convolutional Neural Network*) A 1D Temporal Convolutional approach performing well on short term forecasting.

MIMO – LSTM (*Long Short-Term Memory*) An approach that used to be one of the most popular since Recurrent networks work quite well for short sequence understanding.

Seq2Seq – GRU (*Gated Recurrent Unit*) A newer approach that may be viewed as the ancestor to SoTA transformer models.

Recursive – GRU The simplest forecasting strategy used with a GRU model. LSTM was also tested but GRUs proved to be more effective.

MMRec – 1 layer GRU Uses CNN and TCN for external features. Larger hidden state than the following approach.

MMRec – 2 layer GRU Uses CNN and TCN for external features. Smaller hidden state, when searching for hyperparameters its performance was very close to the previous one.

MMRec – FULL GRU Uses GRUs for external features too. A comparatively large and slow model to view what kind of performance MMRec can reach on the dataset.

The models observe a 24 horizon and forecast 3 hours ahead. The only exception to this are the MIMO - CNN and MIMO - TCN methods as these benefit from observing a 48-hour horizon. The other methods were also tested with shorter and longer horizons but the 24 hour performed best.

2.5 Training details of MMRec

To make the hyperparameter search faster each model for external features were first tuned as a Recursive strategy model for the given feature. The external feature models tried for precipitation and global radiation were CNN, TCN, GRU and LSTM networks. For global radiation CNNs performed better than TCNs and for precipitation the opposite was true. When GRUs were tested the addition of external features in relation to precipitation or global radiation slightly outperformed the Convolutional counterparts.

During the hyperparameter search 2 configurations proved to be powerful in forecasting. One which used a GRU with 1 layer but a larger hidden state and one with 2 layers using a smaller hidden state. Due to this both configurations are part of the final evaluation in addition to the FULL GRU approach.

3 Results

The final evaluation of strategy model pairs listed in Section 2.4 happened according to Section 2.3. Each pair is trained starting with their respective found "best" hyperparameters 6 times for 9 splits each. An interesting observation made during this is that most approaches using LSTMs performed noticeably worse in the first 2 splits. This was lessened by GRUs but the first splits were generally worse than anything else. This is likely due to the amount of data required by these models when training. Due to this the final evaluation lists the best performance of strategy model pairs while excluding the first or first and second splits, whichever gives better results.

Table 1 shows the final results for each of the mentioned metrics. Each metric also has the standard deviation listed over training iterations and the 9 folds per iteration. This gives more insight into certain models that may perform differently over the given folds such as CNNs and the Random Forest approach. MAVIR predictions have a standard deviation of 0 since it is impossible to observe training iterations or folds because it is given as is by MAVIR. MAVIR's predictions provide a higher resolution than 1 hour and are made with a 12-hour forecast horizon. This heavily affects accuracy at 3 hour predictions as forecasts for longer horizons generalize more.

From the table it's possible to observe that in terms of the MIMO strategy LSTMs perform the best closely followed by TCNs. TCNs struggle more in the

later steps of forecasting (steps 2 and 3 in this comparison). The Seq2Seq approach proved to be the most performant for this dataset in terms of metrics but also in approach size. The Recursive method performs quite poorly in comparison to others due to the shortcomings mentioned in Section 1.4.3. The presented MMRec strategy is clearly better than the Recursive strategy it is based on. With the additional ideas taken from Seq2Seq such as teacher forcing its performance is comparable with the MIMO LSTM approach and starts closing the gap on the Seq2Seq strategy as well.

The scores in Table 1 show a MAPE score for the presented approaches of about $1.2 - 2.1\%$ which is in line with the case study mentioned in the Introduction done by Yazici et al. [14]. Although the authors use a different dataset, time horizons and find that 1D CNNs perform best the performance of the forecasts are comparable. This shows that the results in this paper compare to an existing real world study.

Table 1: Table of evaluation results

Strategy Model	**MAE** (MW)	**RMSE** (MW)	**MAPE** $(\%)$	**MPE** $(\%)$
MAVIR prediction	252.58 ± 0	300.81 ± 0	4.97 ± 0	-4.70 ± 0
MIMO RF	69.96 ± 15.29	104.32 ± 24.0	1.42 ± 0.32	0.017 ± 0.25
MIMO CNN	103.13 ± 21.5	146.69 ± 27.55	2.12 ± 0.46	0.196 ± 0.416
MIMO TCN	63.92 ± 10.46	92.96 ± 15.57	1.31 ± 0.22	-0.001 ± 0.198
MIMO LSTM	62.62 ± 6.05	88.97 ± 9.19	1.28 ± 0.13	0.05 ± 0.175
Seq2seq GRU	$\mathbf{58.75 \pm 6.22}$	$\mathbf{84.21 \pm 9.83}$	$\mathbf{1.21 \pm 0.13}$	$\mathbf{0.08 \pm 0.168}$
Recursive GRU	94.41 ± 14.41	128.39 ± 17.39	1.94 ± 0.28	0.119 ± 0.744
MMRec GRU 1L	65.4 ± 7.68	92.43 ± 10.88	1.34 ± 0.17	0.114 ± 0.292
MMRec GRU 2L	64.79 ± 7.31	90.85 ± 10.27	1.32 ± 0.16	-0.029 ± 0.295
MMRec FULL GRU	62.17 ± 7.95	88.42 ± 11.25	1.27 ± 0.17	0.098 ± 0.261

Table 2 shows how much time is required for training and predicting with the models. Average and standard deviation can be understood the same way as for Table 1 described above. The Recursive GRU strategy training times and the MMRec GRU 1L/2L ones are similar. Even though multiple models are used for MMRec, they are much smaller and thus train faster than a large GRU for the Recursive strategy. Prediction times are the worst for MMRec as it uses multiple

Table 2: Table of training and prediction times

Strategy Model	**Training** (minutes)	**Prediction** *circa 7500 entries* (seconds)
MIMO RF	2.48 ± 1.45	0.082 ± 0.026
MIMO CNN	1.99 ± 1.09	0.136 ± 0.011
MIMO TCN	2.82 ± 1.51	0.258 ± 0.045
MIMO LSTM	2.99 ± 1.81	0.153 ± 0.015
Seq2seq GRU	6.16 ± 3.66	0.25 ± 0.019
Recursive GRU	7.16 ± 3.75	0.731 ± 0.035
MMRec GRU 1L	6.16 ± 4.69	2.242 ± 0.285
MMRec GRU 2L	6.89 ± 4.94	1.984 ± 0.044
MMRec FULL GRU	10.9 ± 6.45	1.884 ± 0.072

models. The shown times are for circa 7500 entries, so a single prediction is much faster. The time it takes is insignificant if we consider that it would only be made once an hour in an application.

For reproducibility the implementation of all strategies, the evaluation suite and the dataset can be found in the referred repository[3].

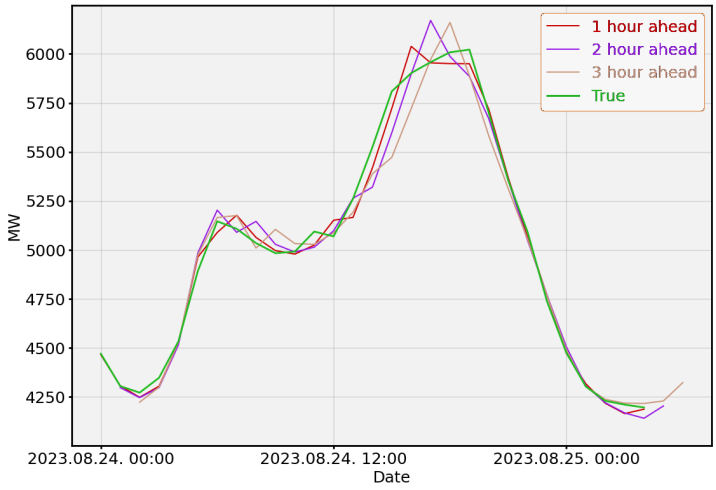
3.1 MMRec vs Seq2Seq

It was shown that MMRec outperforms the Recursive strategy but lacks the performance at its current stage to perform better than the Seq2Seq strategy on this dataset. This section provides a detailed explanation of the differences.

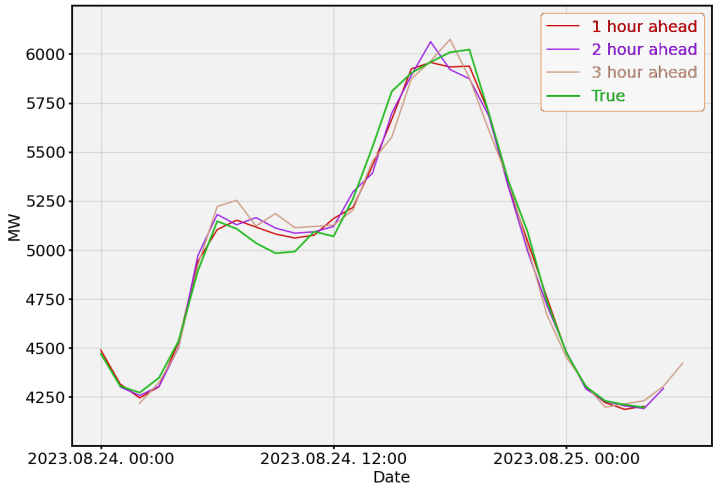
MAE and RMSE show a small difference between the two approaches. MPE varies more for MMRec, but this metric usually depends on the specific training run for these approaches. It isn't indicative of performance in this comparison. In terms of speed, the Seq2Seq model trains faster than MMRec - FULL GRU. Against the GRU 1L and 2L variants it doesn't have a clear advantage. MMRec however is a lot slower in prediction which can be critical for certain applications but isn't for this one. A comparison of the exact predictions for a specific date can be seen on Figure 7. In this example MMRec performs better in predicting the afternoon and Seq2Seq performs better in the morning.

Figure 8 shows the RMSE of the Seq2Seq and MMRec (FULL GRU approach here) strategies by how many hours they predicted. The main issue with MMRec that this graph displays is the first step being inaccurate in comparison to Seq2Seq (a). Interestingly MMRec accumulates less error as the prediction reaches farther distances (b). This hints at MMRec maybe performing closely to or better than Seq2Seq at longer prediction lengths. It could be argued that the algorithm of MMRec may cause this. For the first step the external feature models are not

[3] https://github.com/MeepOwned13/es_load_fs_HUN



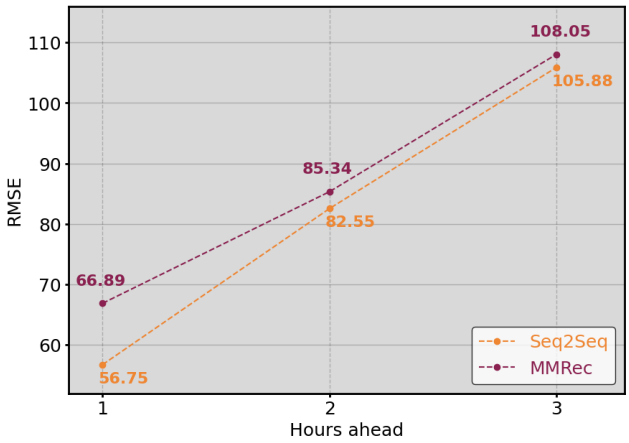
(a) Seq2Seq predictions example



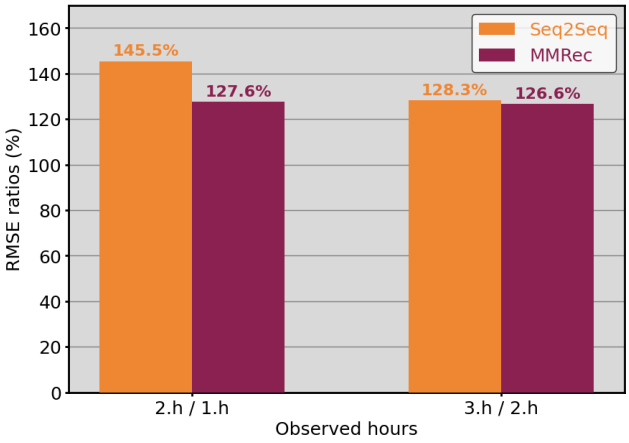
(b) MMRec predictions example

Figure 7: Comparing Seq2Seq and MMRec predictions for a specific date

involved since the real values are known at that stage. This being a 3-step prediction the optimization algorithm by default will prefer to optimize for the overall best average. Since all 3 models only get involved on step 2 and 3 it may lean heavier on the prediction of external feature models. This can cause a performance difference at step 1. This may be addressed by taking $n_T = n_T + 1$ for the input and using the external feature models to predict the ones it would already know.



(a) Error by hours predicted ahead



(b) Error accumulation bar plot

Figure 8: Comparing Seq2Seq and MMRec by RMSE per hour ahead prediction

4 Conclusion

This paper presented the MMRec (*Multi Model Recursion*) strategy for short term time series forecasting. Comparisons with existing time series forecasting strategies and models reveal that it outperforms the Recursive strategy it is based on. For the dataset constructed from Hungarian electricity load and weather data with the task of electricity load prediction it isn't the best performer in the comparison. The Seq2Seq strategy outperforms it in terms of MSE, RMSE, MPE and MAPE metrics.

MMRec shows promise in longer forecast horizons because it accumulates less error over time than Seq2Seq. This is counteracted in the 3-hour horizon by MMRec's worse performance at the first forecasting step. The conclusion is thus that MMRec may become a competitor to the mentioned strategies on some datasets given further refinements. This is backed by the observed error metrics being fairly close for MMRec and Seq2Seq.

4.1 Possible applications

Apart from the application to electricity load forecasting MMRec may be applied to any time series forecasting task where the external features that need to be forecasted by deep learning models are few. An interesting future application is choosing a task and dataset where external features more heavily influence the target feature. An example to this would be solar or wind electricity production. These heavily correlate with external weather features where getting a decent prediction for them could make a difference.

MMRec can also be used if forecasts of future external variables are sparse, for example if a better forecast for precipitation can be provided by outside models in certain cases but not always. In this case the model can take external forecasts by not using its own for that specific feature. Disruptions could also be caused by unforeseen events like hardware failures. In this case MMRec can operate without the need for its external feature models when everything is working as intended but use the external ones in the event that some forecasts are unavailable. For this use case MMRec doesn't require additional changes where other strategies would. The Recursive strategy also has these advantages but it was shown that MMRec outperforms it.

4.2 Outlook

The MMRec strategy has shown some advantages and disadvantages against the Seq2Seq strategy on the given dataset. A single dataset doesn't provide the full picture in these kinds of cases. Thus, the following future works can be specified:

- Apply the strategy model pairs to other datasets in electricity load forecasting.
- Apply the strategy model pairs to different time series forecasting tasks such as solar electricity production.
- Compare the strategies at longer forecast horizons such as 6, 12 and 24 hours.
- Compare the strategies at higher resolution on any time series forecasting task.
- Compare the strategy with more advanced methods such as Spacetimeformers introduced by Grigsby et al. [6].

References

[1] Azeem, A., Ismail, I., Jameel, S. M., and Harindran, V. R. Electrical load forecasting models for different generation modalities: A review. *IEEE Access*, 9:142239–142263, 2021. DOI: 10.1109/ACCESS.2021.3120731.

[2] Bai, S., Kolter, J. Z., and Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv, 2018. DOI: 10.48550/arXiv.1803.01271.

[3] Ben Taieb, S., Bontempi, G., Atiya, A. F., and Sorjamaa, A. A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. *Expert Systems with Applications*, 39(8):7068–7070, 2012. DOI: 10.1016/j.eswa.2012.01.039.

[4] Gasparin, A., Lukovic, S., and Alippi, C. Deep learning for time series forecasting: The electric load case. *CAAI Transactions on Intelligence Technology*, 7(1):3–14, 2022. DOI: 10.1049/cit2.12060.

[5] Ghavidel, S., Li, L., Aghaei, J., Yu, T., and Zhu, J. A review on the virtual power plant: Components and operation systems. In *IEEE International Conference on Power System Technology (POWERCON)*, pages 1–6, 2016. DOI: 10.1109/POWERCON.2016.7754037.

[6] Grigsby, J., Wang, Z., and Qi, Y. Long-range transformers for dynamic spatiotemporal forecasting. arXiv, 2021. DOI: 10.48550/arXiv.2109.12218.

[7] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., and Chen, T. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–357, 2018. DOI: 10.1016/j.patcog.2017.10.013.

[8] Howell, E. How to correctly perform cross-validation for time series. Towards Data Science, 2024. URL: https://towardsdatascience.com/how-to-correctly-perform-cross-validation-for-time-series-b083b869e42c.

[9] Masood, Z., Gantassi, R., Ardiansyah, and Choi, Y. A multi-step time-series clustering-based Seq2Seq LSTM learning for a single household electricity load forecasting. *Energies*, 15(7):5–6, 2022. DOI: 10.3390/en15072623.

[10] Masum, S., Liu, Y., and Chiverton, J. Multi-step time series forecasting of electric load using machine learning models. In Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R., and Zurada, J. M., editors, *Artificial Intelligence and Soft Computing*, pages 151–153, Cham, 2018. Springer International Publishing. DOI: 10.1007/978-3-319-91253-0_15.

[11] Nti, I. K., Teimeh, M., Nyarko-Boateng, O., and Adekoya, A. F. Electricity load forecasting: a systematic review. *Journal of Electrical Systems and Information Technology*, 7(1):13, 2020. DOI: 10.1186/s43067-020-00021-8.

[12] Probst, P., Wright, M. N., and Boulesteix, A.-L. Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, 9(3):1–7, 2019. DOI: 10.1002/widm.1301.

[13] Shen, G., Tan, Q., Zhang, H., Zeng, P., and Xu, J. Deep learning with gated recurrent unit networks for financial sequence predictions. *Procedia Computer Science*, 131:897–898, 2018. DOI: 10.1016/j.procs.2018.04.298.

[14] Yazici, I., Beyca, O. F., and Delen, D. Deep-learning-based short-term electricity load forecasting: A real case application. *Engineering Applications of Artificial Intelligence*, 109:104645, 2022. DOI: 10.1016/j.engappai.2021.104645.