Prediction Intervals With LSTM Networks Trained By Joint Supervision

1st Nicolás Cruz
Department of Electrical Engineering
University of Chile
Santiago, Chile
nicolascruz2187@gmail.com

2nd Luis G. Marín
Department of Electrical Engineering
University of Chile
Santiago, Chile
luis.marin@ing.uchile.cl

3rd Doris Sáez
Department of Electrical Engineering
University of Chile
Santiago, Chile
dsaez@ing.uchile.cl

***Abstract*—This paper presents an approach for prediction interval generation by training a LSTM neural network with a joint supervision Loss Function. The prediction interval model provides the expected value and the upper and lower bounds of the interval given a desired coverage probability. The prediction interval models based on LSTM networks are compared with the classical recurrent neural network approach and are tested using two case studies. The first case corresponds to the forecasting up to one day ahead of the demand profile of 20 dwellings from a town in the UK, and the second case corresponds to the net power from an energy community made up 30 dwellings with a 50% level of photovoltaic power penetration. By using LSTM networks as the backbone of the proposed architecture, high-quality intervals are obtained with a narrower interval width compared with the classical recurrent neural network approach. Furthermore, the information provided by the prediction interval based on the LSTM network could be used to develop robust energy management systems that, for example, consider the worst-case scenario.**

***Index Terms*—Prediction interval, LSTM, joint supervision, neural network, time series, microgrids, renewable.**

I. Introduction

Prediction interval models have been proposed to characterize uncertain phenomena, and therefore, they have been widely used in many real world applications [1]. The predicted outputs are intervals that represent, with a given coverage probability, the most likely region defined by the upper and lower bounds (black lines in Fig. 1) of the interval to which the output (targets) of the uncertain phenomena will belong (see Fig. 1). Point prediction generates a value close to mean or median and is unable to predict that level of randomness or uncertainty [2]. Therefore, prediction interval models aim to find the upper and lower values of the interval such that the interval is as narrow as possible and that the interval contains a certain percentage of measured data [3]. In this paper, the prediction interval models are used to represent both nonlinear behaviour and uncertainty derived from renewable energy sources and electrical demand. Additionally, developed intervals can be used in the design of robust control strategies for control of microgrids based on renewable resourses. The worst-case scenario defined by the lower and upper bounds of the interval or the various scenarios generated using this interval are the strategies used to design the robust controllers and the quality of the prediction interval play an important role in the performance of these controllers [4], [5].

On the other hand, in recent years Deep Neural Network approaches have been used to solve a variety of problems, from object classification and detection to text recognition and translation [6]. However, deep algorithms have only recently found success on the task of time series prediction. Structures such as Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) are starting to become the standard method to obtain accurate predictions of complex systems, such as time-series forecasting [7], [8]. Although a lot of work has been done to improve the accuracy of LSTM based predictions, few studies have been conducted on the task of estimating the uncertainty of such predictions. For instance, in [9] a Monte Carlo dropout approach for representing model uncertainty was presented. The same approach was used in

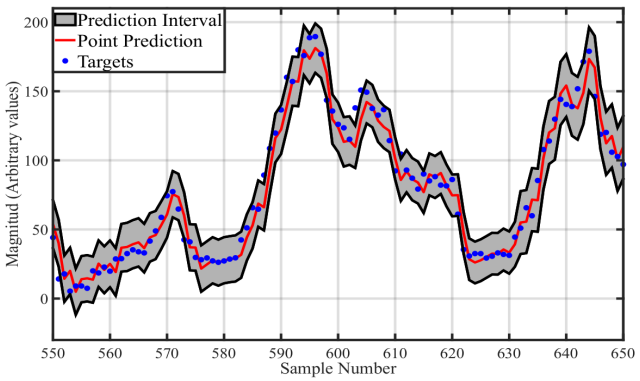

Fig. 1. Rough sketch for representing a Prediction Interval

This study was partially supported by the Complex Engineering Systems Institute (CONICYT – PIA – FB0816), the Solar Energy Research Center SERC-Chile (CONICYT/FONDAP/ Project under Grant 15110019) and FONDECYT Chile Grant Nr.1170683 "Robust Distributed Predictive Control Strategies for the Coordination of Hybrid AC and DC Microgrids".

978-1-7281-2009-6/$31.00 ©2019 IEEE paper N-20262.pdf

[10] where a LSTM encoder-decoder architecture was used to predict the daily completed trips processed by the Uber platform using uncertainty estimation based on the Monte Carlo dropout of hidden units. However, these approaches make several assumptions on the process distribution in order to obtain the prediction interval model.

LSTM networks are the current state of the art in time series prediction, however no research had been done to obtain prediction intervals with this type of network. This work proposes a methodology to predict crisp values and generate a prediction interval by using a single LSTM network. We show that the proposed methodology is able to achieve state of the art results in the prediction of crisp values as well as in the estimation of the uncertainty interval.

The main contribution of this work is the development of a methodology to generate prediction intervals for LSTM networks. To achieve this, we used a Joint Supervision Loss function, presented in [11], to train the network in order to estimate an upper and a lower bound as well crisp prediction. We tested the proposed approach with real data from two different datasets. The first corresponding to electrical demand of 20 dwellings from a town in the UK, while the second study case consists of net power data from an energy community (microgrid) based on non-conventional energy sources. Furthermore, the results show that the prediction intervals generated by these types of network are more accurate than those generated by classical recurrent neural network (RNN). This is very important since an accurate estimation of the uncertainty is of paramount importance for certain process, such as renewable energy-based generation and electrical demand.

The paper is organized as follows: The neural network structure used to generate the crisp and interval predictions is introduced in Section II. The proposed methodology to train the proposed LSTM network prediction interval is provided in Section III. The results obtained on real case studies for electrical demand and data from an energy community (microgrid) are presented in Section IV. In both cases, the LSTM prediction interval and predicted crisp values are compared to the ones achieved by classical RNN. The conclusions of this study are presented in the last section.

II. Long short-term memory (LSTM) network

Traditional fully connected Neural Networks do not have memory, which is a major problem when trying to estimate future steps of a time series. To solve this problem, Recurrent Neural Networks (RNN) were proposed. These networks can communicate their outputs to their inputs, allowing the information of previous time steps to persist in order to calculate predictions at future time steps. While in theory classical RNN should be able to learn complex, long term patterns in the data, in reality RNN struggle to do so. Long short-term memory networks (LSTM) were proposed in [12] to solve the problems associated with classical RNN. LSTM networks have become the current state of the art in time series prediction and are used to solve a variety of problems such as speech recognition [13], handwriting recognition [14], reinforcement learning [15] and many others. The network's structure is specifically designed to backpropagate the gradient effectively, solving the vanishing gradient problem found on classical recurrent neural network architectures. LSTM networks have also the advantage to have long term memory, which allows them to learn complex patterns found on the dataset. The basic structure for an LSTM cell is shown in Fig. 2.

A LSTM network is composed of several LSTM cells, where the cell state $c(t)$ is designed to let the information flow through the network with only some minor modifications in the form of linear operations, and it can be considered as the memory of the LSTM cell. $h(t)$ represents the hidden state of the cell, which is equivalent to the ouput of the cell $y(t)$. By using the hidden state and cell state of the previous cell, $h(t-1)$ and $c(t-1)$ respectively, in conjunction with the current input $x(t)$, the LSTM cell is able to determine which information to forget or remember. Several variations of this basic structure have been developed, however none of them offer significant improvements over the original implementation [16]. For this paper we use a combination of one or more LSTM networks followed by a fully connected layer. The LSTM network is used to find long term dependencies in the data. The representation obtained by the LSTM layer is then used by a three output fully connected layer to generate the interval (upper and lower bounds) and crisp predictions. The first neuron of the fully connected layer outputs the upper bound of the interval, the second neuron outputs the crisp prediction, while the third neuron outputs the lower prediction interval.

III. Proposed method for developing the prediction interval

While the LSTM architecture certainly increases the accuracy of the crisp prediction when compared to classical RNN, no information about the uncertainty associated with the prediction is given. In this paper, we solve this issue by proposing a methodology to generate a prediction interval for LSTM networks. A prediction interval is composed of an upper and a lower bound in which the prediction output is

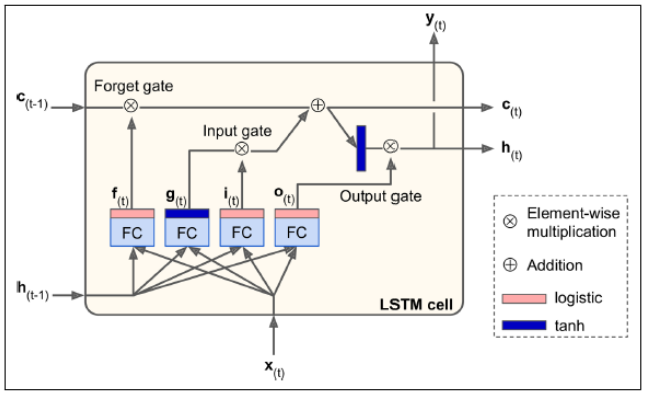

Fig. 2. LSTM Cell. Figure taken from [12]

found with a certain probability [1]. The interval is generated by training the LSTM network with a joint supervision loss function which consists of the combination of the classical RMSE loss function and the interval loss function as presented in [11].

Given this, the network tunable parameters are changed by backpropagating two errors. The first one is given by $e_l = y(k) - \hat{y}_l(k)$ that corresponds to the difference between the model's predicted value $\hat{y}_l(k)$ and the real value $y(k)$ at time k. Therefore, the classical loss function is defined as:

$$L_s^l = \frac{1}{N}\sum_{k=1}^{N} e_l^2(k) \qquad l = \{upper, crisp, lower\} \quad (1)$$

where N corresponds to the amount of data in the training dataset. By applying this classical loss function, it is possible to minimize the difference between the predictions of the network and the real values of the dataset. However, it is impossible to define an interval by using this metric alone. To account for this, the authors in [11] propose a second metric called the interval loss function (L_i). The specific loss for the upper interval (L_i^{upper}) and the lower interval (L_i^{lower}) are defined as follows:

$$L_i^{upper} = \frac{1}{N}\sum_{k=1}^{N}(ReLU(e_{upper}(k)))^2 \quad (2)$$

$$L_i^{lower} = \frac{1}{N}\sum_{k=1}^{N}(ReLU(-e_{lower}(k)))^2 \quad (3)$$

where RELU is the rectified linear unit defined by:

$$RELU(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases} \quad (4)$$

As was proposed in [11], each data point $y(k)$ with a greater value than $L_i^{upper}(k)$ introduces a cost equivalent to the square of the difference between the real data point and its upper interval prediction in accordance with eq. 2. Likewise, any point $y(k)$ with a lower value than $L_i^{lower}(k)$ is penalized in accordance to eq. 3. On the other hand, data points $y(k)$ that belong to the interval introduce no cost (see eq. 4).

By combining the upper and lower cost functions, a penalization for each point $y(k)$ laying outside of the prediction interval can be calculated using the same target data during the training process. The total cost function (L_{total}) is defined as the weighted sum of the classical (see eq. 1) and the interval loss function (see eqs. 2 and 3) for the upper and lower interval respectively:

$$L_{total}^{upper} = L_s^{upper} + \lambda L_i^{upper} \quad (5)$$

$$L_{total}^{lower} = L_s^{lower} + \lambda L_i^{lower} \quad (6)$$

where λ is a tunable parameter that represents the relative importance of proposed classical and interval loss functions. Finally, the crisp output is obtained using only the classical loss function (see eq. 1).

In Fig. 3 a plot of the total loss given by the sum of the classical and the interval loss function can be seen. Fig. 4 presents the RMSE loss function during training process.

These figures present the behavior of a network trained by the proposed loss function. First, the network increases the coverage probability of the network by increasing the interval width. This generates a sharp decrease in the total loss function as the cost from the interval loss function is reduced (see Fig. 3). However, this process comes at the cost of an increment in the RMSE loss function, which increases as the interval loss function decreases (see Fig. 4). Once the desired coverage probability is achieved, the network starts minimizing the *RMSE* which results in a small decrease on the total loss function. We found that iterativly decreasing the learning rate is of most importance to achieve good *RMSE* and *PINAW* results, since this the minimization of the *RMSE* loss function could be considered a fine-tuning of the network.

A. Identification of the Parameters

In the literature, it is common that the training process of the prediction interval models is performed for one-step-ahead prediction, however, the N-steps-ahead prediction criteria are used to assess the quality of a model and when the model is not performing correctly at future predictions horizon, new regressors or other classes of models are selected. Unlike the above mentioned, in this paper, the parameters of the LSTM network are tuned at future steps ahead (N_p) such that the interval contains the largest possible amount of measured data given by the desired coverage probability and so that it will be as narrow as possible at future prediction horizon [1].

The minimization of the interval width is achieved through of the classical loss function (see eq. 1) by the minimization of the error between the interval bounds (upper and lower) and the real data. On the other hand, the interval loss function (L_i) tends to increase the interval width, introducing a trade-off between the number of points that fall into the interval and its width which can be regulated by modifying the parameter λ in eqs. 5 and 6.

In order to find the parameters of the LSTM network, we collected sufficient information to represent the various oper-

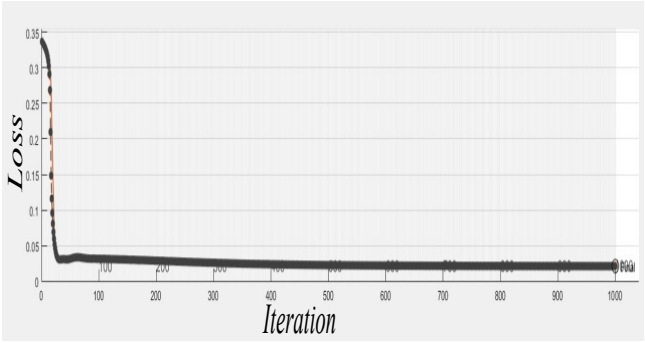

Fig. 3. Total loss function behavior in training process

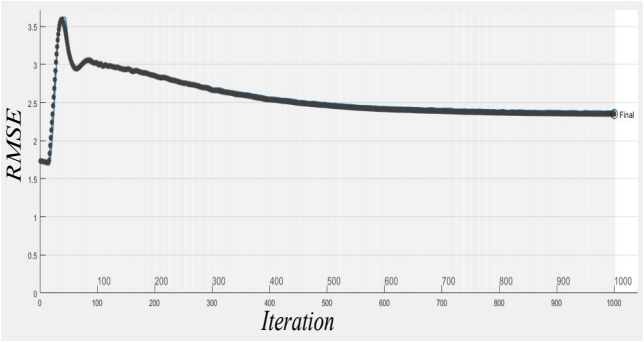

Fig. 4. RMSE loss function behavior in training process

ational points of the process to be modelled. The data is split into training, validation and testing sets. The training dataset is used to estimate the network parameters. The validation dataset is used to evaluate the model generalization capacity. Finally, the test data set is used to evaluate the performance of the obtained model [1].

In order to evaluate the quality of the prediction interval two indicators are used. The first is the prediction interval coverage probability $(PICP)$ that quantify the number of measured values that fall within the interval defined by the model:

$$PICP = \frac{1}{N}\sum_{k=1}^{N} \delta_{k+h} \qquad \forall h = 1, ..., N_p \tag{7}$$

where $\delta_{k+h} = 1$ if $y(k+h) \in [\hat{y}_{lower}(k+h), \hat{y}_{upper}(k+h)]$ otherwise $\delta_{k+h} = 0$ and N_P is the prediction horizon. While the second metric is called prediction interval normalized average width $(PINAW)$ that is used to measure the width of the interval:

$$PINAW = \frac{1}{NR}\sum_{k=1}^{N} \hat{y}_{upper}(k+h) - \hat{y}_{lower}(k+h) \qquad \forall h = 1, ..., N_p \tag{8}$$

where R is the distance between the maximum and minimum values measurements in the data set [1], [11], [17], [18]. Finally, the root mean square error (RMSE) is used as a metric to evaluate the accuracy of the prediction model associated with the crisp value. All indices are evaluated for several prediction horizons with the test dataset.

Fig. 5 presents the algorithm to train the network as was presented in [11]. In this methodology, the λ parameter is iteratively increased. This results in a larger interval width in each iteration as the coverage probability increases. In each iteration, the *PICP* is estimated in accordance to eq. 7. Once the desired coverage probability has been achieved, the algorithm stops increasing the λ parameter. Since this process only changes the relative importance of the interval and the classical loss function, the methodology can be applied to any type of neural network. Then, some extra iterations are performed while keeping the λ parameter constant in order to compensate for the random initialization of the weights. In each iteration the neural network model with the lower interval width is kept and it is used to evaluate the study cases.

IV. Experiment and results

In this paper, comparisons are made between the proposed method based on LSTM network by joint supervision, and the classical recurrent networks based on the work developed in [11]. The output of the RNN used for comparison is defined as:

$$\hat{y}_l(k) = \sum_{j=1}^{L} w_{j,l}^0 \left(\tanh\left(\sum_{i=1}^{p} w_{j,i}^h z_i(k) + b_j^h \right) \right) + b_l^0 \tag{9}$$

with $l = \{upper, crisp, lower\}$ is the upper, crisp and lower output units respectively, L is the number of neurons in the hidden layer and $i = 1, \ldots, p$ is the number of inputs to the network. To obtain the outputs $(upper, crisp, lower)$ of the prediction interval, the parameters $\hat{\theta} = \{w_{j,i}^h, b_j^h, w_{j,l}^o, b_j^o\}$ are tuned through backpropagation method. Therefore, both LSTM network and recurrent neural network (RNN) are trained using the same total loss function defined in eqs. (5) and (6).

In this paper, two case studies are used in order to verify the performance of the proposed methodology for developing prediction intervals using LSTM networks. The first case corresponds to the forecasting up to one day ahead of the demand profile of 20 dwellings from a town in the UK, and

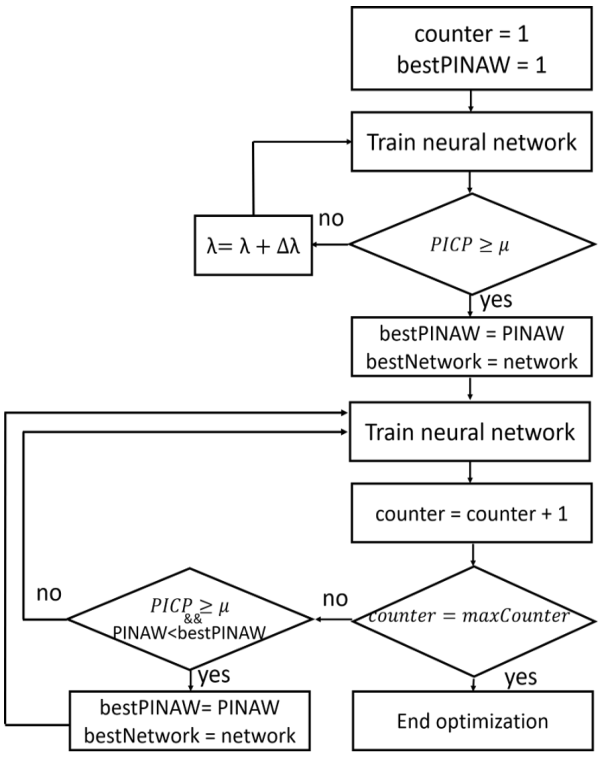

Fig. 5. Iterative training process of joint supervision. Figure taken from [11]

the second case corresponds to the net power from an energy community (microgrid) made up 30 dwellings with a 50% level of photovoltaic power penetration. Next, the results of case studies are presented.

A. Application for load forecasting

We test our proposed methodology on load forecasting data, and we compare it to the results obtained by using classical RNNs presented in [11]. By applying these algorithms the uncertainty of the future load at various steps ahead was estimated. The data originates from 20 residential dwellings from a town in UK and was collected for the summer season of 2008 for 94 days [19], which is divided into 52 days for training, 23 days for validation and 19 days for test data, corresponding to 55%, 25% and 20% of the total dataset, respectively. The maximum electric load is 29.54 kW with a sample time of 15 minutes. Fig. 6 shows five typical days of demand profile of the training dataset from the 20 dwellings used to evaluate the performance of the prediction interval models.

In order to perform load predictions, several structures were evaluated by modifying the number of hidden units and regressors. This was done by an iterative process, were the number of hidden units was increased while the RMSE decreased. Once the RMSE stagnated or started to increase the iteration was stopped and the structure with the best RMSE value was selected. The LSTM network selected for this test is composed of 200 LSTM cells followed by a fully-connected layer. In order to avoid overfitting, dropout is used while training the network. For a fair comparison, the classical RNN uses 200 regressors, and 100 neurons in its hidden layer. In this study, one-step-ahead (15 minutes), one-hour-ahead, and one-day-ahead prediction horizons were evaluated (see Table I).

The results shown in Table I indicate similar performance for interval and crisp prediction of both networks. This may be attributed to the very high variance of the data and the relatively low number of samples to train the network. Both approaches are able to maintain 90% of coverage probability for all prediction horizons considered because the prediction interval models are trained for this desired coverage probability as was explained in previous Section.

TABLE I
PERFORMANCE INDICES

Prediction Horizon	Performance Indexes	Classical RNN	LSTM Network
One step ahead	RMSE (kW)	2.17	2.10
	PINAW (%)	39.56	36.34
	PICP (%)	91.01	91.43
one hour ahead	RMSE (kW)	2.35	2.49
	PINAW (%)	48.98	42.23
	PICP (%)	89.32	90.40
One day ahead	RMSE (kW)	2.31	2.48
	PINAW (%)	47.24	42.20
	PICP (%)	91.17	90.42

This validates the proposed loss function for interval generation as a reliable method to generate prediction intervals for different kind of networks and architectures. Figs. 7 and 8 show the prediction interval at one-day ahead for the RNN and LSTM network, respectively. The prediction interval generated by LSTM network is smooth (see Fig. 8), which is an indicator that the neural network has learned to generalize a representation from the data. By contrast, Fig. 7 which shows the prediction by the classical RNN presents a very stochastic nature, particularly in the upper and lower bound predictions, which can explain the larger interval width (see Table I).

Another important observation is the similar shape of the taken by the crisp, upper and lower predictions. This is probably the result of the network's structure. Since a single neural network is responsible for the prediction of the three outputs, the network needs to be able to learn a representation

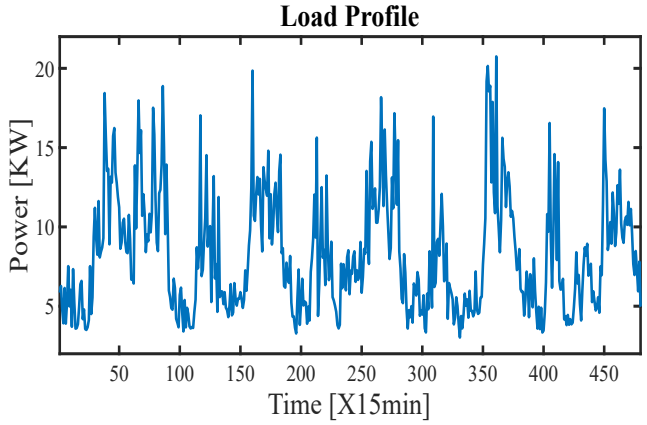

Fig. 6. Five typical days of load profile used during the training process

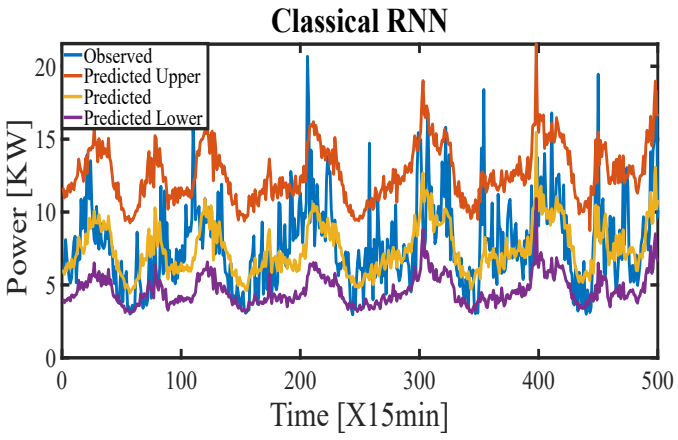

Fig. 7. One-day-ahead Prediction Interval based on recurrent neural network

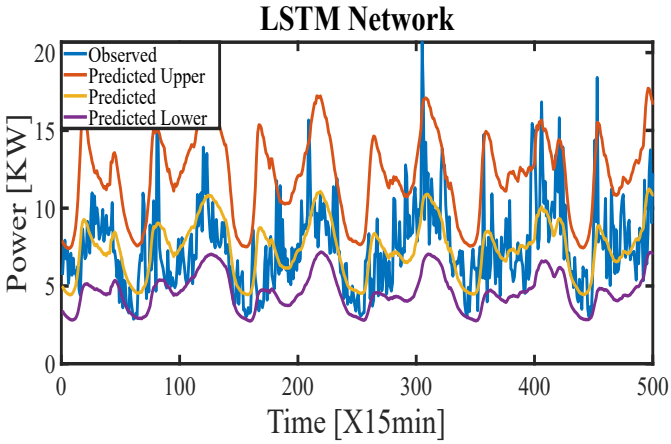

Fig. 8. One-day-ahead Prediction Interval based on LSTM network

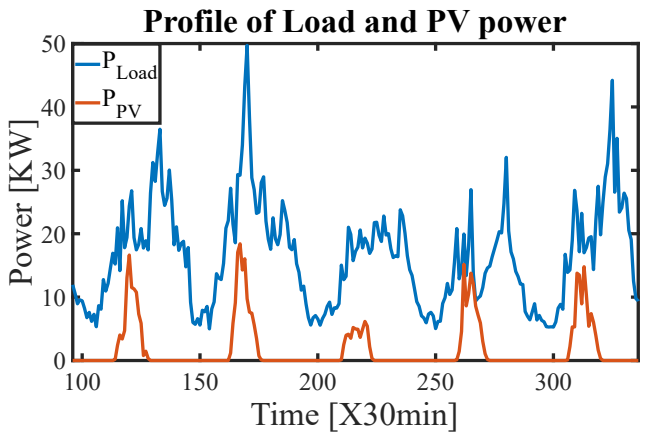

Fig. 9. Profiles of Load and PV power of five days from training dataset

of the data in the hidden layers that satisfies the requirements needed to generate each one of the different outputs with minimal variations. Given this, all outputs have similar trends, with slight variations to specialize in the assigned task (crisp prediction or interval prediction). This could potentially lead to a lower accuracy of the crisp and interval predictions since the network is forced to satisfy three different objectives simultaneously. However, we argue that this could be avoided by simply increasing the number of tunable parameters in the network to form a more complex set of outputs.

B. Application for Energy Community

This section presents the prediction interval models from the net power (P_{net}) of an energy community. In this context, an energy community is considered as a microgrid composed by a group of interconnected loads and distributed energy resources (PV array for this study) within clearly defined electrical boundaries that acts as a single controllable entity with respect to the main grid [20]–[22]. The energy community (microgrid) considered in this study is made up of 30 dwellings with a 50% level of photovoltaic power penetration (i.e. 15 dwellings have a photovoltaic array). Therefore, the active power of both the microgrid consumption (P_L) and photovoltaic array (P_{PV}) with a sample rate of 30 minutes is used to calculate the net power (P_{net}) of the microgrid give by $P_{net}(k) = P_L(k) - P_{PV}(k)$. The winter season data of 2008 from a town in the UK was used [19] to validate the proposed LSTM network and the classical recurrent neural network (RNN) used for comparison of the performance of the prediction interval models. Fig. 9 shows five days of load and photovoltaic power from the microgrid proposed for evaluation of the prediction interval models. This data corresponds to training dataset with a sample time of 30 minutes and the maximum value of the P_{net} is 67.57kW and the minimum value is -45.09kW.

In this case, the LSTM network is composed of 100 LSTM cells followed by a fully-connected layer. Once again overfitting was prevented by using a dropout layer. The classical RNN uses 100 regressors and two hidden layers with 50 and 25 neurons respectively. In this study, one-step-ahead (30 minutes), one-hour-ahead, and one-day-ahead prediction horizons were evaluated. Results presented in Table II show that LSTM networks achieve vastly improved results in terms of RMSE, when compared to classical neural networks. This may be explained by the greater ability of LSTM networks to find long term dependencies in the data allowing it to perform more accurate predictions. As with previous datasets, both network architectures are able to consistently maintain the desired 90% coverage probability, which once again validates the proposed interval loss function for interval prediction.

However, the LSTM network archives smaller *PINAW* values than its classical counterpart for an overall better prediction interval. This may once again be the result of the long term memory of the LSTM network. As in the previous experiment, the prediction for the RNN, shown in Fig. 10 is more stochastic that it's LSTM counterpart shown in Fig. 11, which indicates that the LSTM network is better at generalizing the data. The differences found in the quality of the generated interval can

TABLE II
PERFORMANCE INDICES

Prediction Horizon	Performance Indexes	Classical RNN	LSTM Network
One step ahead	RMSE (kW)	9.46	5.60
	PINAW (%)	28.77	26.32
	PICP (%)	90.36	90.78
One hour ahead	RMSE (kW)	9.56	7.47
	PINAW (%)	35.37	34.94
	PICP (%)	89.81	90.77
One day ahead	RMSE (kW)	13.50	8.10
	PINAW (%)	38.73	37.77
	PICP (%)	88.66	89.21

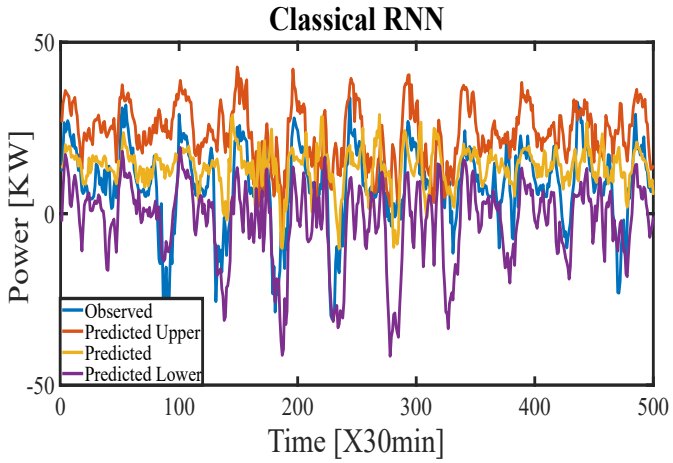

Fig. 10. One-day-ahead Prediction Interval based on recurrent neural network

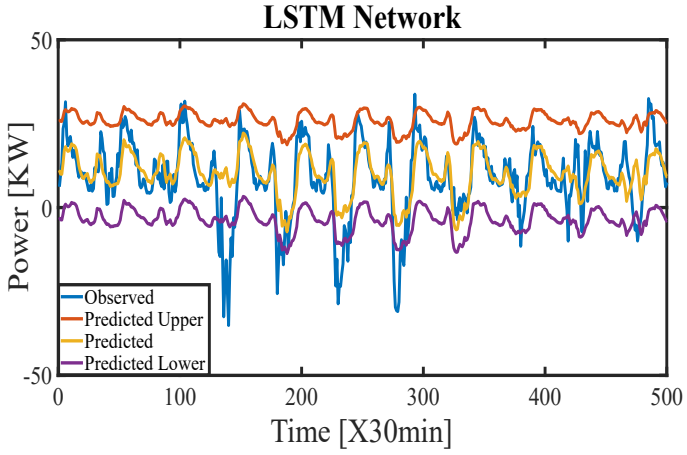

Fig. 11. One-day-ahead Prediction Interval based on LSTM network

also be explained by which data the network leaves outside of the interval. While the RNN has no clear criteria of which data is inside or outside the interval, the LSTM network has generated a lower bound in which spikes are left outside of the prediction interval. This allows for an interval with little variation in it's values while maintaining the coverage probability in a very efficient fashion.

C. Generated data

In order to better compare the performance of classical and LSTM networks for interval predictions we test both networks using generated data in the form of the modified Chen-series. The original Chen series was presented in [8] and it corresponds to a nonlinear, stochastic time-series. We modified the classical Chen series [11] in order to incorporate state dependant noise:

$$\begin{aligned} y(k) =&(0.8 - 0.5 \cdot exp(-y^2(k-1))) \cdot y(k-1) - \\ &(0.3 + 0.9 \cdot exp(-y^2(k-1))) \cdot y(k-2) + \\ &u(k-1) + 0.2 \cdot u(k-2) + \\ &0.1 \cdot u(k-1) \cdot u(k-2) + e(k) \end{aligned} \tag{10}$$

By using generated data, we can increase the number of samples as necessary. The greater sample number help both networks to generalize the problem and find patterns in noisy data. Given this, theoretically, LSTM networks should achieve better results than their classical counterparts, since the limits on the accuracy of both networks is given by their capacity to learn the function mapping input to output rather than by an sub-optimal tuning of the networks parameters given insuficient data. Again, the LSTM network's number of hidden units was selected by following an iterative process in order to minimize the network's RMSE. The optimal number of neurons was found to be 15 for this network. For a fair comparison the same number of neurons was chosen for the classical recurrent neural network. Results are reported for one, eight and sixteen steps ahead.

Fig. 12) and 13) show that both networks are able to learn the state dependant noise in order to minimize the PINAW while maintaining the PICP. However, the results in Table III show that LSTM network outperforms classical approaches in terms of PINAW. This can be attributed to the larger amount of available data as well as the lower variance of this test series, which makes easier for the networks to find and adapt to patterns in the data.

V. Conclusions

In this paper, we presented a methodology to generate prediction intervals for LSTM networks by using a Joint Supervision Loss Function. The improved capacities of LSTM networks to model complex data patterns translates in highly accurate crisp and interval predictions. We tested our approach with real data from two distinct data sets and found that LSTM networks consistently achieve better *PINAW* and *RMSE* metrics than those produced by classical RNN while maintaining similar coverage probabilities. This translates in a better overall estimation of the uncertainty. The proposed methodology simplifies the process of generating a prediction interval. While most of the alternative approaches consider the prediction of the crisp and interval values as two separate

TABLE III
Performance Indices

Prediction Horizon	Performance Indexes	Classical RNN	LSTM Network
One step ahead	RMSE	0.47	0.40
	PINAW (%)	14.40	11.68
	PICP (%)	92.28	92.23
eight steps ahead	RMSE	0.62	0.55
	PINAW (%)	15.80	13.50
	PICP (%)	89.09	90.45
sixteen steps ahead	RMSE	0.60	0.61
	PINAW (%)	15.80	13.50
	PICP (%)	92.02	89.69

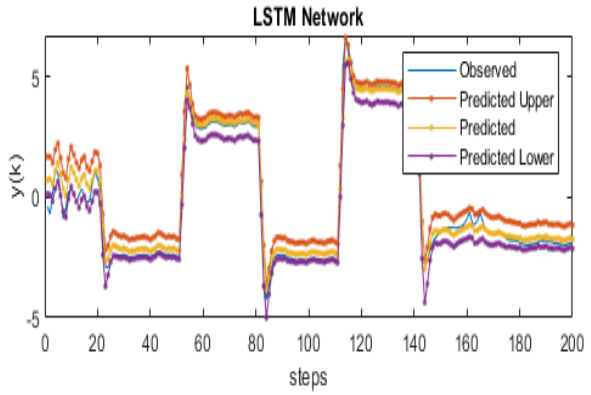

Fig. 12. One-step-ahead Prediction Interval based on LSTM neural network

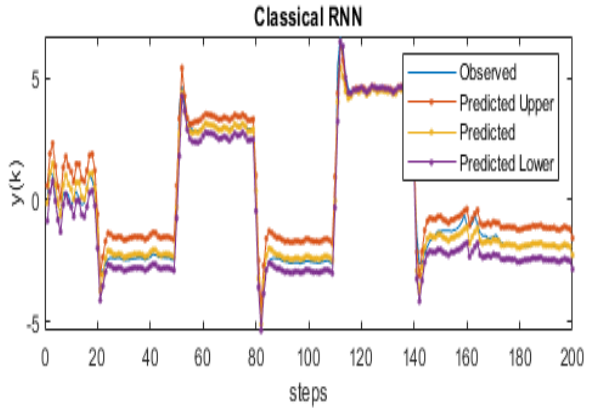

Fig. 13. One-step-ahead Prediction Interval based on recurrent neural network

problems, our approach is able to generate both the crisp and interval predictions using a single neural network with three outputs. Additionally, our model is trained by using standard backpropagation in a supervised end-to-end optimization process which is faster and simpler than alternative approaches, which suffer from slow convergence, a great number of tunable hyper-parameters and an over-reliance on heuristics to form the interval predictions.

This work also proves the flexibility of the proposed loss function, since it can be used to train multiple network architectures to generate prediction intervals with only minor modifications.

References

[1] L. G. Marín, N. Cruz, D. Sáez, M. Sumner, and A. Núñez, "Prediction interval methodology based on fuzzy numbers and its extension to fuzzy systems and neural networks," *Expert Systems with Applications*, vol. 119, pp. 128–141, 2019.

[2] H. M. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, "Neural network-based uncertainty quantification: A survey of methodologies and applications," *IEEE Access*, vol. 6, pp. 36 218–36 234, 2018.

[3] L. G. Marín, F. Valencia, and D. Sáez, "Prediction interval based on type-2 fuzzy systems for wind power generation and loads in microgrid control design," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 2016, pp. 328–335.

[4] F. Valencia, J. Collado, D. Sáez, and L. G. Marín, "Robust Energy Management System for a Microgrid Based on a Fuzzy Prediction Interval Model," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1486–1494, May 2016.

[5] Y. Xiang, J. Liu, and Y. Liu, "Robust energy management of microgrid with uncertain renewable generation and load," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 1034–1043, March 2016.

[6] F. Rodrigues, I. Markou, and F. C. Pereira, "Combining time-series and textual data for taxi demand prediction in event areas: a deep learning approach," *Information Fusion*, pp. 1–20, 2018.

[7] H. Salehinejad, J. Baarbe, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," *CoRR*, vol. abs/1801.01078, 2018.

[8] J. Chen, G.-Q. Zeng, W. Zhou, W. Du, and K.-D. Lu, "Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization," *Energy Conversion and Management*, vol. 165, pp. 681–695, 2018.

[9] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, 2016, pp. 1050–1059.

[10] L. Zhu and N. Laptev, "Deep and confident prediction for time series at uber," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017, pp. 103–110.

[11] N. Cruz and L. G. Marín and D. Sáez, "Neural network prediction interval based on joint supervision," in *2018 International Joint Conference on Neural Networks (IJCNN)*, July 2018, pp. 1–8.

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

[13] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, K. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," *CoRR*, vol. abs/1712.01769, 2017. [Online]. Available: http://arxiv.org/abs/1712.01769

[14] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in Neural Information Processing Systems 21*. Curran Associates, Inc., 2009, pp. 545–552. [Online]. Available: http://papers.nips.cc/paper/3449-offline-handwriting-recognition-with-multidimensional-recurrent-neural-networks.pdf

[15] M. J. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," *CoRR*, vol. abs/1507.06527, 2015. [Online]. Available: http://arxiv.org/abs/1507.06527

[16] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *CoRR*, vol. abs/1503.04069, 2015. [Online]. Available: http://arxiv.org/abs/1503.04069

[17] N. Shrivastava, K. Lohia, and B. K. Panigrahi, "A multiobjective framework for wind speed prediction interval forecasts," *Renewable Energy*, vol. 87, no. Part 2, pp. 903–910, 2016.

[18] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower upper bound estimation method for construction of neural network-based prediction intervals," *IEEE Transactions on Neural Networks*, vol. 22, no. 3, pp. 337–346, March 2011.

[19] I. Richardson and M. Thomson, "One-Minute Resolution Domestic Electricity Use Data, 2008-2009 [computer file]," *Colchester, Essex: UK Data Archive [distributor]*, 2010.

[20] S. Parhizi, H. Lotfi, A. Khodaei, and S. Bahramirad, "State of the Art in Research on Microgrids: A Review," *IEEE Access*, vol. 3, pp. 890–925, 2015.

[21] N. Hatziargyriou, H. Asano, R. Iravani, and C. Marnay, "Microgrids: An Overview of Ongoing Research, Development, and Demonstration Projects," *IEEE Power & Energy Magazine*, vol. July/August, pp. 78–94, July 2007.

[22] Department of Energy office of Electricity Delivery and Energy Reliability, "Summary Report : 2012 DOE Microgrid Workshop," Office of Electricity Delivery and Energy Reliability Smart Grid R&D Program, Tech. Rep., 2012.