#560: Intuitive lower upper bound estimation for short-term wind speed interval prediction

Adnan SAEED[1], Chaoshun LI[1], Qiannan ZHU[1], Sana ANWAR [2]

[1] *Huazhong University of Science and technology, China, adnansaeed@hust.edu.cn, csli@hust.edu.cn, qnzhu@hust.edu.cn*
[2] *Aligarh Muslim University, India, anwar.sana9@gmail.com*

Abstract: Estimating the uncertainty associated with wind speed forecasts is crucial for evaluating wind farms' power quality and operation. Prediction Intervals can communicate the uncertainty by prescribing a range of bounds within which the real value is expected to lie. While many approaches for PI generation exist, direct PI generation approach is usually preferred due to its simplicity. In this article we proposed a novel loss function to train neural networks to generate prediction intervals directly at its output. Building upon the fundamentals of the two state-of-the-art approaches; lower upper bound estimation (LUBE), and quality driven (QD), we proposed intuitive lower upper bound estimation loss (ILUBE). Utilizing the same objective of heuristic optimization of the two PI characteristics, coverage and interval size between, weapplied a dynamic approach to obtain each of these two characteristics. The proposed ILUBE loss shows its strength in computational experiments involving eight different datasets for two different wind fields. The PIs generated by the proposed approach are tighter while covering a greater number of observations, in comparison to both LUBE and QD. Thus, by improving the PI quality, while keeping the essence of simplicity of the direct PI generation approaches, the proposed ILUBE loss emerged as a strong contender in this category.

Keywords: Prediction Intervals, Wind Speed Prediction, Lower Upper Bound Estimation, Direct Prediction Interval Estimation

1. INTRODUCTION

In recent decades, there have been huge technological advancements and rapid urbanization across the globe. This sheer pace of growth brings challenges like growing energy demand and carbon neutrality, leading to a surge in the global demand for clean and sustainable energy solutions (Summerfield-Ryan and Park, 2023). Wind energy, with its abundant availability, has emerged as a frontrunner in this transition. However, its inherent volatility poses a challenge in harnessingits full potential. Traditionally, wind power forecasts are utilized to make informed decisions to mitigate the effects of intermittency in wind power dispatch. Since wind turbines are provided with associated power curves by the manufacturers,the focus has shifted towards using wind speed forecasts to obtain wind power estimates (Wu and Hong, 2007). Neighbouring wind farms often share similar wind patterns despite using different turbines, this approach improves efficiency by leveraging the common wind speed forecast for multiple turbines (Zhu and Genton, 2012). Therefore, improving the accuracy of wind speed forecasts is crucial for maximizing wind power integration into the power grid and several studies have been performed to obtain accurate wind speed forecasts (Ren, Suganthan and Srikanth, 2015; Jiangand Huang, 2017). However, these studies focus on improving the deterministic forecasts, which are often unreliable dueto the fact that they give single values which may have a large downside in case of large deviation from the true value.

Quantifying the uncertainty in predictions may provide valuable flexibility for the decision makers. Therefore, estimating prediction intervals (PIs) is often preferred over deterministic forecasts (Ma, Huang and Shi, 2023; Saeed, Li and Gan, 2024). PIs provide a range of values within which the true value is likely to fall and are therefore more reliable than singl e deterministic value as shown in (Figure 1).

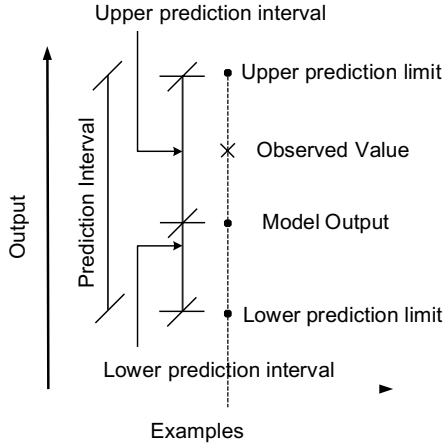

Figure 1: Visualisation of a prediction interval

There are two main approaches to construct PIs for any time series. The first approach constructs the PIs by approximating a distribution around the point forecasts. Various methods of this approach exist, including bootstrapping (Chen et al., 2012), Bayesian modelling (Miranda and Dunn, 2006), and estimating mean and variance (MVE) (Yao, Zeng and Lian, 2017). However, real-world wind speed data can be complex and making such assumptions about its distribution may be impractical. The second approach doesn't assume any data distributions, instead it generates the PIs directly by training a regression network using a loss function specially tailored to construct PIs. The fundamental idea in the direct PI generation approach is that the PIs should capture some desired portion of observations and should not be too wide. Since widening the PIs may satisfy the former condition but larger interval width would not communicate meaningful results for real world applications. Lower upper bound estimation (LUBE) (Khosravi et al., 2011), the first in this approach defined a loss function defined (Equation 62)

Equation 62: lower upper bound estimation loss function.

$$\mathcal{L}_{LUBE} = \left({MPIW}/{r}\right)\left(1 + exp\left(\lambda\, max(0, (1-\alpha) - \mathrm{C})\right)\right)$$

Equation 63: Mean Prediction Interval Width

$$MPIW = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_{ui} - \hat{y}_{li})$$

Equation 64: Actual coverage

$$\mathrm{C} = \sum_{i=1}^{N} k_i$$

Where:

- MPIW = Mean Prediction Interval Width, defined in (
- Equation 63)
- C = actual number of observations covered by the PIs, defined in (
- Equation 64)

- $k_i = \begin{cases} 1, & (\hat{y}_{li} \leq y_i \leq \hat{y}_{ui}) \\ 0, & \text{else} \end{cases}$
- α = PI confidence expressed as to capture (1-α) portion of observations
- r = range of the response variable
- λ = parameter for the trade-off between interval width and coverage
- y_i = true data value
- $\hat{y}_{Li}$ and $\hat{y}_{Ui}$ = lower bound and upper bound of the corresponding PI

The f_{LUBE} loss function constructs the PIs through heuristically balancing the interval width and coverage. In (Equation 1) the interval width is minimized by the MPIW term while the exponential term maximizes the coverage as the probability $(Pr(\hat{y}_{Li} \leq y_i \leq \hat{y}_{Ui}) > (1-\alpha))$ to cover the desired $(1-\alpha)$ number of observations. The first drawback with this loss function is the non-differentiability, due to which it requires non-gradient based algorithms to train the model. In its original form it is proposed with simulated annealing and research have put much effort in trying different non-gradient based training methods like gravitational search algorithm (Lian *et al.*, 2016), particle swarm optimization (Galván *et al.*, 2017), and artificial bee colony (Shen, Wang and Chen, 2018). This is inconvenient since gradient descent is the standard training method used by all modern neural network APIs. Another drawback is that the loss function does not admit any theoretical guarantee over the coverage.

Quality Driven (QD) loss function (Pearce *et al.*, 2018) brings some important modifications in LUBE framework. They argued that usage of MPIW causes artificial shrinking of the already invalid PIs. Therefore, to consider only the widths of the intervals that contain the true value, they replaced MPIW by capturing MPIW. Moreover, ditching the idea of maximizing coverage by exponential term as expressed in (Equation 1), they expressed the number of captured points using probability mass function (PMF) of binomial distribution which they finally approximated to normal distribution considering it as mild assumption for sufficiently large value of n i.e. $\mathrm{C} \sim \mathrm{Binomial}\,(n, (1-\alpha)) \approx \mathrm{N}(n(1-\alpha), n\alpha(1-\alpha))$ as in (Equation 4)

Equation 65: probability mass function (PMF) of binomial distribution

$$PMF(\mathrm{C}) = \frac{1}{\sqrt{2\pi n\alpha(1-\alpha)}} exp - \frac{\left(\mathrm{C} - \left(n(1-\alpha)\right)\right)^2}{2n\alpha(1-\alpha)}$$

To fit it to the nature of loss function, they converted it to minimize its negative log-likelihood to obtain the loss expression as:

Equation 66: Quality Driven loss function

$$\mathcal{L}_{QD} = MPIW_{capt} + \lambda \frac{n}{\alpha(1-\alpha)} max(0, (1-\alpha) - \mathrm{C})^2$$

The QD loss, though theoretically more advanced than LUBE, however, it does bring some constraints and limitations. For their argument of normal assumption to hold true, the loss is constrained to use larger batch sizes. In addition, although the loss function is defined initially without imposing any data distribution assumptions, but the final form is reduced to minimizing the nll of normal distribution. So, it may not be wrong to say that it indirectly brings the context of some data distribution. Lastly, it is worth noting that QD succeeds in mitigating the non-differentiability of LUBE and can be used with usual gradient descent. But the limitation of assurance over coverage still remains untouched.

The present study aims to build upon the limitations of these loss functions and propose a novel loss function in the realm of direct PI generation category that could train neural networks to directly construct PIs with some degree of assurance of achieving the desired coverage while avoiding excessively wide intervals. Rethinking the concept of coverage and interval width in a dynamic sense, we proposed a novel loss function and dubbed it as Intuitive Lower Upper Bound Estimation (ILUBE). The loss function dynamically adjusts the PI width based on coverage for better capturing of true values motivating the model to construct tighter intervals with high coverage. Apart from PI construction method, the choice of regression architecture also plays a significant role in the performance of a PI generation model. The selection of regression architecture has seen dramatic variations over the years. With statistical techniq ues like ARMA (Erdem and Shi, 2011) and ARIMA (Liu, Tian and Li, 2015) as the preferred regression mechanism in earlier studies, the choice gradually shifted to machine learning-based methods like extreme learning machines (Chen *et al.*, 2012), and support vector regression (Ren, Suganthan and Srikanth, 2016) due to their advantage of non-linear modelling capabilities over the statistical methods. Recently, the focus has shifted to using deep learning-based methods due to their advantage of automatic extraction of innate hidden features over machine learning methods where we need to pre-process data for hand-crafted features to improve the model performance. Deep learning methods like recurrent neural network (RNN) and its variations like (long short-term memory) LSTM and gated recurrent unit (GRU) posses special ability to incorporate temporal information thus becoming important choice for modelling sequential data (Li *et al.*, 2020; Saeed *et al.*, 2020; Peng *et al.*, 2021). Therefore, the present study adopts LSTM as the regression architecture due to its advantage of keeping long term memory while handling vanishing and exploding gradients.

2. METHODOLOGY

In this section, we first derive the ILUBE loss function discussing the importance of each component in comparison to LUBE and QD approaches, and later integrate it into LSTM to construct wind speed PIs.

2.1. ILUBE Derivation

Like LUBE and QD we also follow the fundamental principle of maintaining a heuristic balance between coverage and interval size. The difference lies in the approach of calculating the coverage and interval width. Our approach is to look for dynamic calculations for both PI size and coverage. Let's explore each of these components individually.

Coverage: Both LUBE and QD, looks for maximizing the coverage by comparing actual coverage to the desired coverage, exponential term in (Equation 1) for LUBE and maximum term in (Equation 5) for QD. We strive to maximize the coverage dynamically by calibrating each PI for non-departure of true value from the interval bounds. We define a calibration function to punish the PIs which fail to cover the true value. The applied penalty is proportional to how far the true value departs from the corresponding interval. For a PI with confidence $(1-\alpha)\times 100\%$ PI, the calibration function is expressed as:

Equation 67: Calibration function for an individual PI

$$l_{\mathbb{C}} = \begin{cases} (\hat{y}_{li} - y_i)/(\alpha/2), & if \quad y_i < \hat{y}_{li} \\ 0, & if \quad y_i \in [\hat{y}_{li}, \hat{y}_{ui}] \\ (y_i - \hat{y}_{ui})/(\alpha/2), & if \quad y_i > \hat{y}_{ui} \end{cases}$$

Interval Size: Both LUBE and QD incorporate terms related to the average PI width into the loss function for its minimization. The difference is that LUBE takes a simple approach, directly adding the Mean Prediction Interval Width (MPIW) to the loss function as shown in (Equation 1). QD refines this by considering only the width of PIs that actually capture the true values ($MPIW_{capt}$) to avoid artificial shrinking of invalid PIs as shown in (Equation 5). We took it a step further, giving a different treatment to interval width. Instead of ignoring the invalid PIs, we defined an intelligent function to penalize the width of PIs for which true values escape the PI bounds. Utilizing the coverage of batch forecasts, we incorporated the context of coverage in calculation of the average interval width. The idea is to design a scaling factor that adapts PI width dynamically based on the actual coverage. The first step towards the design of the dynamic scaling factor is to calculate the actual coverage for which (Equation 3) could be utilized. But to avoid non-differentiability, the definition of k_i is modifiedaccording to (Pearce *et al.*, 2018) using sigmoid function. The actual coverage therefore can be expressed as:

Equation 68: Soft form of k_i for smooth probability calculations

$$k_{si} = \sigma\big(s \cdot (y - \hat{y}_{li})\big) \odot \sigma\big(s \cdot (\hat{y}_{ui} - y)\big)$$

Equation 69: Soft form of actual coverage achieved by PIs

$$\mathbb{C}_s = \sum_{i=1}^{N} k_{si}$$

Where:

- s = a positive valued softening factor
- σ = sigmoid function
- ⊙ = element-wise product

Comparing the actual coverage with the target coverage the dynamic scaling factor is calculated using (Equation 9) and the scaled interval width is expressed as (Equation 10)

Equation 70: dynamic scaling factor

$$l_{sf} = \begin{cases} 1 + \big((1-\alpha) - \mathbb{C}_s\big)/(1-\alpha), & \mathbb{C}_s < (1-\alpha) \\ 1, & otherwise \end{cases}$$

Equation 71: Scaled Mean PI width

$$MPIW_{scaled} = \frac{1}{N}\sum_{i=1}^{N} l_{sf}(\hat{y}_{ui} - \hat{y}_{li})$$

Combining (Equation 6) and (Equation 10) we express our ILUBE loss as:

Equation 72: ILUBE loss function

$$\mathcal{L}_{ILUBE} = \frac{1}{N}\left(\sum_{i=1}^{N} l_{sf}(\hat{y}_{ui} - \hat{y}_{li}) + \sum_{i=1}^{N} l_{\mathbb{C}}\right)$$

2.2. Advantages of ILUBE over LUBE and QD

The derived loss function in (Equation 11) may have the same objective as of LUBE and QD, but in comparison of approach the differences are significant, and are summarised as follows

i) While coverage term was derived based on likelihood of normal distribution in QD and the exponential term in LUBE was selected qualitatively, the approach of ILUBE towards coverage is intuitive, calibrating each PI for non-departure of true value from the interval bounds.
ii) Purely distribution free and not rely on any indirect likelihood of any distribution like QD.
iii) Unlike LUBE and QD, which do not provide theoretical guarantee for the coverage of PIs, ILUBE prioritizes achieving the desired coverage while avoiding wider intervals. Through its dynamic scaling mechanism which works on comparison of the actual coverage to the target coverage level. As shown in (Equation 9), if the actual coverage is lower than desired level, it scales the PI width motivating the model to refine its predictions and improve the coverage. Conversely, if the coverage already meets the target, the scaling factor remains at 1 to avoid unnecessarywidening of the PIs.

2.3. Integrating ILUBE into LSTM framework for PI generation model

To generate PIs, we adopted LSTM as the regression architecture and trained it using the proposed ILUBE loss to generate the wind speed prediction intervals. The framework of the LSTM-based architecture trained using the proposed ILUBE loss is shown below in (Figure 2). After pre-processing, the wind speed data is passed as input with shape (batch size, time steps, features) to the LSTM layer. In forecasting tasks, the temporal characteristics are really important. LSTMs excel in this domain by using memory cells and special gates as described in (Equation 12) to (Equation 17). These gates act as intelligent filters, deciding what past information is crucial to remember and what can be forgotten. By effectively managing memory, LSTMs can learn from short and long dependencies and make accurate predictions.

Equation 73: Forget gate $$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right)$$

Equation 74: Input gate $$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

Equation 75: Output gate $$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_f\right)$$

Equation 76: Cell candidate vale $$\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c)$$

Equation 77: cell state $$c_t = \sigma(\tilde{c}_t \otimes i_t + c_{t-1} \otimes f_t)$$

Equation 78: LSTM output $$h_t = o_t \otimes tanh(c_t)$$

Where:

- x_t = input to LSTM at time step t
- W, U = weight matrices
- b = bias
- ⊗ = Hadamard product

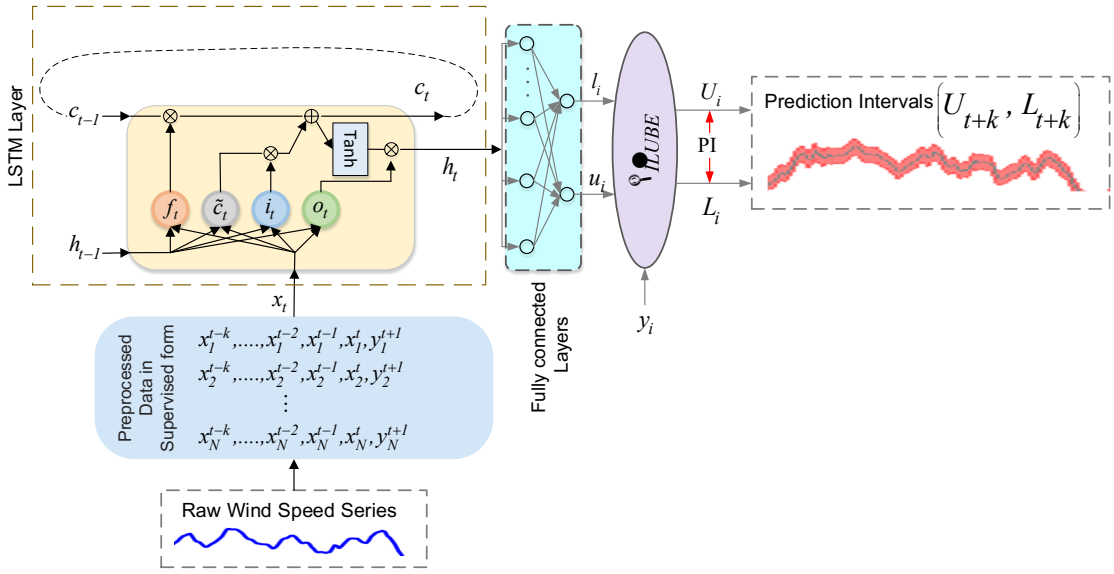

Figure 2: LSTM-based PI generation model trained using ILUBE loss function

The features extracted by the LSTM layer are passed on to a fully connected layer which further applies nonlinear computations (ReLU activation) transforming it to a dense representation which is finally converted into two bound outputs by the output layer having two neurons with linear activation. To prevent intermixing of the bounds, the output layer is initialized with constant values 1 and -1 to ensure that the upper bound is always greater than lower bound. These bounds optimized using ILUBE loss function through iterative weight update process of gradient descent using Adam optimizer.

3. EXPERIMENTAL DATA AND EVALUATION INDICES

This section presents the details of the datasets considered for experiments and the evaluation indices adopted for comparing the results of our approach with earlier studies.

3.1. Datasets

This study employs datasets from the Wind Integration National Dataset (WIND) Toolkit (Draxl *et al.*, 2015), developed by the National Renewable Energy Laboratory (NREL). To evaluate the model's performance across diverse data types we randomly selected one on-shore location and one off-shore location and collected data for the two distinct wind fields for the years 2011 and 2012 from the NREL website. The offshore wind field in Lake Huron (site ID 110197) is located at 45.07209 latitude and -83.111816 longitude while the onshore wind field in Pennsylvania (site ID 71764) is situated at 40.7147 latitude and -78.3109 longitude. Wind speeds were recorded at a hub height of 100 meters for both locations.

To evaluate the generalization capabilities of the competing models, we created four distinct datasets by dividing the data for each location into four sub-series using a scrolling window format. This approach was based on the understanding that wind characteristics typically vary across the four seasons of the year. Consequently, we aimed to assess how the developed models perform in predicting wind characteristics across different seasons. We first obtained two years of data (2011 and 2012) for each location. The 2012 data was then divided into four sets of three months each (January-March, April-June, July-September, and October-December), resulting in four test datasets roughly corresponding to different seasons. For the training datasets, we used data from both 2011 and 2012, ensuring that each training dataset contained 12 months of data preceding the corresponding test dataset. For instance, the first test dataset, covering January 1, 2012, to March 31, 2012, had a corresponding training dataset from January 1, 2011, to December 31, 2011. Similarly, the The second test dataset, covering April 1, 2012, to June 30, 2012, had a training dataset from April 1, 2011, to March 31, 2012. This process was repeated for the third and fourth datasets, with each dataset containing one year of training data followed by three months of test data. This data splitting method can be viewed as a form of cross-validation, ensuring the model's generalization capability. The four sub-series were named "spring," "summer," "autumn," and "winter" for readability, although the division was not strictly based on external temperatures. The datasets are graphically represented in (Figure 3), where the training data is shown in blue and the test data in red for each of the eight datasets. Significant fluctuations in wind speed are clearly visible across all datasets. To validate these visual observations, a statistical analysis was conducted. (Table 1) provides a statistical summary of the training and test data for each dataset. (Figure 4) presents the correlation coefficients between the different training datasets for each location. The low correlation coefficients indicate a very weak correlation among the training datasets for each location. Additionally, (Table 1) reveals that the statistical parameters for the test data vary considerably among the four datasets for each location. Consequently, it can be inferred that these datasets are suitable for evaluating the generalization capabilities of different models.

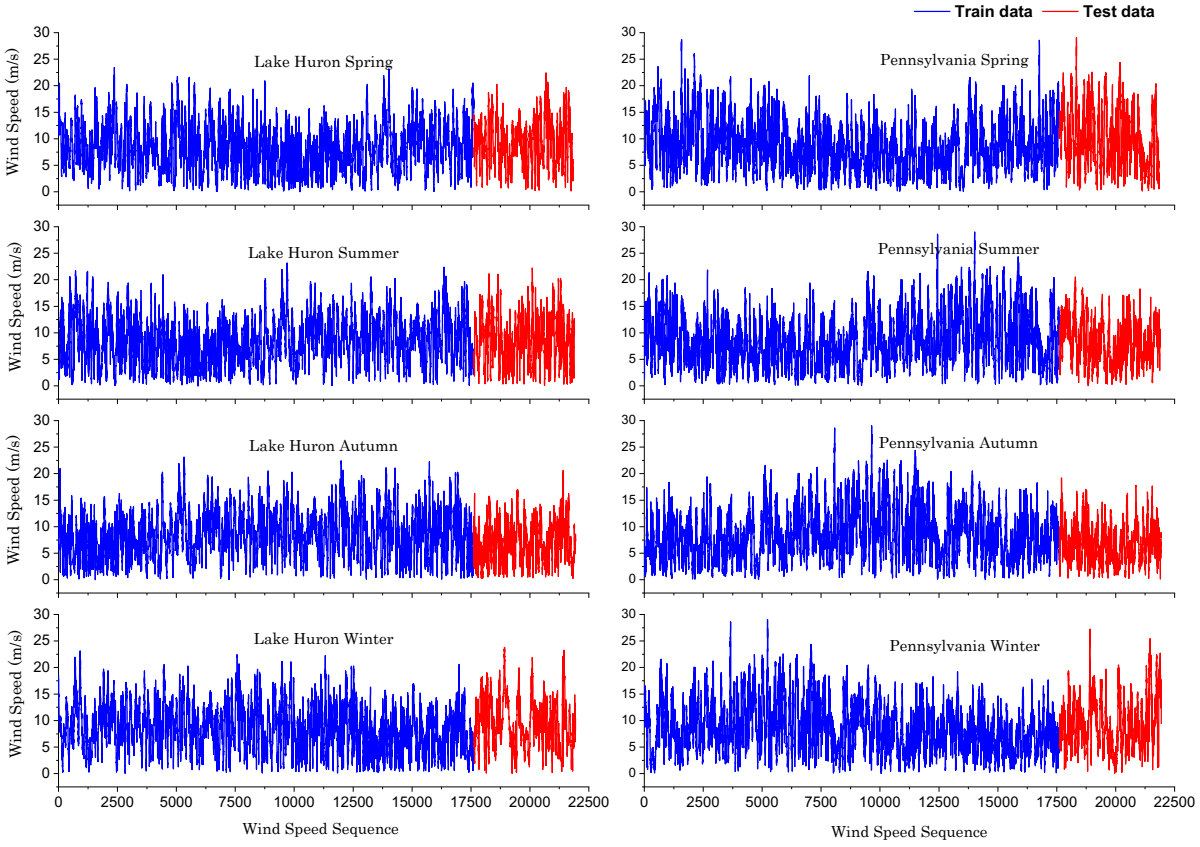

Figure 23: Visualisation of the different wind speed datasets

Table 23: Statistical description of the eight wind speed datasets

Location	Dataset	Count		Min		Max		Mean		Std. Dev	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Lake Huron	Spring	17519	4225	0.02	0.15	23.42	22.39	8.43	9.30	4.10	4.08
	Summer	17568	4273	0.02	0.14	23.14	22.23	8.55	8.81	4.15	4.19
	Autumn	17616	4321	0.02	0.26	23.14	20.59	8.57	7.22	4.09	3.52
	Winter	17616	4321	0.08	0.08	23.14	23.78	8.58	9.59	4.07	4.43
Pennsylvania	Spring	17519	4225	0.13	0.15	28.71	29.02	8.53	10.03	4.41	5.24
	Summer	17568	4273	0.13	0.06	29.03	20.49	8.51	8.11	4.52	3.79
	Autumn	17616	4321	0.05	0.17	29.03	19.21	8.45	6.47	4.48	3.33
	Winter	17616	4321	0.05	0.10	29.03	27.20	8.41	9.26	4.49	4.82

Lake Huron	Spring	Summer	Autumn	Winter
Spring	1.0000			
Summer	-0.0289	1.0000		
Autumn	0.0201	0.0345	1.0000	
Winter	0.0683	0.0520	0.0133	1.0000

(a)

Pennsylvania	Spring	Summer	Autumn	Winter
Spring	1.0000			
Summer	-0.0054	1.0000		
Autumn	-0.1031	-0.0053	1.0000	
Winter	0.0759	-0.0618	0.0547	1.0000

(b)

Figure 4: Correlation matrix plots between different training datasets for (a) Lake Huron, and (b) Pennsylvania datasets

3.2. Evaluation Indices

PIs are mainly evaluated based on coverage and interval size. This study adopted the standard indices for coverage and interval size called PI coverage probability (PICP) and PI normalized root-mean-square width (PINRW). These indices, however, are contradicting in nature and neither can describe the quality of PIs single handedly. Therefore, another evaluation index called coverage width criterion (CW C) is adopted. CWC effectively maintains a balance between coverage and PI width. The mathematical definitions of all these evaluation indices are described below:

Equation 79: Prediction Interval Coverage Probability

$$PICP = \frac{1}{N}\sum_{i=1}^{N} k_i \, , where \;\; k_i = \begin{cases} 1, & y_i \in [L_i, U_i] \\ 0, & y_i \notin [L_i, U_i] \end{cases}$$

Equation 80: PI normalized root-mean-square width

$$PINRW = \frac{1}{R}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(U_i - L_i)^2}$$

Equation 81: Coverage Width Criterion

$$CWC = \begin{cases} \beta \cdot PINAW, & PICP > \mu \\ (\rho + \beta * PINAW) \cdot \left(1 + exp\big(-\eta(PICP - \mu)\big)\right), & PICP < \mu \end{cases}$$

Where:

- μ = PI normal confidence (PINC)
- ρ, β, η = governing hyperparameters
- PINAW = PI normalized average width expressed as $\frac{1}{nR}\sum_{i=1}^{n}(U_i - L_i)$

4. RESULTS

4.1. Comparative models and Parameters selection

To make effective comparisons we considered studies involving bootstrapping method (Chen *et al.*, 2012), and LUBE method (Quan, Srinivasan and Khosravi, 2014). The parameters for training the models for these studies were kept as per the description present in these studies. In addition, for comprehensive comparison we trained LSTM-based models using the QD approach and our proposed ILUBE loss. The parameters for the LSTM were identified through trial and error and the same parameters were kept for both the LSTM-based models for unbiased results. The parameters for different loss expressions and models are presented in (Table 2).

Table 2: Parameters for different Models and Expressions

Model/ Expression	Parameters	Quantity
CWC	μ, β, ρ, η	0.9, 6, 0.1, 15
QD	λ, s, (1-α)	0.01, 100, 0.9
ILUBE	s, (1-α)	100, 0.9
Bootstrap ELM	model confidence	0.9
ANN LUBE	layers	3
	neurons in each layer	9, 8, 2
	population size of PSO	100
	maximum number of iterations of PSO	100
QD-LSTM / ILUBE-LSTM	input batch size *(B)*	128
		9
	neurons in LSTM layer	64
	neurons in fully connected layers	64, 2

4.2. Comparison of forecast performance

Performance is evaluated by comparing the values of PICP, PINRW, and CWC indices obtained from each model. The efficiency of the models is assessed based on the average training time. To mitigate the impact of randomness due to the initialization of weights in the neural network layers, each experiment was repeated 10 times, and the mean values of the indices were used for comparison. The average values of PICP, PINRW, CWC indices, and training time for the different models are presented in (Table 3). To evaluate the overall performance of the models across different seasons, the mean and standard deviation values of the indices for the eight cases are provided in (Table 4).

Table 3: Forecasting performance comparison of different models by wind fields

Mean Value of 10 Times		Off-shore wind field in Lake Huron				On-shore wind field in Pennsylvania			
		Spring	Summer	Autumn	Winter	Spring	Summer	Autumn	Winter
Bootstrap ELM	PICP	0.9122	0.9225	0.9185	0.9132	0.9088	0.9232	0.9109	0.9128
	PINRW	0.6014	0.6601	0.5897	0.4935	0.6159	0.5697	0.5803	0.4976
	CWC	3.6032	3.9552	3.5334	2.9569	3.6896	3.4135	3.4765	2.9815
	TIME	71.32	75.45	76.42	72.86	72.42	76.28	75.64	71.24
LUBE ANN	PICP	0.987	0.9582	0.9517	0.9832	0.9766	0.9638	0.9539	0.9834
		0.2321	0.2302	0.2101	0.2165	0.1988	0.2632	0.2487	0.1979
	CWC	1.3926	1.3814	1.2606	1.2991	1.1925	1.579	1.4923	1.1877
	TIME	137.52	141.65	142.48	138.85	136.25	143.25	141.51	137.44
QD-LSTM	PICP	0.9453	0.9223	0.9223	0.9523	0.9259	0.9416	0.9303	0.9474
		0.1649	0.1725	0.1869	0.1505	0.1585	0.2221	0.2190	0.1537
	CWC	0.9408	1.0245	1.0402	0.8543	0.8993	1.2653	1.2412	0.8147
	TIME	191.55	209.93	205.93	203.15	199.53	219.06	214.25	205.75
ILUBE-LSTM	PICP	0.9723	0.9440	0.9440	0.9718	0.9585	0.9485	0.9420	0.9730
	PINRW	0.1421	0.1643	0.1641	0.1414	0.1456	0.1862	0.1853	0.1412
	CWC	0.8184	0.9168	0.9307	0.8007	0.8279	1.0633	1.0684	0.8120
	TIME	190.21	210.27	206.66	201.37	202.42	217.86	212.09	203.46

Table 4: Average and Standard Deviation of the Indices obtained for the Eight different Datasets

	PICP		PINRW		CWC	
	MEAN	STDEV	MEAN	STDEV	MEAN	STDEV
Bootstrap Arima	0.9153	0.0054	0.5760	0.0567	3.4512	0.3397
LUBE ANN	0.9697	0.0144	0.2247	0.0233	1.3482	0.1400
QD-LSTM	0.9359	0.0121	0.1785	0.0284	1.0100	0.1687
ILUBE-LSTM	0.9568	0.0139	0.1588	0.0192	0.9048	0.1105

Coverage: In terms of coverage all the models were able to achieve PICP above the desired 90% coverage. LUBE ANN with the highest average PICP of 0.9697 as observed from (Table 4) appears to win in terms of coverage. However, QD-LSTM, and ILUBE-LSTM are also competitive with average PICP values well above 0.9 (Table 4). Since each model is meeting the criterion of coverage, any strong inferences about the performance cannot be made based on coverage, and inspection of interval size will help in getting a clear picture.

Interval Size: Observing the values of PINRW from (Table 3), it is easy to say that Bootstrap ELM loses the competition badly with excessively wide intervals in all cases. LUBE ANN though better than Bootstrap based model, however, PINRW

values are still quite high compared to QD-LUBE and ILUBE-LSTM. Thus, the higher coverage of LUBE ANN canbe attributed to the wider intervals which makes the PIs less credible compared to the two LSTM based models. Comparison of PINRW values for QD-LSTM and ILUBE-LSTM, reveals that ILUBE-LSTM excels in constructing tighter PIs. Since both these models utilize the same regression architecture, ILUBE loss appears to perform strongly in comparison to QD loss in motivating the model to refine the width of PIs. This can be observed visually from (Figure 5) which shows PI generated by different models for a portion of observations. Red coloured PIs generated by ILUBE are distinctively narrow compared to other loss functions.

CWC: Comparison of the CWC values for the different models from (Table 3) shows that ILUBE loss is winning the competition all the cases due to which ILUBE-LSTM achieves the least average CWC value (Table 4). For a clear observation bar plots of CWC are shown in (Figure 6). This strong performance of ILUBE loss in terms of CWC can be explained easily from the PICP and PINRW values from (Table 4). ILUBE-LSTM achieves higher average PICP value of 0.9568 compared to 0.9359 for QD while constructing narrow PIs with lower average PINRW of 0.1588 compared to 0.1758 for QD. Thus, outshining QD by covering higher number of observations while managing narrow interval size, ILUBE loss emerged as a strong contender direct PI generation category. In addition, the least average deviation value in CWC across the eight cases for ILUBE shows the stability of the proposed loss further strengthening the position of ILUBE to be selected for direct PI generation.

Efficiency: Observing the model training time values from (Table 3), the computational efficiency is slightly worse than LUBE ANN. This can be attributed to the sequential processing mechanism of LSTM. Although, efficiency can be improved using efficient regression architectures, however, a value of approximately 3 minutes is still very competitive for 30 minutes prediction.

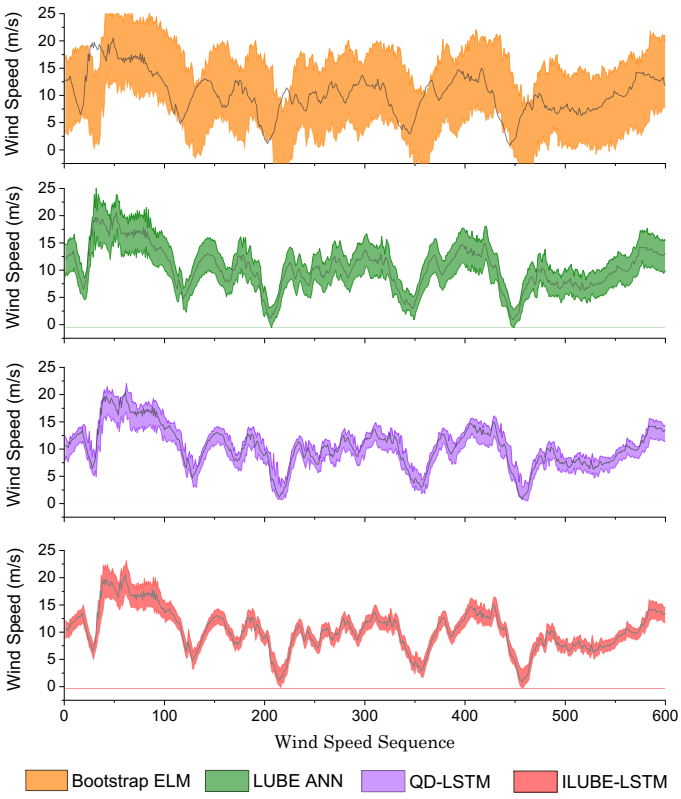

Figure 24: Visualisation of PIs constructed by different models

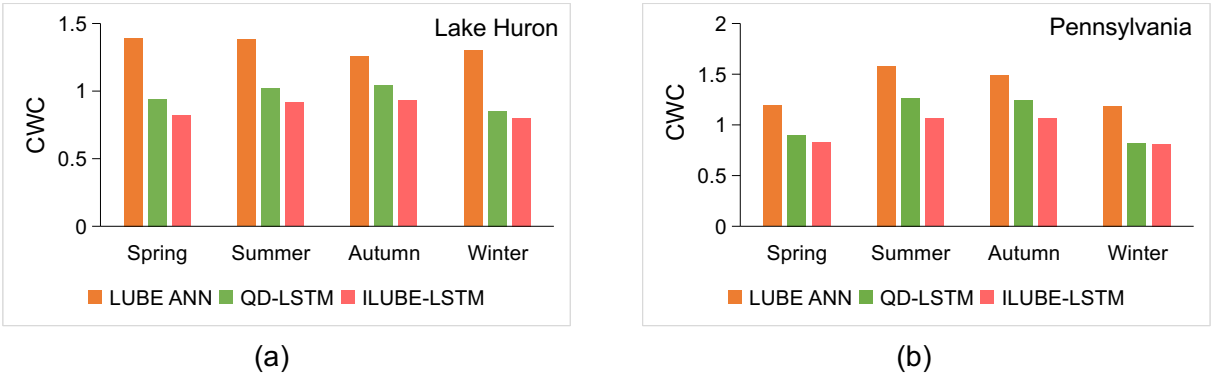

Figure 25: CWC plots for LUBE ANN, QD-LSTM, and ILUBE-LSTM for the eight different cases

5. CONCLUSION

The present article focuses on proposing a novel loss function in the category of direct PI generation methods. We build upon the fundamentals of existing methods like LUBE and QD and mitigating the problems in these we proposed ILUBE loss function to train neural networks to predict wind speed PIs at the output. Utilizing the same fundamentals of maintaining heuristic balance between coverage and interval size, we took a dynamic approach to obtain these two PI characteristics for our proposed ILUBE loss. In the computational experiments our approach constructed tighter PIs with improved coverage in comparison to the existing approaches. Thus, the proposed loss function can be a strong contender in the category of direct PI generation methods.

6. REFERENCES

Chen, X. et al. (2012) 'Electricity Price Forecasting With Extreme Learning Machine and Bootstrapping', IEEE Transactionson Power Systems, 27(4), pp. 2055–2062.

Draxl, C. et al. (2015) 'The Wind Integration National Dataset (WIND) Toolkit', Applied Energy, 151, pp. 355–366.

Erdem, E. and Shi, J. (2011) 'ARMA based approaches for forecasting the tuple of wind speed and direction', Applied Energy, 88(4), pp. 1405–1414.

Galván, I.M. et al. (2017) 'Multi-objective evolutionary optimization of prediction intervals for solar energy forecasting with neural networks', Information Sciences, 418–419, pp. 363–382.

Jiang, Y. and Huang, G. (2017) 'Short-term wind speed prediction: Hybrid of ensemble empirical mode decomposition, feature selection and error correction', Energy Conversion and Management, 144, pp. 340–350.

Khosravi, A. et al. (2011) 'Lower Upper Bound Estimation Method for Construction of Neural Network -Based Prediction Intervals', IEEE Transactions on Neural Networks, 22(3), pp. 337–346.

Li, C. et al. (2020) 'Short-Term Wind Speed Interval Prediction Based on Ensemble GRU Model', IEEE Transactions on Sustainable Energy, 11(3), pp. 1370–1380.

Lian, C. et al. (2016) 'Landslide Displacement Prediction With Uncertainty Based on Neural Networks With Random Hidden Weights', IEEE Transactions on Neural Networks and Learning Systems, 27(12), pp. 2683–2695.

Liu, H., Tian, H. and Li, Y. (2015) 'An EMD-recursive ARIMA method to predict wind speed for railway strong wind warning system', Journal of Wind Engineering and Industrial Aerodynamics, 141, pp. 27–38.

Ma, L., Huang, L. and Shi, H. (2023) 'A novel spatial–temporal generative autoencoder for wind speed uncertainty forecasting', Energy, 282, p. 128946.

Miranda, M.S. and Dunn, R.W. (2006) 'One-hour-ahead wind speed prediction using a Bayesian methodology', in 2006 IEEE Power Engineering Society General Meeting, pp. 6

Pearce, T. et al. (2018) 'High-Quality Prediction Intervals for Deep Learning: {A} Distribution-Free, Ensembled Approach',in J.G. Dy and A. Krause (eds) Proceedings of the 35th International Conference on Machine Learning, {ICML} 2018, Stockholm, Sweden, July 10-15, 2018. PMLR (Proceedings of Machine Learning Research), pp. 4072–4081.

Peng, X. et al. (2021) 'EALSTM-QR: Interval wind-power prediction model based on numerical weather prediction and deep learning', Energy, 220, p. 119692.

Quan, H., Srinivasan, D. and Khosravi, A. (2014) 'Particle swarm optimization for construction of neural network-based prediction intervals', Neurocomputing, 127, pp. 172–180.

Ren, Y., Suganthan, P.N. and Srikanth, N. (2015) 'A Comparative Study of Empirical Mode Decomposition -Based Short-Term Wind Speed Forecasting Methods', IEEE Transactions on Sustainable Energy, 6(1), pp. 236–244.

Ren, Y., Suganthan, P.N. and Srikanth, N. (2016) 'A Novel Empirical Mode Decomposition With Support Vector Regressionfor Wind Speed Forecasting', IEEE Transactions on Neural Networks and Learning Systems, 27(8), pp. 1793–1798. A

Saeed, A. et al. (2020) 'Hybrid Bidirectional LSTM Model for Short-Term Wind Speed Interval Prediction', IEEE Access, 8,pp. 182283–182294.

Saeed, A., Li, C. and Gan, Z. (2024) 'Short-term wind speed interval prediction using improved quality-driven loss based gated multi-scale convolutional sequence model', Energy, 300, p. 131590.

Shen, Y., Wang, X. and Chen, J. (2018) 'Wind Power Forecasting Using Multi-Objective Evolutionary Algorithms for Wavelet Neural Network-Optimized Prediction Intervals', Applied Sciences, 8(2).

Summerfield-Ryan, O. and Park, S. (2023) 'The power of wind: The global wind energy industry's successes and failures', Ecological Economics, 210, p. 107841.

Wu, Y.-K. and Hong, J.-S. (2007) 'A literature review of wind forecasting technology in the world', in 2007 IEEE Lausanne Power Tech, pp. 504–509.

Yao, W., Zeng, Z. and Lian, C. (2017) 'Generating probabilistic predictions using mean-variance estimation and echo state network', Neurocomputing, 219, pp. 536–547.

Zhu, X. and Genton, M.G. (2012) 'Short-Term Wind Speed Forecasting for Power System Operations', International Statistical Review, 80(1), pp. 2–23.