energies

MDPI

Article

Prediction of Icing on Wind Turbines Based on SCADA Data via Temporal Convolutional Network

Yujie Zhang [1], Nasser Kehtarnavaz [2], Mario Rotea [3,*] and Teja Dasari [4]

[1] Center for Wind Energy, Department of Electrical and Computer Engineering, University of Texas at Dallas, Richardson, TX 75080, USA; yxz162530@utdallas.edu
[2] Department of Electrical and Computer Engineering, University of Texas at Dallas, Richardson, TX 75080, USA; kehtar@utdallas.edu
[3] Center for Wind Energy, Department of Mechanical Engineering, University of Texas at Dallas, Richardson, TX 75080, USA
[4] Xcel Energy, Minneapolis, MN 55401, USA; teja.dasari@xcelenergy.com
* Correspondence: rotea@utdallas.edu

Abstract: Icing on the blades of wind turbines during winter seasons causes a reduction in power and revenue losses. The prediction of icing before it occurs has the potential to enable mitigating actions to reduce ice accumulation. This paper presents a framework for the prediction of icing on wind turbines based on Supervisory Control and Data Acquisition (SCADA) data without requiring the installation of any additional icing sensors on the turbines. A Temporal Convolutional Network is considered as the model to predict icing from the SCADA data time series. All aspects of the icing prediction framework are described, including the necessary data preprocessing, the labeling of SCADA data for icing conditions, the selection of informative icing features or variables in SCADA data, and the design of a Temporal Convolutional Network as the prediction model. Two performance metrics to evaluate the prediction outcome are presented. Using SCADA data from an actual wind turbine, the model achieves an average prediction accuracy of 77.6% for future times of up to 48 h.

Keywords: wind energy; prediction of icing on wind turbines; Temporal Convolutional Network (TCN)

check for updates

Citation: Zhang, Y.; Kehtarnavaz, N.; Rotea, M.; Dasari, T. Prediction of Icing on Wind Turbines Based on SCADA Data via Temporal Convolutional Network. *Energies* **2024**, *17*, 2175. https://doi.org/10.3390/en17092175

Academic Editors: Francesco Castellani, Martin Otto Laver Hansen and Galih Bangga

Received: 1 March 2024
Revised: 22 April 2024
Accepted: 26 April 2024
Published: 2 May 2024

1. Introduction

The share of wind energy among energy sources has been steadily growing in recent years [1]. This growth is driven by the fact that wind energy is a renewable source of energy, which has allowed lowering dependency on fossil fuels [2]. In cold climates, a challenge facing the use of wind energy involves the reduction or loss of wind-generated power due to icing on the blades of wind turbines during winter seasons [3–5]. Ice accretion on the blades causes rotation imbalance, leading to the turbine structure getting damaged [6]. Also, ice accretion causes power reduction or downtimes and hence loss of revenue [7].

Most of the works in the open literature on turbine icing using SCADA data only have focused on ice detection. Detection addresses the question “Is there ice now?” on a turbine. The purpose of our work is to answer the question “Will there be ice at a future time?” That is, we seek to predict the future occurrence of icing on a turbine. The prediction of icing on wind turbines before it occurs would allow taking actions to minimize power losses. Examples of mitigating actions that can be taken once icing is predicted include changing the turbine control settings, turning on heaters if available on the turbines, or applying anti-icing fluids to the blades. In addition, the prediction of icing would allow one to conduct a cost/benefit analysis by comparing the cost of the mitigating action with the revenue losses due to icing. The prediction of icing on wind turbines before it occurs would allow taking actions to minimize power reduction or loss due to icing. Mitigating actions that can be taken include changing the turbine control settings, turning on heaters for those turbines that are equipped with them, or applying anti-icing fluids to the blades.

A number of articles have already appeared in the literature addressing the prediction of icing on wind turbines. The approaches presented can be grouped into two main categories. The first approach involves the installation or utilization of special-purpose sensors to detect icing on wind turbines. Examples of such sensors include thermal imaging cameras and ultrasonic sensors [8–10]. There exists installation, operation, and maintenance costs associated with these sensors which can pose a challenge for their use. The second approach involves the use of Supervisory Control and Data Acquisition (SCADA) data [11] that are already available as part of a wind turbine infrastructure. In this approach, a data-driven method is used to develop a prediction model based on past values of the variables or features in SCADA data, as well as past values of meteorological data when available [12]. In [13], Kreutz et al. presented a method for predicting icing on wind turbines based on SCADA and meteorological data using a deep neural network. In [14], Tao et al. discussed a method for icing prediction on wind turbines by combining a Convolutional Neural Network (CNN) with a Gated Recurrent Unit (GRU). Zhang et al. [15] utilized Federated Learning (FL) to predict icing on wind turbines. The framework in this reference does not require the installation of any new sensors on the turbine.

The choice of variables or features from the SCADA data used by a prediction model can greatly impact the outcome of the prediction. A data preprocessing step, dubbed feature selection, is often carried out to select those variables or features that are informative for icing conditions. As discussed in [16], the selection of informative features led to lower training time as well as higher prediction accuracy. Furthermore, since, in practice, the occurrence of icing events is less frequent than in normal operation, there exists a data imbalance between the number of samples associated with icing and normal conditions. Such a data imbalance, if present during training, can adversely impact the prediction outcome [17]. In [18], Liu et al. studied the impact of data imbalance by examining different methods to establish balanced numbers of data samples for icing and normal conditions. In [19], it was shown that this data preprocessing improved prediction accuracy.

The main contribution of this paper is the development of a data processing framework for the prediction of icing in wind turbines. This framework covers key aspects of SCADA data processing, the labeling of SCADA data for icing conditions when labels are not available, the selection of informative features or variables in SCADA data for icing prediction, and using a deep neural network Temporal Convolutional Network (TCN) for icing prediction. In summary, the main contribution of this work is the development of an icing prediction framework based on SCADA data.

Temporal Convolutional Networks (TCNs) are being increasingly used in various prediction tasks and have been shown, e.g., [20,21], to be more effective in capturing long-range dependencies in data sequences compared with RNNs (recurrent neural networks) and LSTMs (Long Short-Term Memory networks). This is due to their use of dilated convolutions, which enable them to capture a larger receptive field with fewer parameters as compared with the above predictive models. Furthermore, they exhibit more stable consistent training compared with these models, which suffer from the vanishing gradient problem. Another predictor architecture superior to RNNs and LSTMs is the Bi-LSTM [22]. This network architecture has recently been proposed by [23,24] to predict the flow past a wind turbine rotor.

It is worth restating that the main contribution in our paper is the development of a data-driven framework to predict icing conditions on a wind turbine based on historical SCADA data without installing any new sensors on the turbine. As far as the predictor part of this framework is concerned, we utilized TCN in this paper, noting that in prior works [20,21], a better performance was shown for TCN compared with LSTM. It is very much possible to use other predictors such as Bi-LSTM, as discussed in [23,24].

The manuscript is organized as follows. Section 2 describes the dataset used and the preprocessing steps. Section 3 provides an overview of our TCN prediction model, followed by the icing prediction results and their discussion in Section 4. The processes for conditioning the data and training the predictor, as well as the results, are summarized in Section 5, which also describes potential future work.

2. Dataset and Data Preprocessing

The SCADA dataset considered, as well as its labeling of icing conditions, is described in this section. This dataset is from a wind turbine located in the northern part of the US. There are 11 SCADA data variables or features in the dataset, which are listed in Table 1, covering a time duration from 1 January 2023 to 1 July 2023. A binary feature denoting the operation state of the turbine is also listed in this table, which is used for data preprocessing. Consecutive samples in the dataset are 10 min apart. There are a total of 26,209 data samples in the dataset. Missing values are filled by performing a linear interpolation between adjacent nonmissing data samples. In addition, weather data listed in Table 2 for the same time period and location are obtained from the VisualCrossing weather database [25]. The rated power of this wind turbine is 2 MW with cut-in wind speed of 4 m/s, rated wind speed of 12 m/s, and cut-out wind speed of 25 m/s.

Table 1. Variables or features in the SCADA dataset.

Variable or Feature	Unit	Description
Power_Avg	kW	Generated power
Wind Speed	m/s	Wind speed
Gen_RPM	RPM	Generator speed
Wind Direction	degree (°)	Wind direction
Nacel_Direct	degree (°)	Nacelle direction
Blade_Pitch	degree (°)	Blade pitch angle
Yaw_Error	degree (°)	Yaw error
Temper_Nac	Celsius (°C)	Nacelle temperature
Temper_Amb	Celsius (°C)	Ambient temperature
Temper_Gen	Celsius (°C)	Generator bearing temperature
Temper_Gear	Celsius (°C)	Gear bearing temperature
Oper_State	-	Flagged for normal operating condition

Table 2. Variables or features in the VisualCrossing weather database.

Variables or Features	Unit	Description
Temperature	Celsius (°C)	Air temperature from the weather database
Relative Humidity	%	Relative humidity from the weather database

Data samples in this dataset or other SCADA datasets may not be labeled for icing conditions. Three rules are considered here to label the data samples as "ice" or "normal" based on the icing conditions described in [26,27]. These rules are listed in Table 3, which reflect temperature, relative humidity, and actual power. If all the conditions in these rules are met for a data sample, that data sample is labeled as "ice" and is represented by "1" as the output of our prediction model. Otherwise, it is labeled as "normal" and is represented by "0" as the output of our prediction model.

Table 3. Rules for ice labeling of data samples.

Region 2 of Power Curve	Region 3 of Power Curve
Temperature < 0 °C	Temperature < 0 °C
Relative Humidity > 85%	Relative Humidity > 85%
Actual Power < 85% × Power Curve	Actual Power < 85% × Rated Power

Different window sizes were examined, and the one with the best match between the cases of normal state (Oper_State, obtained from Table 1) and labels created from Table 3 (normal, represented by "0") was selected. After the ice labeling process, it is noted that the ratio of ice data samples to normal data samples was 1 to 6. To balance the number between ice and normal samples, downsampling was performed on normal data samples within the time series to select the most representative samples in a deterministic way.

Feature selection is a data preprocessing step to reduce the number of features or variables used in a prediction model. A lower number of features reduces the complexity as well as the training time of the prediction model. This step involves removing redundant or noninformative features for icing detection and prediction. In this work, feature selection is performed by computing the Fisher Distance [28], which is a separation measure between the distributions of a feature for the normal and icing conditions or classes as per Equation (1):

$$FD = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{1}$$

where μ_1 and σ_1 denote the mean and standard deviation of the "ice" class distribution, and μ_2 and σ_2 denote the mean and standard deviation of the "normal" class distribution. Fisher Distance is used to rank order all the features from the most significant to the least significant ones in terms of their ability to distinguish between icing and normal conditions. There are a total of 13 features or variables, 11 features or variables from the SCADA dataset and 2 features or variables from the weather database. All 13 Fisher Distances are computed and ranked in descending order, see Table 4. A feature with a higher Fisher Distance provides a higher separation between the "ice" and "normal" distributions or classes. As an example, the distributions for the feature or variable with the highest Fisher Distance, i.e., generated power, between "ice" and "normal" classes is shown in Figure 1. Note the concentration of the icing case around the low-power region for this particular wind turbine. All samples shown for the "normal" and "ice" labels are flagged as "normal operating condition" (see Table 1).

In order to select the most informative features for icing, first the highest Fisher Distance in Table 4 or Power_Avg is used as the only input feature to our prediction model discussed in Section 3. Then, the first two highest Fisher Distance features, that is, Power_Avg and Temp_Gear, are used as the inputs to our prediction model. The number of features is steadily increased according to their Fisher Distance till all 13 features are used as the inputs to our prediction model. Each time, the model prediction accuracy (described later in Section 4) is computed; see Table 5. As seen from this table, the highest accuracy is obtained when the top 12 features are used as the inputs to our prediction model. The last feature does not provide any added benefit, and thus, it is not considered as an input to our prediction model. This feature selection strategy is general and can be applied to any other SCADA dataset.

Table 4. Fisher Distances for all the features or variables examined.

Feature	Fisher Distance
Power_Avg	1.01×10^{0}
Temp_Gear	8.34×10^{-1}
Gen_RPM	6.35×10^{-1}
Relative Humidity (local weather station)	6.09×10^{-1}
Wind Speed	5.07×10^{-1}
Temp_Gen	4.59×10^{-1}
Temper_Nac	1.40×10^{-1}
Blade_Pitch	7.98×10^{-2}
Temperature (local weather station)	3.98×10^{-2}
Wind Direction	2.56×10^{-3}

Table 4. *Cont.*

Feature	Fisher Distance
Nacel_Direct	2.31×10^{-3}
Temper_Amb	1.73×10^{-3}
Yaw_Error	1.01×10^{-3}

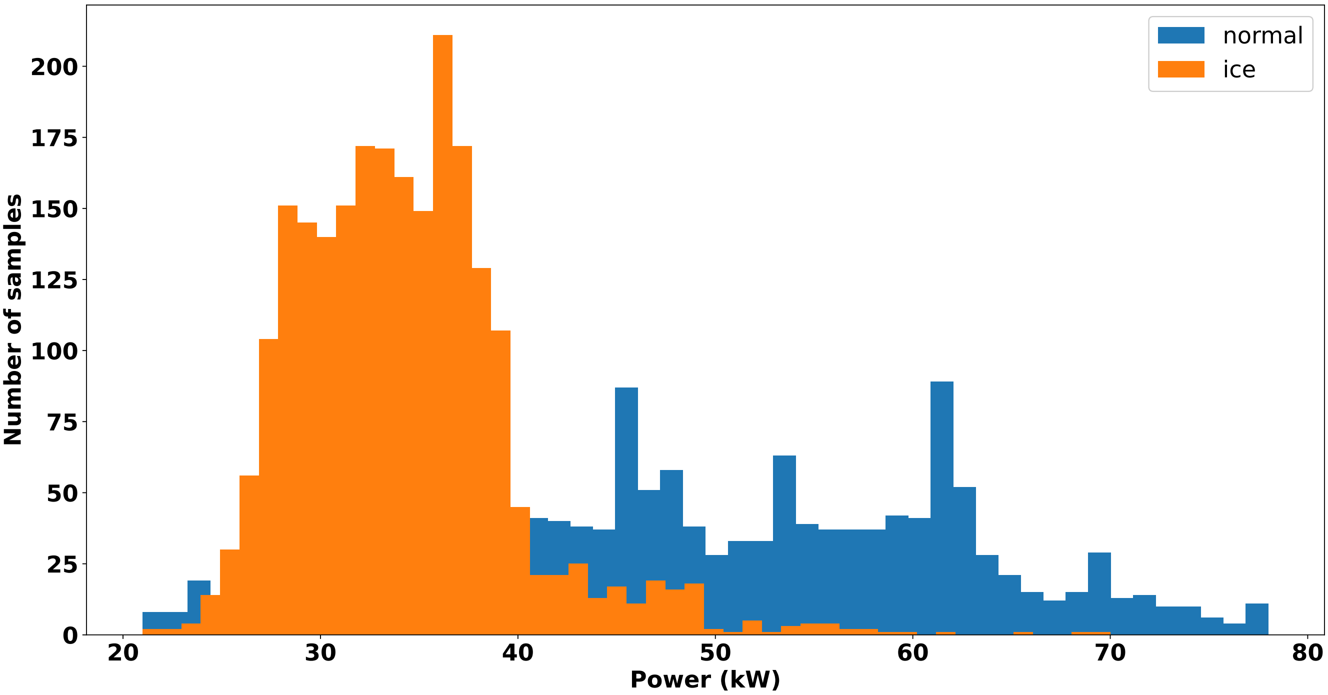

Figure 1. Distributions of the feature or variable power for "ice" and "normal" classes.

Table 5. Prediction performance using different number of features.

Number of Features in Descending Order of Their Fisher Distance	Prediction Accuracy (%)
1	74.70
2	74.69
3	74.77
4	76.47
5	76.86
6	77.24
7	77.24
8	77.61
9	77.97
10	78.85
11	79.36
12	**80.07**
13	79.92

3. Prediction Model

An overview block diagram of our prediction framework is shown in Figure 2. Our prediction framework includes three data processing blocks: Data Preprocessing, Prediction Model Components, and Prediction Evaluation. The steps of the Data Preprocessing block are described in the previous section. In this section, the Prediction Model Components block and the Prediction Model Evaluation block are described.

A Temporal Convolutional Network (TCN) is used in this work as our prediction model, noting that in other applications, this network has been shown to be more effective than other similar prediction models [29,30]. An illustration of the TCN architecture is shown in Figure 3. This architecture consists of so-called convolution layers, ReLU (Rectified Linear Unit) layers, and dropout layers. The convolution layer takes in a two-dimensional input tensor or matrix of size ws by F, where ws denotes the input window size and F denotes the number of features. To obtain the output corresponding to each feature sequence or time series, the dot product of the input sequence and a kernel vector of size 3 is computed. This kernel size was found to produce the highest accuracy compared with other kernel sizes. A zero padding of size 2 is performed to ensure that the output sequence for each feature has the same length as the input sequence. Prediction is normally conducted by considering the present and previous values of the features. The number of previous feature values is shown in Figure 4, which illustrates both the input window size and the number of features used. The experiments reported in Section 4 showed that an input window size of 21 data samples (corresponding to 3.5 h) achieved the highest accuracy. No major changes were seen beyond this input window size for the dataset considered. The output is a binary integer number, indicating the prediction outcome (1 if the output is "ice" and 0 if the output is "normal"). After the convolution layer, a ReLU layer is used as the activation function, which addresses the vanishing gradient problem [31]. Then, a dropout layer with a dropout probability of 0.2 as normally performed is then used for regularization to prevent overfitting during the training session [32].

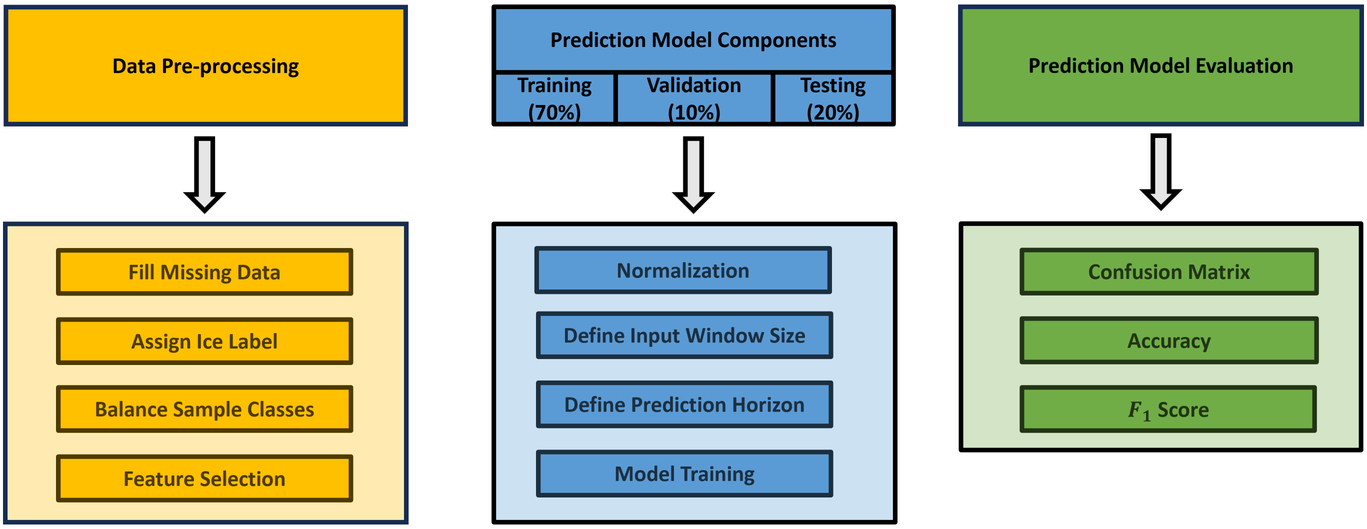

Figure 2. Block diagram of icing prediction framework consisting of Data Preprocessing, Prediction Model Components, and Prediction Model Evaluation.

The training session starts by first ensuring that all the selected 12 features have the same dynamic range for the prediction model training; they are normalized using the standard min–max normalization approach expressed by Equation (2):

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{2}$$

where x denotes a feature value, and $x_{\min}$ and $x_{\max}$ denote its minimum and maximum values in the dataset, respectively. After this normalization, the values for all the features to carry out the model training are between 0 and 1.

During the training session, for each epoch or one complete pass through the dataset, the TCN model takes in a batch of training input samples and generates an output label, which is then compared with the actual icing label. Based on the Binary Cross-Entropy loss function, the error between the predicted icing label and the actual icing label is computed via the Adam optimizer with a learning rate of 0.001 in order to update kernel weights to minimize the error. The TCN model parameters used in the training session are shown in Table 6.

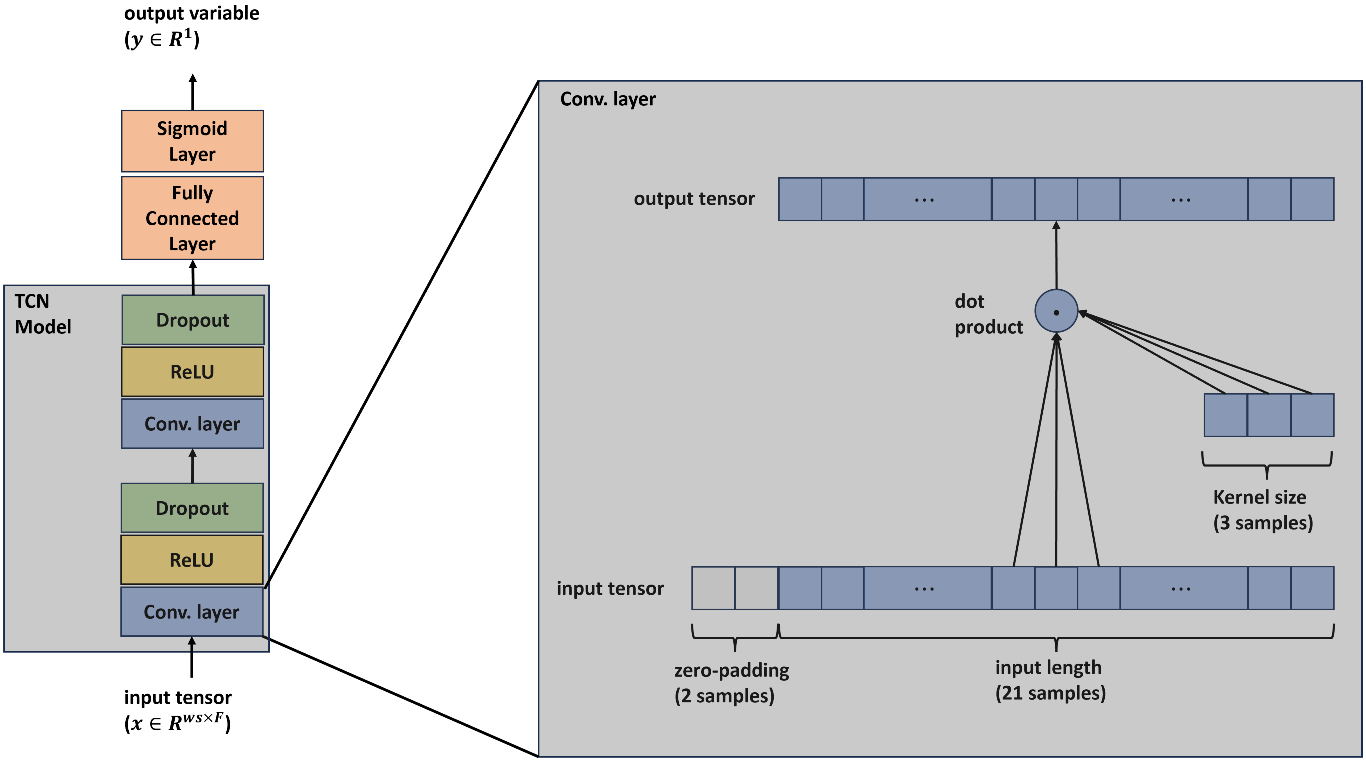

Figure 3. TCN architecture of the developed icing prediction framework.

Table 6. TCN model parameters used.

Parameters	Value or Setting
Optimizer	Adam
Loss Function	Binary Cross-Entropy
Epoch	10
Learning rate	0.001
Batch size	8
Kernel size	3
Dropout Probability	0.2

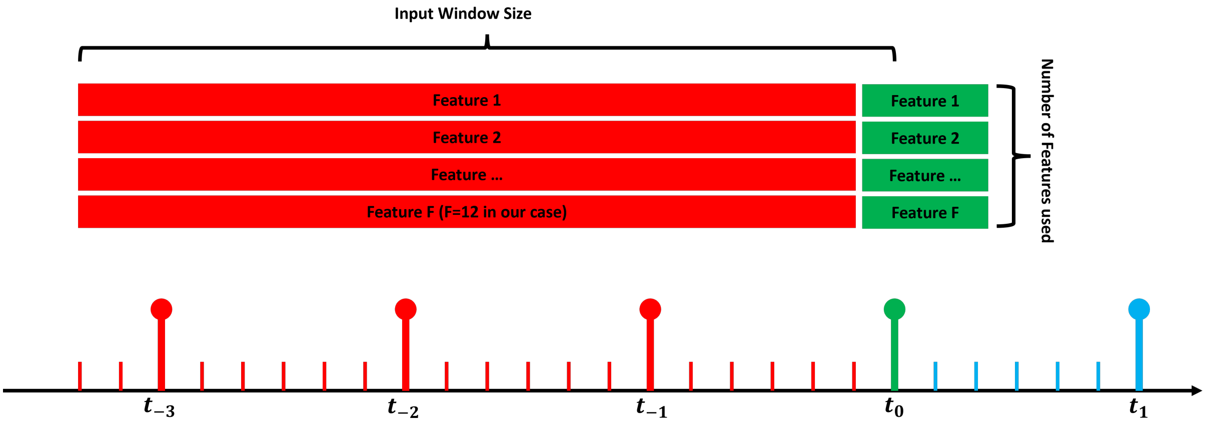

Figure 4. Construction of input tensors illustrating the input window size and the number of features.

There are 288 predictors designed with prediction horizons from 10 min ahead until 2 days ahead. For example, if one desires to predict icing one hour into the future (or 6 samples ahead), each training output sample needs to be assigned an icing label corresponding to 1 h ahead, as illustrated in Figure 5. Our prediction framework is designed in such a way that different prediction horizons can be selected or specified by the user.

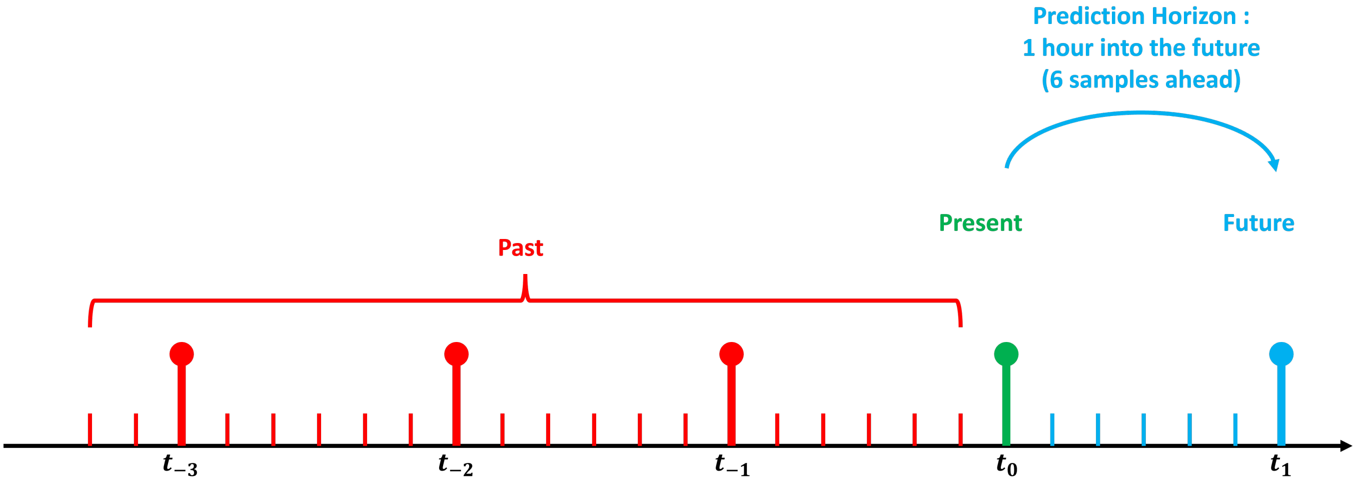

Figure 5. Illustration of prediction horizon.

4. Results and Discussion

In this section, two widely used measures derived from the so-called confusion matrix are used to evaluate the prediction model introduced in Section 3. The measures are accuracy and F_1 score, which are reported for different prediction horizons. The confusion matrix for the problem under consideration consists of two entries, corresponding to "ice" and "normal"; see Table 7. This matrix consists of four entries or metrics, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), and their definitions are stated in Table 8. The confusion matrix is a table that can assess the performance of a prediction model in a complete manner as compared with just accuracy. In our case, a binary prediction is performed by classifying each test sample into "ice" or "normal". The test samples correspond to 20% of the dataset samples. These test samples were randomly selected and have no overlap with the training samples, i.e., they represent fresh data used to evaluate prediction performance.

Table 7. Confusion matrix for icing prediction.

	Predicted Ice	Predicted Normal
Actual ice	TP	FN
Actual normal	FP	TN

Table 8. Definition of metrics.

Metric	Definition
True Positive (TP)	Portion of the samples the model correctly predicted as positive cases ("ice").
True Negative (TN)	Portion of the samples the model correctly predicted as negative cases ("normal").
False Positive (FP)	Portion of the samples the model incorrectly predicted as positive cases ("ice") when they were actually negative cases ("normal").
False Negative (FN)	Portion of the samples the model incorrectly predicted as negative cases ("normal") when they were actually positive cases ("ice").

Based on TP, TN, FP, and FN, the two commonly used performance metrics of accuracy and F_1 score are computed as follows:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \tag{3}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{4}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{5}$$

$$F_1 \text{ score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \tag{6}$$

Accuracy indicates the proportion of correctly classified samples out of all the samples. Precision indicates the proportion of true positive predictions among all positive predictions. It reflects the ability of the model to avoid false positives. Recall indicates the proportion of true positive predictions among all actual positive predictions. It reflects the ability of the model to identify all positive predictions.

We use two metrics, "accuracy" and "F_1 score", as they are complementary. Accuracy indicates the number of times predictions are correct as reflected in the diagonal elements of the confusion matrix. F_1 score provides a measure of incorrect predictions or type I and type II errors, as reflected in the off-diagonal elements of the confusion matrix. Type I errors denote the false positive prediction samples, where normal prediction samples are incorrectly classified as ice prediction samples, and type II errors denote the false negative prediction samples, where ice prediction samples are incorrectly classified as normal prediction samples. Since the F_1 score is obtained by the harmonic mean of type I and type II errors, lower incorrectly predicted samples result in higher F_1 score values.

Our prediction framework was used to conduct icing prediction for up to 48 h or two days into the future, corresponding to 288 prediction units, with one unit denoting 10 min. The duration of the predicted icing time series covered the winter season time frame from January to April. Figure 6 shows a plot of accuracy and F_1 scores obtained for all prediction horizons up to two days. The best accuracy (87.2%) and F_1 score (0.67) are obtained for the 10 min prediction horizon, which is expected, since this is the smallest horizon. To provide a better idea of performance across several prediction horizons, note that the average accuracy for prediction horizons from 10 min (zero hour) to 1 day (dash vertical line in Figure 6) is 81.6%. The average accuracy drops to 77.6% when calculated across all prediction horizons up to 2 days. Similarly, F_1 score averages are 0.6 and 0.5, for up to 1 day and 2 days, respectively. Additional averages can be found in Table 9, which shows average accuracy and F_1 scores for four different prediction horizon cases: from 10 min (0 h) to 12 h, from 12 h to 24 h, from 24 h to 36 h, and from 36 h to 48 h.

With a prediction horizon of ten minutes into the future (see Predictor1 in Figure 6), the best prediction accuracy of 87.2% was obtained among all the cases. The confusion matrix for this case is shown in Table 10. Also, to visualize the predictions as a function of time, a portion of the predicted icing time series for this case is shown in Figure 7. In this figure, blue regions correspond to actual icing labels, while green regions correspond to predicted icing labels generated by our prediction model. Most of the icing durations were correctly predicted by our model, while some false alarms were also generated. Recall that the predicted time series corresponds to the one-step ahead (10 min) predictor.

Table 9. Average accuracy and F_1 score across varying prediction horizons.

Prediction Horizon (Hours)	Average Prediction Accuracy (%)	Average F_1 Score
0–12	81.8	0.60
12–24	81.4	0.59
24–36	73.7	0.48
36–48	73.4	0.45

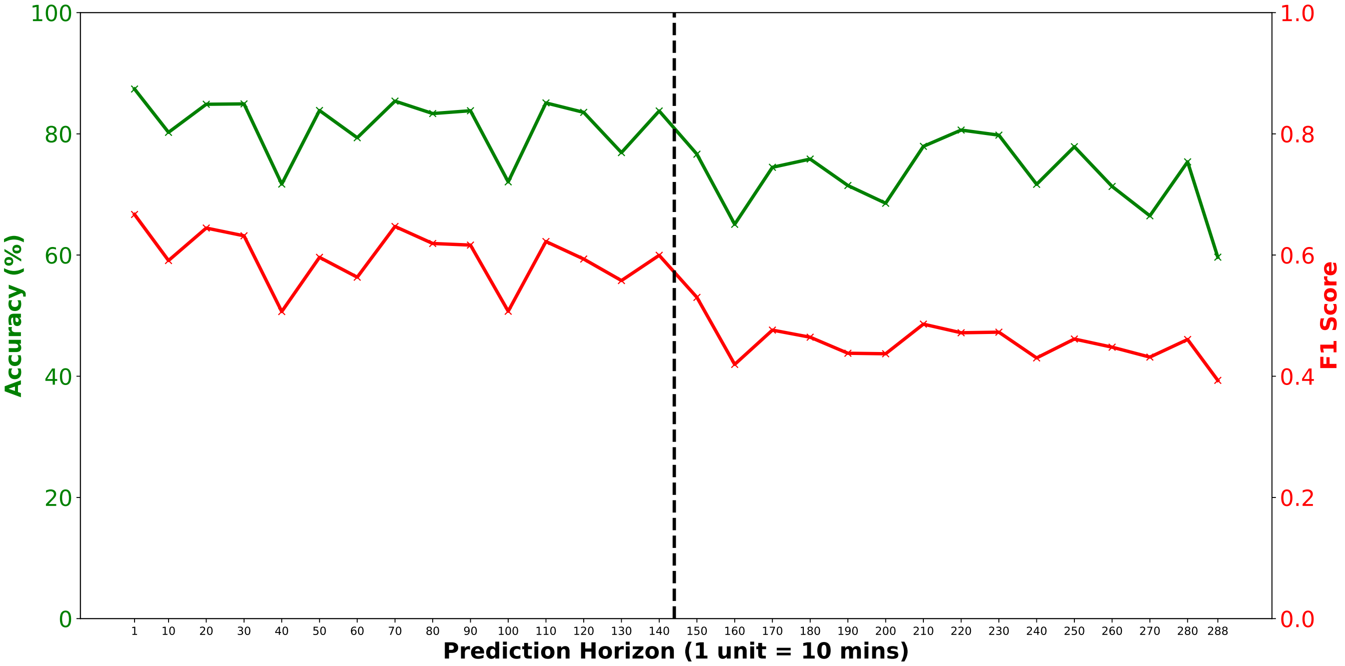

Figure 6. Accuracy and F_1 score variation with increasing prediction horizon.

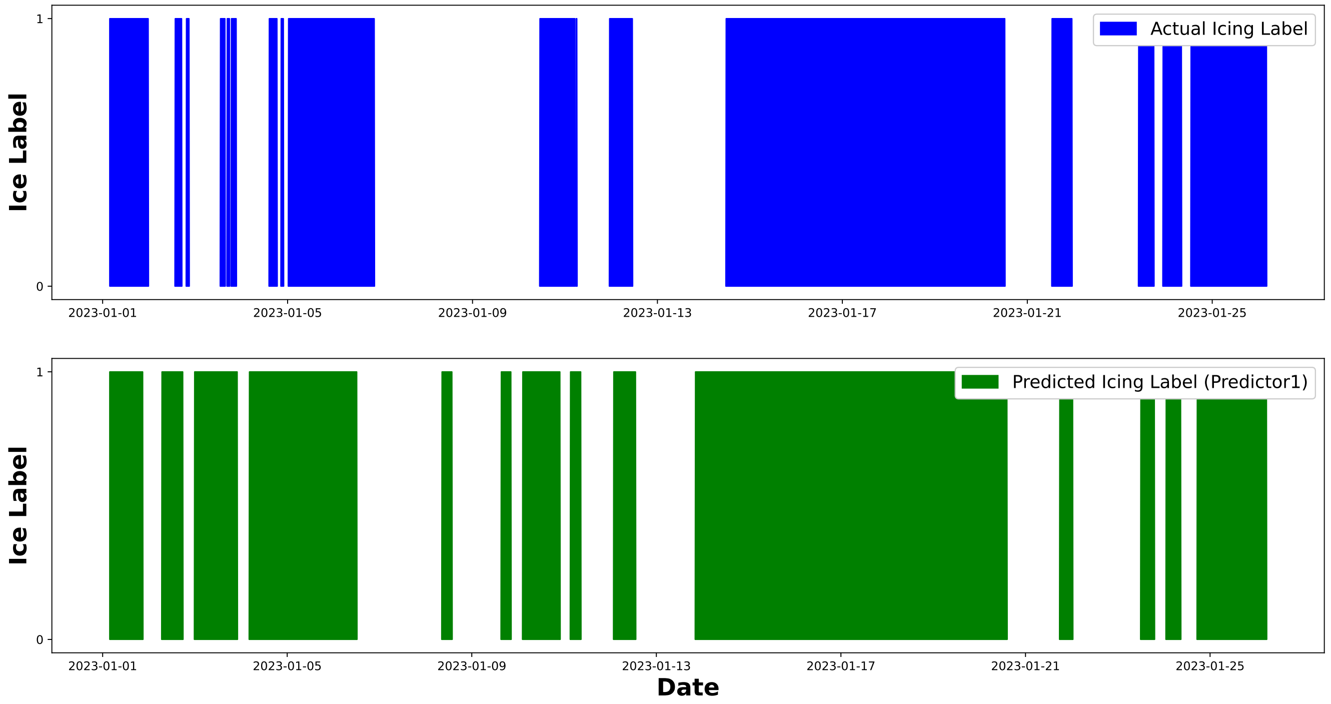

Figure 7. Time series of icing prediction when using the one-step-ahead predictor.

It should be noted that since different datasets are used in different papers, and these datasets are not publicly available, it is not possible to compare the performance of our prediction model with the prediction models previously published. Nevertheless, the prediction accuracies obtained in this work are comparable to those previously reported. Furthermore, it is worth pointing out that when the true ice labels are unknown, a portion of the prediction errors could be due to the ice labeling procedure and not due to the prediction algorithm. What distinguishes our results from previously reported work is that we explained all the steps involved when developing a framework for icing prediction from SCADA data.

It takes 0.33 milliseconds on a PC equipped with an Nvidia GPU RTX A4000 for the TCN predictor network to provide one output sample at a specified future time in response to a time window of past SCADA data measurements. This means that the prediction can be conducted in real time for the SCADA data measurements captured every 10 min.

Table 10. Confusion matrix of one unit or sample ahead case (Predictor1).

	Predicted Ice	Predicted Normal
Actual ice	2185	674
Actual normal	1536	12,863

5. Conclusions and Future Work

In this paper, a data-driven framework has been developed to predict icing on wind turbines based on SCADA data. The prediction of icing enables the assessment of a possible drop in energy supply from wind power due to incoming cold weather. This information would allow operators to prepare for shifting supply to generators that are less impacted by icing events and/or to take actions to mitigate ice accumulation in order to reduce energy losses from wind turbines.

The proposed icing prediction framework can be summarized as follows:

- Data preprocessing: Fill in missing data, assign icing labels, balance the number of samples between normal and ice conditions, and select the features from the SCADA and meteorological data for icing prediction.
- Predictor: Separate the selected features into “training” and “testing” samples to develop and evaluate a Temporal Convolutional Network (TCN) with binary output (ice or normal condition).

The experimental results from an actual wind turbine in a cold climate can be summarized as follows:

- The accuracy and F_1 score of the TCN predictor from a 10 min prediction horizon to a 48 h prediction horizon are shown in Figure 6. This figure shows how predictor performance decreases with the prediction horizon, which is expected.
- The average accuracy and F_1 score every 12 h interval are in Table 9.
- The average accuracy for a short-term prediction horizon (from 10 min to 1 day) is 81.6%, while the average accuracy for a long-term prediction horizon (from 10 min to 2 days) drops to 77.6%.

In most applications, it is of interest to know the icing conditions across an entire wind farm. An extension of the work reported in this paper would be to predict icing conditions at the farm level.

Author Contributions: Conceptualization, M.R., N.K. and T.D.; methodology, Y.Z., M.R. and N.K.; software, Y.Z.; validation, Y.Z., M.R. and N.K.; formal analysis, Y.Z., M.R. and N.K.; investigation, Y.Z., M.R. and N.K.; resources, M.R. and N.K.; data curation, Y.Z., T.D., M.R. and N.K.; writing—original draft preparation, Y.Z. and N.K.; writing—review and editing, Y.Z., M.R., N.K. and T.D.; visualization, Y.Z., M.R. and N.K.; supervision, M.R., N.K. and T.D.; project administration, M.R. and N.K.; funding acquisition, M.R. and N.K. All authors have read and agreed to the published version of the manuscript.

Funding: This material is based upon work supported by the National Science Foundation under Award Number 1916776, Phase II IUCRC at UT Dallas: Center for Wind Energy Science, Technology and Research (WindSTAR) and the WindSTAR IUCRC Company Members. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or WindSTAR members.

Data Availability Statement: Dataset not available due to nondisclosure restrictions.

Acknowledgments: The authors would like to acknowledge the industrial advisory members of the WindSTAR NSF IUCRC for their feedback.

Conflicts of Interest: Author Teja Dasari was employed by the company Xcel Energy. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Sadorsky, P. Wind Energy for Sustainable Development: Driving Factors and Future Outlook. *J. Clean. Prod.* **2021**, *289*, 125779. [CrossRef]
2. Chang, L.; Saydaliev, H.B.; Meo, M.S.; Mohsin, M. How renewable energy matter for environmental sustainability: Evidence from top-10 wind energy consumer countries of European Union. *Sustain. Energy Grids Netw.* **2022**, *31*, 100716. [CrossRef]
3. Hochart, C.; Fortin, G.; Perron, J.; Ilinca, A. Wind turbine performance under icing conditions. *Wind Energy* **2008**, *11*, 319–333. [CrossRef]
4. Li, Y.; Tagawa, K.; Feng, F.; Li, Q.; He, Q. A wind tunnel experimental study of icing on wind turbine blade airfoil. *Energy Convers. Manag.* **2014**, *85*, 591–595. [CrossRef]
5. Parent, O.; Ilinca, A. Anti-icing and de-icing techniques for wind turbines: Critical review. *Cold Reg. Sci. Technol.* **2011**, *65*, 88–96. [CrossRef]
6. Alsabagh, A.; Tiu, W.; Xu, Y.; Virk, M. A review of the effects of ice accretion on the structural behavior of wind turbines. *Wind. Eng.* **2013**, *37*, 59–70. [CrossRef]
7. Lacroix, A.; Manwell, J. Wind Energy: Cold Weather Issues. University of Massachusetts at Amherst, Renewable Energy Laboratory. 2000. Available online: https://www.researchgate.net/publication/237706746_Wind_Energy_Cold_Weather_Issues (accessed on 23 December 2023).
8. Gantasala, S.; Luneno, J.; Aidanpaa, J. Detection of ice mass based on the natural frequencies of wind turbine blade. *Wind. Energy Sci. Discuss.* **2016**, *2016*, 1–17.
9. Guan, B.; Su, Z.; Yu, Q.; Li, Z.; Feng, W.; Yang, D.; Zhang, D. Monitoring the blades of a wind turbine by using videogrammetry. *Opt. Lasers Eng.* **2022**, *152*, 106901. [CrossRef]
10. Rizk, P.; Al Saleh, N.; Younes, R.; Ilinca, A.; Khoder, J. Hyperspectral imaging applied for the detection of wind turbine blade damage and icing. *Remote Sens. Appl. Soc. Environ.* **2020**, *18*, 100291. [CrossRef]
11. Dong, X.; Gao, D.; Li, J.; Zhang, J.; Zheng, K. Blades icing identification model of wind turbines based on SCADA data. *Renew. Energy* **2020**, *162*, 575–586. [CrossRef]
12. Karami, F.; Zhang, Y.; Rotea, M.; Bernardoni, F.; Leonardi, S. Real-time wind direction estimation using machine learning on operational wind farm data. In Proceedings of the 2021 60th IEEE Conference on Decision and Control (CDC), Austin, TX, USA, 14–17 December 2021; pp. 2456–2461.
13. Kreutz, M.; Ait-Alla, A.; Varasteh, K.; Oelker, S.; Greulich, A.; Freitag, M.; Thoben, K.D. Machine Learning-based Icing Prediction on Wind Turbines. *Procedia Cirp.* **2019**, *81*, 423–428. [CrossRef]
14. Tao, C.; Tao, T.; Bai, X.; Liu, Y. Wind Turbine Blade Icing Prediction Using Focal Loss Function and CNN-Attention-GRU Algorithm. *Energies* **2023**, *16*, 5621. [CrossRef]
15. Zhang, D.; Tian, W.; Cheng, X.; Shi, F.; Qiu, H.; Liu, X.; Chen, S. FedBIP: A Federated Learning Based Model for Wind Turbine Blade Icing Prediction. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 3516011. [CrossRef]
16. Jovi, A.; Brki, K.; Bogunovi, N. A review of feature selection methods with applications. In Proceedings of the 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 25–29 May 2015; pp. 1200–1205.
17. Batista, G.; Prati, R.; Monard, M. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [CrossRef]
18. Liu, L.; Guan, D.; Wang, Y.; Ding, C.; Wang, M.; Chu, M. Data-Driven Prediction of Wind Turbine Blade Icing. In Proceedings of the 2021 China Automation Congress (CAC), Beijing, China, 22–24 October 2021; pp. 5211–5216.
19. Bai, X.; Tao, T.; Gao, L.; Tao, C.; Liu, Y. Wind turbine blade icing diagnosis using RFECV-TSVM pseudo-sample processing. *Renew. Energy* **2023**, *211*, 412–419. [CrossRef]
20. Bai, S.; Kolter, Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
21. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y. Convolutional sequence to sequence learning. *Int. Conf. Mach. Learn.* **2017**, *70*, 1243–1252.
22. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [CrossRef] [PubMed]
23. Geibel, M.; Bangga, G. Data reduction and reconstruction of wind turbine wake employing data driven approaches. *Energies* **2022**, *15*, 3773. [CrossRef]
24. Ali, N.; Calaf, M.; Cal, R. Clustering sparse sensor placement identification and deep learning based forecasting for wind turbine wakes. *J. Renew. Sustain. Energy* **2021**, *13*, 2. [CrossRef]

25. Visualcrossing. Available online: https://www.visualcrossing.com (accessed on 13 November 2023).
26. Swenson, L.; Gao, L.; Hong, J.; Shen, L. An Efficacious Model for Predicting Icing-induced Energy Loss for Wind Turbines. *Appl. Energy* **2022**, *305*, 117809. [CrossRef]
27. Davis, N.N.; Pinson, P.; Hahmann, A.N.; Clausen, N.E.; Žagar, M. Identifying and Characterizing the Impact of Turbine Icing on Wind Farm Power Generation. *Wind Energy* **2016**, *19*, 1503–1518. [CrossRef]
28. Mika, S.; Ratsch, G.; Weston, J.; Scholkopf, B.; Mullers, K. Fisher discriminant analysis with kernels. In Proceedings of the Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No. 98TH8468), Madison, WI, USA, 25 August 1999; pp. 41–48.
29. Lea, C.; Vidal, R.; Reiter, A.; Hager, G. Temporal convolutional networks: A unified approach to action segmentation. In Proceedings of the Computer Vision–ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–10 and 15–16 October 2016; Proceedings, Part III 14; pp. 47–54.
30. Jin, M.; Wen, Q.; Liang, Y.; Zhang, C.; Xue, S.; Wang, X.; Zhang, J.; Wang, Y.; Chen, H.; Li, X. Large models for time series and spatio-temporal data: A survey and outlook. *arXiv* **2023**, arXiv:2310.10196.
31. Li, Y.; Yuan, Y. Convergence analysis of two-layer neural networks with relu activation. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 598–608.
32. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.