Deep Learning for Solar Power Forecasting – An Approach Using Autoencoder and LSTM Neural Networks

André Gensler, Janosch Henze, Bernhard Sick
Intelligent Embedded Systems Group
University of Kassel, Germany
Email: {gensler | jhenze | bsick}@uni-kassel.de

Nils Raabe
enercast GmbH
Kassel, Germany
Email: n.raabe@enercast.de

***Abstract*—Power forecasting of renewable energy power plants is a very active research field, as reliable information about the future power generation allow for a safe operation of the power grid and helps to minimize the operational costs of these energy sources. Deep Learning algorithms have shown to be very powerful in forecasting tasks, such as economic time series or speech recognition. Up to now, Deep Learning algorithms have only been applied sparsely for forecasting renewable energy power plants. By using different Deep Learning and Artificial Neural Network algorithms, such as Deep Belief Networks, AutoEncoder, and LSTM, we introduce these powerful algorithms in the field of renewable energy power forecasting. In our experiments, we used combinations of these algorithms to show their forecast strength compared to a standard MLP and a physical forecasting model in the forecasting the energy output of 21 solar power plants. Our results using Deep Learning algorithms show a superior forecasting performance compared to Artificial Neural Networks as well as other reference models such as physical models.**

***Index Terms*—Multi-layer neural network, Recurrent neural networks, Power system analysis computing, Solar Energy, Forecasting**

I. Introduction

Research in techniques for regenerative power forecasting has been a major area of interest during the last decade, as more and more regenerative generators are integrated into the power grid. Regenerative generators have volatile energy characteristics, which means that they cannot be controlled such as conventional power plants, but generate energy based on the current weather situation.

The integration of these novel forms of power plants into the power grid is one of the big challenges that the industry currently faces. Due to the increasing portion of regenerative energy in the power mix, sophisticated algorithms have to predict the future energy generation in a reliable manner. These forecasts are likewise relevant to power plant operators, the energy trading market, and grid operators. With knowledge about the future energy generation development in the grid, the technical, as well as the financial risks, are reduced for all participants in the market.

As regenerative power generation mainly depends on the weather situation, most forecasting algorithms also take the weather forecast into account. The typical process thereby is:

- A *numerical weather prediction* (NWP) is created for a certain time horizon and a certain location using physical weather forecasting models.
- Based on this NWP, a *forecasting algorithm* predicts the future power generation for a certain regenerative energy generator.

In this article we focus on the second step of this forecasting process, i.e., we take an NWP forecast as given.

The quality of the power forecasting algorithm heavily depends on the NWP, which means it decreases with increasing time horizon. Forecasting algorithm horizons range from short-term forecasts in the range of hours, mid-term forecasts up to some days, to long-term forecasts in the range of some weeks. Of course, different time horizons are of interest for different market participants. In particular, the *day-ahead* forecast ($24\text{h} - 48\text{h}$), a typical mid-term forecast, is of interest to many actors in the field, and we will, therefore, put an emphasis on this type of forecast.

Forecasting algorithms typically are either based on physical models, which use knowledge about the weather situation and the power plant to generate a forecast. These models typically calculate wind turbine / solar power curves. An advantage of these models is easy comprehensibility and high accuracy if all parameters can be measured precisely.

On the other hand, models from machine learning or statistics have proven to be successful in the domain of power forecasting. In contrast to physical models, these forms of models do not explicitly model the physical process including the power generation, but the input-output relationship of NWP parameters to the power output. Thereby, they can abstract from inherent errors during measurement and can be applied to a wide variety of tasks in a flexible manner. Typical approaches in the area of machine learning are (multi-)linear regression techniques or artificial neural networks (ANN).

During the last years, classes of neural networks with many hidden layers, which are referred to as *Deep Neural Networks* (DNN), have emerged in the machine learning community with

978-1-5090-1897-0/16/$31.00 ©2016 IEEE

tremendous success. DNN have shown strengths in different fields such as representation learning and time series forecasting. DNN find applications in speech and image recognition, machine translation, and forecasting of financial time series, for instance. Especially, DNN architectures have proven to be powerful in learning feature representations of data. Therefore, the effort to manually engineer features, e.g., for a forecasting task, is minimized. Even more recently, DNN architectures have shown to improve regenerative power forecasting results.

The main contribution of this article is the investigation of different ANN and DNN architectures in the field of solar power forecast. In this work, we analyze a physical forecasting model, Multilayer Perceptrons, Long-Short-Term-Memory Networks (LSTM), Deep Belief Networks, and AutoEncoders. All models are compared regarding their forecasting accuracy of the solar power output of 21 photovoltaic facilities.

The remainder of this article is organized in the following way: In Section II, we highlight related work in the area of regenerative power forecasting and Deep Learning techniques. Section III outlines the different forms of Deep Neural Networks we used. In Section IV, we describe our experimental setup and results, which are discussed in Section V. The key findings of this article are summarized in Section VI.

II. Related Work

Forecasting the power of regenerative energy sources is not a new research topic. An excellent survey of prior work in solar power forecasting can be found in [1]. Other overview papers also cover the forecasting of other types of regenerative power plants [2], [3], [4].

One possible way to categorize different approaches is to sort them by their forecasting horizon. The area of short-term forecasting is covered in [3], [5], mid-term forecasting is described in [6], long-term forecasting techniques are described in [7].

Another possibility to group forecasting techniques is by their methodical foundations. A typical categorization would thereby be physical models, such as NWP in combination with wind or solar power curves, machine-learning models, such as artificial neural networks, and statistical models, e.g., the ARIMA model.

Artificial Neural Networks (ANN) are a popular method for forecasting tasks. The applied architectures range from simple MLP [8] to more complex networks such as recurrent neural networks [9] or time delay neural networks [10]. Different network architectures have been applied to forecast renewable energy sources [11]. [12] shows different network topologies perform better than other in forecasting tasks.

In recent years, ANN have regained attention in research as Deep Learning emerged [13]. Research in Deep Learning focuses on different network structures, e.g., AutoEncoders, Deep Belief Networks, and LSTM, for tasks such as data encoding, information extraction, or forecasting [14]. AutoEncoders and Deep Belief Networks are used to learn representations of the data [13], [14]. LSTM networks are a subcategory of recurrent neural networks and use additional memory cells to be able to store states. Hence, they have shown to be powerful for time series prediction tasks [15], [16].

Deep Learning architectures have recently been used to forecast regenerative energy sources. For example, [17] uses Deep Belief Networks to predict wind power and [18] uses Stacked AutoEncoders to predict short term wind speed.

In other domains, special forms of polynomial approximations, which turn out to be very computationally inexpensive [19] have been successfully used. Probabilistic forecasts are an increasingly popular field in power forecasting, covered, e.g., in [20]. In the domain of statistical forecasting techniques, [21] use wavelets and an ARIMA technique for forecasting. A forecasting technique based on Kalman filters is described in [22]. The authors of [23] use a physical model with an up-scaling technique for power forecasting. In the area of hybrid models, [24] use machine learning techniques in conjunction with turbine power curves. The creation of forecasting ensembles can improve the overall forecasting quality. An overview of various ensemble techniques is shown in [25], a hybrid ensemble technique based on the gradual weighting of weather situations is described in [26].

Deep Learning is an active field of research, which has several applications in renewable energy forecasting. Mostly, these applications are in the field of wind power forecast. Applications of Deep Learning algorithms are only sparsely publicized in the field solar power forecasts. Other research in solar power forecasting focuses on methods such as ANN, regressive methods, or NWP. Hence, we aim to introduce different Deep Learning algorithms, such as LSTM, DBN, and AutoEncoder for solar power forecasting.

III. Deep Neural Network Approaches

In this section, we introduce the ANN and Deep Neural Network architectures we used for the forecasting of solar power output. In Section III-A a standard Multilayer perceptron (MLP) is introduced, followed by the Long Short-Term Memory network (LSTM) in Section III-B, the Deep Belief Network (DBN) in Section III-C and an AutoEncoder with an LSTM in the back of the topology (Auto-LSTM) in Section III-D. The applied activation functions are explained in Section III-E.

A. Multilayer Perceptron

As a basic architecture, a Multilayer Perceptron (MLP) [27] is used. As shown in Figure 1, the MLP consists of multiple fully connected layers of neurons.

$$output = f(\sum_{i}^{Inputs} (x_i \cdot w_i + b_i)) \quad (1)$$

Equation 1 calculates the output of a neuron, where x is the input of the neuron, w is the weight on each connection to the neuron, b is the bias and $f(\ldots)$ is the activation function, e.g., the tanh activation function.

The training of the MLP is performed using a back propagation algorithm.

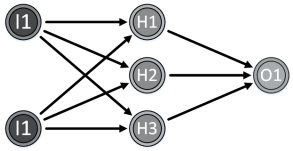

Fig. 1. An exemplary MLP network topology.

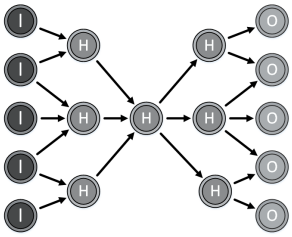

Fig. 2. An exemplary AutoEncoder topology.

B. Long Short-Term Memory

The Long Short-Term Memory networks (LSTM) are based on Recurrent Neural Networks (RNN). Unlike "normal" MLP, RNN use temporal information of the input data. An RNN realizes this ability by recurrent connections between the neurons. An LSTM has a special neuron structure called memory cell. These memory cells have the ability to store information over an arbitrary time. Three gates control the information flow into and out of the neuron's memory cell: the input, output, and forget gate. Each gate in the LSTM gets the same input as the input neuron. Furthermore, each gate possesses an activation function.

For example, if the input gate notes a high activation, the input will be stored in the memory cell. If the output gate notes a high activation, it will release the stored information to the next neurons. If the forget gate notes a high activation, the memory cell will be cleared.

The main difference in training an LSTM in contrast to a MLP is, that to predict a value p_t at time t, the previous n samples $\{p_{t-n}, \ldots, p_{t-1}\}$ have to be propagated through the net. This settling time n has to be defined while setting up the network. The memory cells will store the temporal information according to their training status and deliver the predicted output p_t.

In this work, we used $n = 2$ previous samples to predict a new value. Figure 3 shows that this is the best choice for the LSTM. The RMSE decreases if the previous two timesteps are considered instead of just one previous time step. If more than two previous time steps are considered, the RMSE rises again.

C. Deep Belief Network

A deep belief network (DBN) for forecasting can be seen as a two-step algorithm:

1) The DBN performs a feature learning to reduce the dimensionality of the input data set.
2) An additional layer, e.g., a linear layer, is added to carry out the forecasting.

In more detail, each layer of a DBN consists of a Restricted Boltzmann Machine (RBM). An RBM is a two layer stochastic ANN which learns a probability distribution over the input data set. The layers are organized like a funnel, which helps to learn features within the data. To train a DBN for regression, two training steps have to be performed: In the first training step, the DBN is trained in an unsupervised manner by contrastive divergence [28]. This results in a topology which can abstract the data to a reduced set of features. The second training step starts with appending an ANN, e.g., one layer of fully connected neurons, to the pre-trained topology. The newly attached layers have to be trained with the desired targets to perform a prediction. It is possible to train only the newly attached layer or to train the whole DBN again.

D. Auto-LSTM

The Auto-LSTM algorithm combines the feature learning of an AutoEncoder with the temporal context usage of an LSTM in a two-step approach:

1) An Auto-Encoder (AE) will be used to realize the feature learning.
2) An LSTM network is attached to the encoding part of the AE. Hence, it uses temporal information in form of sequences of the extracted features.

An AE is an MLP with a particular network topology shown in Figure 2. The input and output layers have the same size. The neurons in each layer are reduced starting from the input layer. This side is called the encoding side. At the center, also called bottleneck of the AE, the layers are mirrored to produce the decoding side of the AE. The AE is trained to reconstruct the input on the output side. The idea is that the bottleneck will serve as a feature extractor of the input data.

After training the AE, the network topology will be cut after the bottleneck and an LSTM is attached. This allows using the learned encoding as an input to the LSTM. Afterward, the newly created network architecture is trained to produce the desired output.

The Auto-LSTM also uses the $n = 2$ previous timesteps to predict a new value. Figure 3 shows the error progression with increasing previous timesteps. The error of the Auto-LSTM increases with additionally considered time steps during the forecast. It has to be noted that there is a sudden increase in the RMSE if two previous time steps are considered in the forecast. Nevertheless, we choose to use two previous time steps as the RMSE increase is very low > 0.0001 and the run time increases with increasing number of considered previous time steps.

E. Activation Functions

All ANN and DNN models used tanh activation function in all layers except in the output layer. In the output layer, a

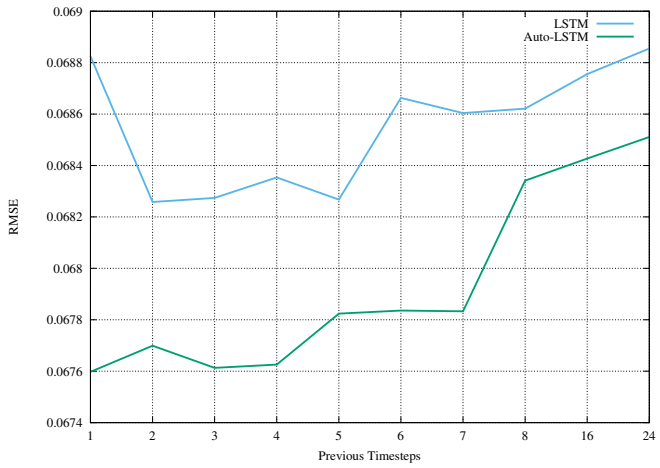

Fig. 3. RMSE progression of the LSTM and Auto-LSTM with increasing consideration of previous time steps to setup the memory cells.

Rectified Linear Unit (ReLU) was used as activation function for the prediction. The ReLU activation function is defined as

$$act_{ReLU}(x) = \max(0, x) \tag{2}$$

The tanh activation function as

$$act_{\tanh}(x) = \tanh(x) \tag{3}$$

Figure 4 shows two different power forecasts from an Auto-LSTM and the measured power. One forecast was done with a tanh activation and one with ReLU activation function in the last layer. It can be seen, that the tanh activation function allows negative power predictions. In contrast the characteristics of the ReLU activation function cut the negative power predictions off.

IV. Experimental Evaluation

This section contains the experimental evaluation of the forecasting methods described in Section III. We start by introducing our reference model in Section IV-A and the Clear-Sky Filter in Section IV-B. Afterward, we explain the data set we used for the experimental evaluation in Section IV-C and the applied error functions in Section IV-D. The experimental setup is described in Section IV-E. This section concludes with the results of the experiment in Section IV-F.

A. Reference Model

A physical photovoltaic forecasting model (P-PVFM) is used as a reference. In contrast to the DNN approaches, the P-PVFM is not trained on the data set. The model uses the angles and cardinal direction of the photo-voltaic panels to initialize itself. For the prediction, it uses NWP parameters such as the temperature and the direct and diffuse solar radiation. In conjunction with this information, the model calculates the produced power for a specific weather situation and a given forecasting horizon.

B. Clear-Sky Filter

The power forecasts are filtered with the Clear-Sky filter. The Clear-Sky filter uses the Clear-Sky curve [29], which estimates terrestrial solar radiation of the sun with a cloudless sky for a specific time and coordinates. The Clear-Sky filter prevents a power forecast during the night time. The power output forecast is matched with the values of the Clear-Sky filter. If the Clear-Sky curve estimated a value of terrestrial solar radiation equal to zero, the power forecast is set to zero as well.

C. Data Set Description

Our experiments are conducted on the *GermanSolarFarm* data set [30]. The data set contains 21 photovoltaic facilities in Germany. Their installed nominal power ranges between 100kW and 8500kW. The PV facilities range from PV panels installed on rooftops to fully fledged solar farms. They are distributed throughout Germany as shown in Figure 5.

For each facility historical NWP data and the produced power in a three-hour resolution for 990 days are available.

All time series in the data set, except the measured power output, are normalized between 0 and 1 using the min-max normalization. The target variable, the measured power output, is normalized using the nominal output capacity of the corresponding PV facility. Therefore, allow the comparison of the forecasting performance without taking the size of the PV facilities into account.

The data set can be downloaded for free at [30].

D. Error Measures

Five error measures are used to evaluate the performance of the different approaches and the reference method: the root-mean-squared error (RMSE) as shown in Eq. 4, the mean absolute error (MAE) as shown in Eq. 5, the average absolute deviation as shown in Eq. 6, the BIAS as shown in Eq. 7 and the correlation between the forecast and the measured power as shown in Eq. 8. The equations use the measured power time series $\mathbf{x}$ and the predicted power time series $\mathbf{x}'$. Both time series have N samples.

$$\mathrm{RMSE}(\mathbf{x}', \mathbf{x}) = \sqrt{\frac{1}{N} \cdot \sum_{n=1}^{N} (x'_n - x_n)^2} \tag{4}$$

$$\mathrm{MAE}(\mathbf{x}', \mathbf{x}) = \frac{1}{N} \cdot \sum_{n=1}^{N} |x'_n - x_n| \tag{5}$$

$$\mathrm{AbsDev}(\mathbf{x}', \mathbf{x}) = \frac{1}{\sum_{n=1}^{N} x_n} \cdot \sum_{n=1}^{N} |x'_n - x_n| \tag{6}$$

$$\mathrm{BIAS}(\mathbf{x}', \mathbf{x}) = \frac{1}{N} \cdot \sum_{n=1}^{N} (x'_n - x_n) \tag{7}$$

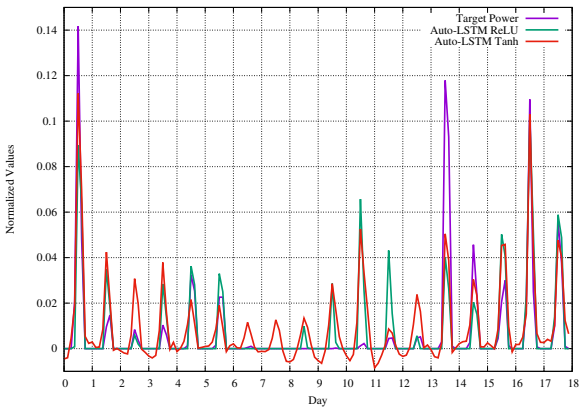

Fig. 4. Forecasting of the Auto-LSTM the with ReLU and tanh activation function.

Fig. 5. Locations of the evaluated photovoltaic facilities of the *German-WindFarm* data set.

$$\text{Correlation}(\mathbf{x}', \mathbf{x}) = \frac{\sum_{n=1}^{N} (x'_n - \overline{x}') \cdot \sum_{n=1}^{N} (x_n - \overline{x})}{\sqrt{\sum_{n=1}^{N} (x'_n - \overline{x}')^2 \cdot \sum_{n=1}^{N} (x_n - \overline{x})^2}} \quad (8)$$

TABLE I
TRAINING PARAMETERS FOR THE ANN AND DNN MODELS

Model	Learning Rate	Training Iterations	Trained Models
MLP	$[0.3, 0.005]$	700	75
LSTM	$1 \cdot 10^{-3}$	200	30
DBN	$1 \cdot 10^{-3}$	150, 700	25
Auto-LSTM	$1 \cdot 10^{-3}$	200, 700	20

These Error Measures can be used to assess the power forecast. If RMSE $\gg$ MAE the forecast has high deviations to the measured power output. If RMSE $\approx$ MAE the forecast has only small deviations to the measured power output. The BIAS allows assessing whether power forecast is predicting higher or lower values than the measured power output. The AbsDev measure is commonly used within the energy sector to assess the quality of the forecast. The Correlation is an additional measure to assess the similarity of the power forecast and the measured power output.

E. Experimental Setup

The training parameters of the ANN and DNN models, introduced in Section III, are stated in Table I.

The data set is split into 490 days for training, 250 days for validation and 250 days for testing. The models are trained using the training set, and their training error is calculated after each training iteration on the validation set. The network with the lowest error on the validation set is chosen for a final performance evaluation on the test data set. Furthermore, all models used an early stopping condition during the training, which stops the training if the RMSE on the validation data does not change within 50 training iterations. As the P-PVFM does not need training, the model is applied to all data partitions. The training error of the P-PVFM is calculated on the validation partition, even though it does not need training.

The MLP and LSTM network use input data selected by the best correlation to the target variable and expert knowledge of a meteorologist. As the DBN and the Auto-LSTM are capable of extracting features within the data, they use all available features as input.

All models had to predict the day-ahead forecast horizon of 24h to 48h. All ANN and DNN models were implemented using Keras [31].

The MLP adapted the learning rate in the range stated in Table I. Further improvements of the MLP include:

- Regularization of the weight update of $1 \cdot 10^{-6}$ to enable appropriate weight decay.
- A momentum term of 0.9 to speed up the training.

In total 75 MLP and 30 LSTM models with a 4 layer topology are trained. The LSTM used the samples of 2 previous time steps, which is equal to 6 hours of weather forecast data, to initialize the memory cells.

After the DBN is trained, a layer with Rectified Linear Unit (ReLU) activation functions is attached to the DBN. Hence, two training cycles (pretraining and fine tuning) are needed to train the DBN fully. During the pretraining, only the DBN is trained using contrastive divergence to optimize the weights. In the second step, a layer with ReLU activation functions is added to the DBN and a fine tuning using an RMSprop optimizer is conducted. During the second step, only the weights of the additional layer are trained. The maximum number of training cycles during pretraining are 150 and 700 during fine tuning, respectively.

The Auto-LSTM is trained similarly to the DBN. During the pretraining only the AutoEncoder is trained. Afterward, the AutoEncoder is split at the bottleneck, and an LSTM network is appended. This LSTM layer also uses the samples of 2 previous time steps to initialize the memory cells. During the fine tuning only the LSTM network weights are trained. The maximum training cycles during pretraining are 200 and 700 during fine tuning.

TABLE II
RMSE VALUES FOR THE DIFFERENT TRAINING AND TEST DATA SETS, CALCULATED FOR EACH MODEL ON EVERY PREDICTED PV FACILITY. THE COLOR CODING INDICATES LARGE ERRORS IN RED AND LOW ERRORS IN GREEN. FURTHER ERROR MEASURES, I.E., MAE, ABSOLUTE DEVIATION, BIAS, AND CORRELATION WERE AVERAGED OVER ALL FACILITIES.

Data	**P-PVFM**		**MLP**		**LSTM**		**DBN**		**Auto-LSTM**	
	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train
pv01	0.0954	0.0987	0.0633	0.0620	0.0636	0.0602	0.0620	0.0607	0.0627	0.0581
pv02	0.1206	0.1265	0.0588	0.0632	0.0571	0.0619	0.0578	0.0614	0.0561	0.0605
pv03	0.1170	0.1208	0.0474	0.0457	0.0474	0.0460	0.0458	0.0444	0.0452	0.0434
pv04	0.1155	0.1269	0.0436	0.0474	0.0445	0.0481	0.0443	0.0473	0.0440	0.0441
pv05	0.1060	0.1505	0.0663	0.0558	0.0726	0.0601	0.0653	0.0552	0.0643	0.0526
pv06	0.1154	0.1105	0.0817	0.0730	0.0814	0.0734	0.0816	0.0725	0.0807	0.0721
pv07	0.1231	0.1000	0.1043	0.0722	0.1035	0.0697	0.1044	0.0664	0.1013	0.0677
pv08	0.0929	0.0893	0.0926	0.0794	0.0911	0.0813	0.0920	0.0808	0.0873	0.0769
pv09	0.0997	0.1110	0.0665	0.0658	0.0669	0.0640	0.0660	0.0625	0.0676	0.0610
pv10	0.1387	0.1403	0.0544	0.0507	0.0537	0.0487	0.0539	0.0479	0.0536	0.0486
pv11	0.1118	0.1162	0.0961	0.0937	0.0939	0.0926	0.0940	0.0878	0.0950	0.0883
pv12	0.1086	0.1208	0.0994	0.1004	0.0963	0.0980	0.0980	0.0993	0.0967	0.0975
pv13	0.1087	0.1107	0.0990	0.0871	0.0978	0.0853	0.0936	0.0801	0.0937	0.0809
pv14	0.0846	0.0958	0.0632	0.0633	0.0645	0.0629	0.0637	0.0629	0.0640	0.0594
pv15	0.0971	0.1013	0.0663	0.0650	0.0692	0.0643	0.0655	0.0639	0.0679	0.0616
pv16	0.0975	0.1062	0.0717	0.0734	0.0718	0.0717	0.0713	0.0716	0.0688	0.0693
pv17	0.1063	0.1198	0.0645	0.0650	0.0676	0.0664	0.0636	0.0628	0.0638	0.0610
pv18	0.1220	0.1259	0.0589	0.0607	0.0578	0.0596	0.0592	0.0591	0.0624	0.0562
pv19	0.1054	0.1100	0.0693	0.0663	0.0677	0.0658	0.0668	0.0650	0.0678	0.0627
pv20	0.0973	0.1195	0.0774	0.0644	0.0762	0.0634	0.0768	0.0633	0.0792	0.0575
pv21	0.1179	0.1106	0.0749	0.0771	0.0762	0.0733	0.0731	0.0777	0.0758	0.0696
Avg. RMSE	0.1086	0.1148	0.0724	0.0682	0.0724	0.0675	0.0714	0.0663	**0.0713**	**0.0642**
Avg. MAE	0.0560	0.0585	0.0372	0.0344	0.0368	0.0337	0.0367	0.0334	**0.0366**	**0.0323**
Avg. Abs. Dev.	0.4368	0.5123	0.2809	0.2891	0.2786	0.2834	0.2772	0.2813	**0.2765**	**0.2714**
Avg. BIAS	0.0399	0.0463	**-0.0011**	**-0.0031**	-0.0073	-0.0042	-0.0024	-0.0043	-0.0021	-0.0041
Avg. Corr.	0.9294	0.9160	0.9344	0.9361	0.9352	0.9375	**0.9363**	0.9399	0.9362	**0.9431**

The forecasting results of all models are filtered using the Clear-Sky Filter.

F. Experimental Results

Table II shows the errors scores of the different models during the experiment. For each model, the RMSE for all facilities is shown. The color coding gives a better overview of the error values. A high error is indicated by a red color, and a low error is indicated by a green color. The P-PVFM was evaluated on all partitions of the data set for comparison. Furthermore, for each model the average of the RMSE, MAE, absolute deviation, BIAS, and correlation is shown. The best-averaged results are bold, for the training and test set.

All analyzed ANN and DNN models outperform the P-PVFM. The best DNN model is the Auto-LSTM. It shows the best score with an average RMSE of 0.0713, an average MAE of 0.0366, and an average absolute deviation of 0.2765. The MLP has the best BIAS values. The best correlation on the unseen test data is obtained with the DBN and on the training data with the Auto-LSTM.

The RMSE on the single PV facilities allows identifying easily predictable facilities such as pv03, pv04, and pv10. These PV facilities have low RMSE values when predicted with the ANN and DNN models. Furthermore, the PV facilities pv11, pv12, and pv13, which possess a high error, are so-called tracker facilities, i.e. they adjust their position to the sun.

V. DISCUSSION

As seen in Table II, the Auto-LSTM network is the best performing DNN. It combines the feature extraction ability of the AutoEncoder with the forecasting ability of the LSTM.

The DBN has error values on par with the Auto-LSTM. The DBN also has the ability to extract features. The slightly increased forecast error might be due to the attached linear ANN layer instead of an LSTM layer.

Additionally, Table II shows that the PV facilities pv11, pv12, and pv13 are not as easy to predict as other PV facilities. Other facilities, i.e., pv07 and pv08, show a low forecasting error on the training data, but the forecasting error on the test data is high. These error values show that some of the PV facilities possess some unpredictable characteristics.

In Section III-E we explained that the preferred activation function in the output layer is a ReLU. The reason for this choice is explained with the help of Figure 6.

Figure 6 shows the predicted power output of the Auto-LSTM during two weeks in January, as a green line. Furthermore, it shows the measured power output (purple line), snow depth (blue line), and snowfall (yellow line) during the same two weeks. The *target power* is the measured energy output of a specific solar facility. The *target power* is nearly zero between the measurements of day 6 and day 11. As the photovoltaic panels may be covered by snow. This is supported by the increase in snowfall and the rise of the snow depth.

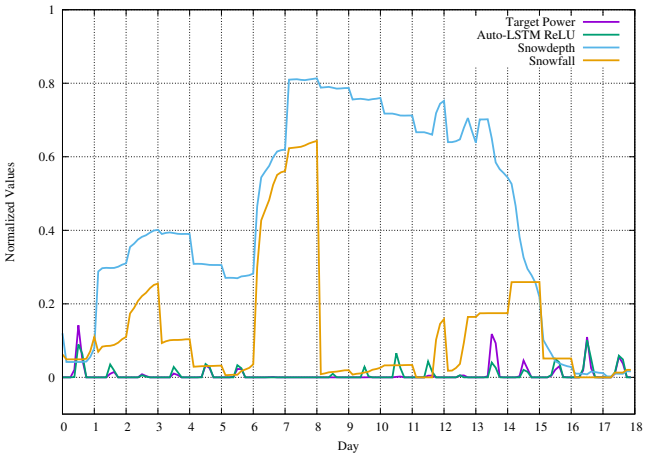

Fig. 6. Forecasting of the Auto-LSTM with ReLU activation function.

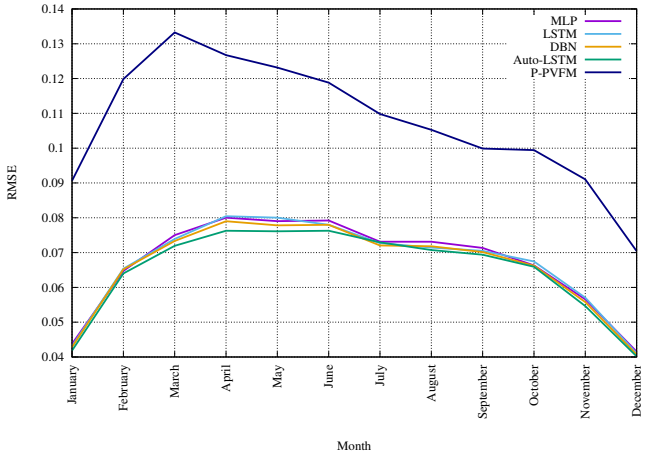

Fig. 7. The RMSE averaged over all facilities throughout the year for the different forecast models.

TABLE III
THE MEAN ABSOLUTE ERROR (MAE) AND THE ABSOLUTE DEVIATION ERROR (ADE) AVERAGED OVER ALL FACILITIES. DEMONSTRATING THE ABSOLUTE ERROR REDUCTION, IF A RELU ACTIVATION FUNCTION WITH CLEAR-SKY FILTER IS USED INSTEAD OF A TANH ACTIVATION FUNCTION.

	avg. MAE		**avg. ADE**	
	tanh	ReLU + CSF	tanh	ReLU + CSF
MLP	0.0361	0.0338	0.3543	0.3294
LSTM	0.0361	0.0333	0.3559	0.3257
DBN	0.0369	0.0334	0.3658	0.3247
Auto-LSTM	0.0377	0.0336	0.3727	0.3288

Furthermore, the better performance of the ReLU over the tanh activation function is supported by the decrease of the MAE as shown in Table III. The mean absolute error between the forecasting with the different activation function and the measured power output. It can be seen that the MAE is higher when using the tanh activation function in the last layer instead of a ReLU activation function.

Figure 7 shows the RMSE averaged over all PV facilities for each month of the test data. During the winter months, the RMSE is smaller than during the summer months. This effect exists because the PV facilities generate less energy during the winter months than in the summer months. Hence, the normalized power output is smaller during the winter.

Furthermore, Figure 7 shows that the ANN and DNN models perform better than the P-PVFM. One reason for this is that the ANN and DNN models take weather situations such as snowfall and snow depth into account while the P-PVFM does not. Additionally, the RMSE of the P-PVFM is not very constant in the summer months; it drops from an RMSE of 0.1332 in March to an error value of 0.0998 in September. The ANN and DNN show a similar but smaller variation in the error. The error for the ANN and DNN rises from 0.0735 in March to about 0.0782 in May and then drops below 0.0703 starting from September. As previously stated, we expected the RMSE to rise during the summer and then to drop during the winter due to less produced power. Apparently, the RMSE is more constant during summer in the ANN and DNN than in the P-PVFM.

VI. CONCLUSION AND OUTLOOK

The results in Table II showed that the ANN and DNN models have a lower RMSE than the physical reference model. The best performing model is the Auto-LSTM with an RMSE of 0.0713, closely followed by the DBN with an RMSE of 0.0714. This shows the feature extraction capability of these models, which enables a good solar power forecast. Both models without this capability, the MLP and the LSTM without an AE, perform worse.

According to the BIAS, the P-PVFM overestimates the power output of the solar facilities and the DNN models slightly underestimate the power output. In an application of these models, it needs to be taken into account if an overestimation or an underestimation is preferred.

If a Rectified Linear Unit (ReLU) as activation function for the neural network is chosen over a tanh activation function, it is possible to eliminate erroneous prediction data. This effect of the tanh activation function is especially prominent during the winter when there is snowfall. The reason for this might be that the network learned that during snowfall, the energy output is reduced and tries to reduce the prediction even further. The ReLU can resolve this issue as it only propagates a value if the inputs are positive.

In future work, we plan to combine DBN and LSTM to improve the forecasting error further. Further improvements might also be made by analyzing additional DNN architectures, such as Convolutional Neural Networks, which can learn filter functions, in combination with LSTMs.

Other research directions include the combination of different forecast models to an ensemble. During our work, we trained multiple models but only used the best performing model. By combining the different models depending on their individual strength, we might be able to increase the forecasting quality further.

The performance achieved by DNN architectures in forecasting of solar power might also be transferred to other regenerative energy sources, e.g., forecasting of wind power output.

The techniques presented in this article will also be applied to energy decision support systems for smart cities in the BESOS [32] project.

ACKNOWLEDGMENTS

This article results from the project Big Energy (HA project no. 472/15- 14), which is funded in the framework of Hessen Modell-Projekte, financed with funds of LOEWE Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz, Förderlinie 3: KMU-Verbundvorhaben (State Offensive for the Development of Scientific and Economic Excellence).

REFERENCES

[1] R. H. Inman, H. T. C. Pedro, and C. F. M. Coimbra, "Solar forecasting methods for renewable energy integration," *Progress in Energy and Combustion Science*, vol. 39, no. 6, pp. 535–576, 2013.

[2] A. M. Foley, P. G. Leahy, A. Marvuglia, and E. J. McKeogh, "Current methods and advances in forecasting of wind power generation," *Renewable Energy*, vol. 37, no. 1, pp. 1–8, 2012.

[3] G. Giebel, R. Brownsword, G. Kariniotakis, M. Denhard, and C. Draxl, "The State-Of-The-Art in Short-Term Prediction of Wind Power A Literature Overview," *Technical Report, ANEMOS.plus*, pp. 1–109, 2011.

[4] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, and Z. Yan, "A review on the forecasting of wind speed and generated power," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 4, pp. 915–920, 2009.

[5] A. Costa, A. Crespo, J. Navarro, G. Lizcano, H. Madsen, and E. Feitosa, "A review on the young history of the wind power short-term prediction," *Renewable and Sustainable Energy Reviews*, vol. 12, no. 6, pp. 1725–1744, 2008.

[6] S. Mirasgedis, Y. Sarafidis, E. Georgopoulou, D. Lalas, M. Moschovits, F. Karagiannis, and D. Papakonstantinou, "Models for mid-term electricity demand forecasting incorporating weather influences," *Energy*, vol. 31, no. 2-3, pp. 208–227, 2006.

[7] T. Hong, J. Wilson, and J. Xie, "Long term probabilistic load forecasting and normalization with hourly information," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 456–462, jan 2014.

[8] R. L. Welch, S. M. Ruffing, and G. K. Venayagamoorthy, "Comparison of Feedforward and Feedback Neural Network Architectures for Short Term Wind Speed Prediction," in *International Joint Conference on Neural Networks*, 2009, pp. 3335–3340.

[9] T. Barbounis and J. Theocharis, "Locally recurrent neural networks for wind speed prediction using spatial correlation," *Information Sciences*, vol. 177, no. 24, pp. 5775–5797, 2007.

[10] W. Ji and K. C. Chee, "Prediction of hourly solar radiation using a novel hybrid model of ARMA and TDNN," *Solar Energy*, vol. 85, no. 5, pp. 808–817, 2011.

[11] R. H. Inman, H. T. C. Pedro, and C. F. M. Coimbra, "Solar forecasting methods for renewable energy integration," *Progress in energy and combustion science*, vol. 39, no. 6, pp. 535–576, 2013.

[12] G. P. Zhang, E. B. Patuwo, and H. Michael Y., "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[13] Y. Bengio, "Learning deep architectures for ai," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

[14] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *arXiv*, p. 10, 2015. [Online]. Available: http://arxiv.org/abs/1503.04069

[17] C. Q. Y Tao H Chen, "Wind Power Prediction and Pattern Feature Based on Deep Learning Method," pp. 1–4, 2014.

[18] M. Khodayar and M. Teshnehlab, "Robust Deep Neural Network for Wind Speed Prediction," 2015.

[19] A. Gensler, T. Gruber, and B. Sick, "Blazing fast time series segmentation based on update techniques for polynomial approximations," in *Proceedings of 13th IEEE International Conference on Data Mining Workshops (ICDMW13)*. Dallas, USA: IEEE, 2013, pp. 1002–1011.

[20] H. A. Nielsen, H. Madsen, T. S. Nielsen, J. Badger, G. Giebel, L. Landberg, K. Sattler, and H. Feddersen, "Wind Power Ensemble Forecasting," *Global Wind Power Conference*, pp. 28–31, 2004.

[21] A. J. Conejo, M. a. Plazas, R. Espínola, S. Member, and A. B. Molina, "Day-Ahead Electricity Price Forecasting Using the Wavelet Transform and ARIMA Models," *IEEE Transactions On Power Systems*, vol. 20, no. 2, pp. 1035–1042, 2005.

[22] W. Zhang and W. Wang, "Wind speed forecasting via ensemble Kalman Filter," in *IEEE International Conference on Advanced Computer Control (ICACC)*, vol. 2, 2010, pp. 73–77.

[23] U. Focken, M. Lange, and H.-P. Waldl, "Previento - A Wind Power Prediction System with an Innovative Upscaling Algorithm," in *Proceedings of the European Wind Energy Conference (EWEC)*, 2001, pp. 1–4.

[24] I. J. Ramirez-Rosado, L. A. Fernandez-Jimenez, C. Monteiro, J. Sousa, and R. Bessa, "Comparison of two new short-term wind-power forecasting systems," *Renewable Energy*, vol. 34, no. 7, pp. 1848–1854, 2009.

[25] Y. Ren, P. Suganthan, and N. Srikanth, "Ensemble methods for wind and solar power forecasting - A state-of-the-art review," *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 82–91, 2015.

[26] A. Gensler and B. Sick, "Forecasting Wind Power - An Ensemble Technique With Gradual Coopetitive Weighting Based on Weather Situation," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN16)*, Vancouver, Canada, 2016, pp. 1–9.

[27] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[28] G. E. Hinton, *Neural Networks: Tricks of the Trade: Second Edition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, ch. A Practical Guide to Training Restricted Boltzmann Machines, pp. 599–619.

[29] M. J. Reno, C. W. Hansen, and J. S. Stein, "Global horizontal irradiance clear sky models: Implementation and analysis," *SANDIA report SAND2012-2389*, 2012.

[30] A. Gensler, J. Henze, N. Raabe, and V. Pankraz, "GermanSolarFarm Data Set," 2016. [Online]. Available: http://ies-research.de/Software

[31] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[32] "Building energy decision support systems for smart cities." [Online]. Available: http://besos-project.eu/