Icing Detection and Prediction for Wind Turbines Using Multivariate Sensor Data and Machine Learning

Feng Ye[a], Ahmed Aziz Ezzat[a]

[a] *Industrial & Systems Engineering, Rutgers University, Piscataway, 08854, NJ, USA*

Abstract

Adverse weather events can significantly compromise the availability and economics of a wind farm. This paper focuses on rotor icing detection, which constitutes a major challenge in wind farm operation. When ice accumulates on wind turbine blades, it causes substantial generation losses, operational disruptions, and safety hazards to the personnel, assets, and equipment in a wind farm. Alerts about early signs of rotor icing can assist operators in proactively initiating icing mitigation measures. To this end, we propose a machine-learning-based framework that effectively learns the unique signatures of icing events. The framework effectively extracts salient features by condensing the multivariate turbine sensor data into a small-sized subset of information-rich descriptors. Those, along with power-curve-derived features, are used to train a deep-learning-based model that flags icing events and estimates icing probabilities. We also propose a new loss measure, called the icing power loss error (IPLE), that realistically quantifies the expected icing-related power losses. Our experiments show that the proposed framework achieves up to 96.4% accuracy in flagging icing events, while keeping the number of false alarms at minimum. When compared to prevalent data-driven benchmarks, up to 18.7% reduction in power loss estimation error is realized.

Keywords: Icing Detection, Machine Learning, Condition Monitoring, Wind Power.

1. Introduction

Wind energy is evolving into a dominant source of electricity generation worldwide, primarily attributed to its numerous environmental benefits and rapidly falling costs [1]. Enabling wind energy to unlock larger penetration levels will be contingent on enhancing the resilience of wind farms against adverse weather events [2]. One promising solution is to invest in intelligent turbine controls and smart operations and maintenance (O&M) decision support systems that can aid wind farm operators in minimizing extreme weather impacts on overall plant availability and economics [3, 4]. Along those lines, this paper specifically addresses the data-driven monitoring of rotor icing, which refers to the formation and accumulation of ice on the rotor of a wind turbine. Due to its impact on the aerodynamic efficiency of the blades, rotor icing can cause substantial reduction in the power output of a wind turbine, as well as in its structural reliability [5, 6]. Ice throw from wind turbine blades also presents safety hazards to nearby personnel, assets, equipment, and facilities [7]. It is thus important for wind farm operators to have an icing monitoring system in place that effectively flags early signs of rotor icing before it massively disrupts wind farm operations. Icing alerts can assist the operators in initiating proactive icing mitigation measures in time, e.g., dispatching icing removal services, initiating de-icing technologies, or at the least, timely shutting down the turbines to hedge against increased safety hazards and minimize post-event downtime and recovery efforts [8, 9].

The utility of an effective icing monitoring system is relevant to wind turbines in cold and mild climates alike. Understandably, turbines located in cold climates frequently experience icing events and may already have some icing protection technology (IPT) in place (e.g., anti- or de-icing solutions). Hence, an effective icing monitoring system would be used as part of their routine operations to proactively trigger those IPTs. On the other hand, turbines in milder climates may also benefit from an icing monitoring system. Because significant winterization investments for those turbines may not be economically justified, an effective icing monitoring system offers a low-cost, high-value solution against extreme cold weather events, especially in light of their escalating rates in recent years. A relevant example is the 2021 Texas blackout events, which have spurred large-scale debates about the vulnerability of wind turbines to extreme cold weather [10].

The International Energy Agency (IEA)'s Task 19 is one of the largest collaborative initiatives to date to address icing-related challenges in wind farms [11]. For rotor icing, it recommends power-curve-based, data-driven approaches that act on sensor data from Supervisory Control and Data Acquisition (SCADA) systems to issue icing alarms for de-icing activation and power loss estimation. The main challenge therein is to attain a sensible

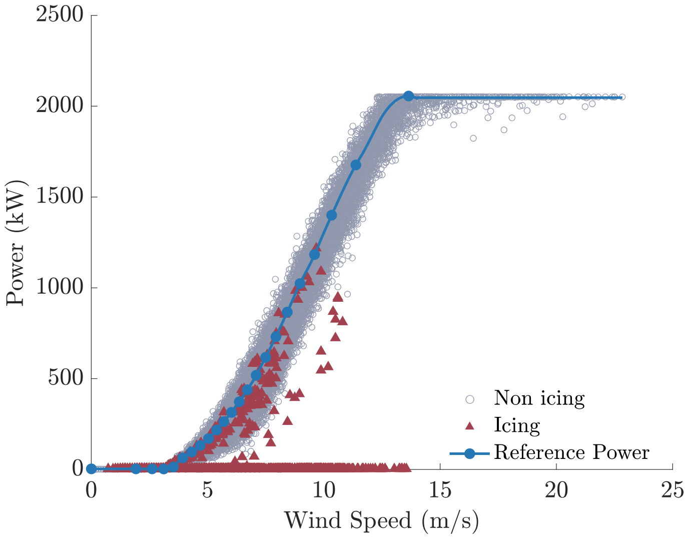


Figure 1: Impact of rotor icing on a wind turbine's power output. The key challenge is to disentangle, as early as possible, icing events (red triangles) from inherent randomness in a turbine's power production (light grey circles).

balance between false and missed alarms, especially in light of the inherent stochasticity in a wind turbine's power production. Figure 1 shows the impact of icing on the power output of a wind turbine, where the blue curve depicts the production under normal conditions (the non-iced power curve) on top of the actual measurements (grey circles), whereas red triangles denote icing events. A monitoring system that is too conservative can mistake the random variations around a power curve (due to weather, control settings, measurement uncertainty, etc.) as icing events, resulting in too many false alarms, gradually eliminating the trust in the system. In contrast, a monitoring system that is too liberal often results in a delayed (or missed) detection, significantly compromising its utility to farm operators.

The literature on rotor icing monitoring can be broadly grouped into two categories: *direct* and *indirect* approaches. Direct approaches rely on dedicated sensor technologies, collectively referred to as "icing detectors," to measure and detect rotor icing [12]. Often installed on rotor blades, the essence of icing detectors is to identify changes in some physical characteristics of the blade relative to its surrounding environment that are triggered by icing events. The large majority of available technologies therein are suited for measuring meteorological and instrumental icing, whereas fewer sensing technologies are available for rotor icing, which has a shorter incubation and ablation time, making it harder to detect, let-alone predict [13]. Examples of such technologies include piezo-electric and fibre-optic accelerators glued inside the blade to measure changes in eigen frequencies and natural oscillations [14, 15], retrofit sensors taped on the blade surface to measure icing-induced changes in impedance and capacitance [16], thermal infrared sensors that measure the surface-emitted radiance under icing versus non-icing conditions [17], as well as approaches based on ultrasonic guided waves [18]. The IEA Task 19 report on ice detection guidelines provides extensive details on the various technologies and their measuring principles [12].

Contrary to the direct rotor icing detectors reviewed above, *indirect approaches* rely on standard SCADA data, without the need for specialized sensing technologies. We distinguish between two clusters of indirect approaches, which are fundamentally different in essence. The first cluster is what we refer to as "power curve monitoring approaches," where the goal is to pre-establish a non-iced power curve that characterizes the power output of a wind turbine under normal operational conditions (see for example the blue curve in Figure 1), and then monitor the incoming observations for significant deviations from that benchmark [19, 20, 21, 22, 23]. A relevant example is the T19iceloss method proposed by the IEA's Task 19, which entails a set of power-curve-derived rules that flags the power output observations that fall outside certain thresholds of the non-iced power curve [24]. Along the same line, methods based on statistical process monitoring and residual diagnostics have also been explored. Albeit not tailored to icing detection, those monitoring methods are designed to signal any statistically significant divergence from the pre-established power curve baseline [25, 26, 27, 28].

The second cluster of indirect approaches draws on the advancements in machine learning (ML) to learn the unique signatures of icing events from historical wind turbine data. Unlike the first cluster, those methods may or may not involve a dedicated step of constructing a non-iced power curve, and typically entail a supervised learning model (e.g., a classifier) which directly acts on SCADA data to distinguish icing from non-icing events. Among those, wavelet-based methods [29], deep learning [30, 31, 32, 33], transfer learning [34], contrastive learning [35], tree- and kernel-based classifiers [36, 37, 38, 39, 40, 41, 42] have been explored. Few recent studies further investigated the potential of ML-based approaches in directly acting on image data, possibly in conjunction with SCADA measurements [43, 44, 45].

For a wind farm operator, the decision of whether to adopt a direct or indirect approach (or both) depends on several factors, including the goal of the icing monitoring task (e.g., performance loss estimation, anti- and de-icing mitigation, ice fall and throw risk assessment), the desired level of sensitivity, budget and resource constraints, among others. One particular advantage of indirect approaches is their marginal operational cost since they operate on SCADA data that are already collected by any modern wind turbine (i.e., they do not require any sensor installations or retrofits). Nevertheless, indirect approaches come with their own set of challenges. In specific, to unlock the value of an indirect approach, one needs to address the following key question: *how to "mine" the complex and hidden signatures of icing events from large streams of multivariate (and often noisy) SCADA data?*" Despite the recent advancements in this space, our review of the literature (See Table 1 for a representative set of related works) reveals three key areas of improvement, which constitute the focus of this paper:

- *Modeling auto- and cross-correlations in SCADA data:* SCADA data is naturally structured as a multivariate time series (i.e. multiple variables or "channels" observed over time), and hence, naturally exhibit auto- and cross-correlations. Auto-correlations refer to temporal dependence *within* a channel. Cross-correlations, on the other hand, refer to the inter-relationship across *distinct* channels. Those complex dependencies, if effectively modeled, can furnish a desirable "learning bias" for ML methods to enhance their ability to learn and predict icing events.

- *Injecting power curve physics:* The large majority of ML-based icing monitoring methods frame the learning task as a classification problem that directly acts on SCADA data as input features. This approach does not fully leverage the well-established knowledge about the physics of wind power generation (encoded in a power curve), and hence may be prone to mistake the systematic, weather-driven variations in a turbine's power output as icing occurrences. On the other hand, power curve monitoring approaches, by design, encode the power curve physics in their pre-established, non-iced benchmark, but do not fully unleash the "learning via data" aspect of ML. A hybrid approach, like the one pursued in this paper, can potentially harness the strengths of both paradigms for enhanced monitoring capabilities.

- *A unified framework for icing detection and prediction:* Most existing approaches are primarily focused on detecting icing events *after* they

Table 1: The contribution of the proposed framework relative to a representative subset of the icing monitoring literature. In the last column, “A” denotes auto-correlations, whereas “C” denotes cross-correlations.

Research Effort	Detection	Prediction	No hardware required	Power curve encoded	Dependence modeling
[16]	✓	-	-	-	-
[17]	✓	-	-	-	-
[19]	✓	-	✓	✓	-
[20]	✓	-	✓	✓	-
[21]	✓	-	✓	✓	-
[23]	✓	-	✓	✓	-
[29]	✓	-	✓	-	A
[30]	-	✓	✓	-	C
[31]	✓	-	✓	-	A, C
[32]	✓	-	✓	-	C
[36]	✓	-	✓	-	C
[40]	-	✓	✓	-	C
[46]	-	✓	✓	-	C
Proposed	✓	✓	✓	✓	A, C

occur. A relatively under-explored area is to anticipate rotor icing *ex-ante* [30, 46]. Unlike existing efforts that either focus on the detection or prediction aspects of the problem, we propose a unified framework that can be easily tailored to operate under any (or both) of those two modes depending on the operator’s preference, without sacrificing the overall monitoring system effectiveness.

With the above three issues in mind, we propose an ML-based approach, which we call the Tensor-based IcinG dEtection and pRediction method, or in short TIGER. A defining aspect of our framework is to regard the SCADA data as a series of tensors (a high-dimensional generalization of a matrix). This enables us to leverage tensor projection approaches to condense the high-dimensional multivariate SCADA time series into a low-dimensional set of descriptors that encapsulates rich information about their complex auto- and cross-dependencies. Those condensed descriptors are then combined with power-curve-derived features that encode the power curve physics, and the combined set is used as input to drive a deep learning model that effectively signals rotor icing occurrences and estimates icing probabilities. We also propose a new loss measure, which we dub as the icing power loss error (IPLE). Motivated by the lack of a unified mechanism to assess icing monitoring methods, IPLE provides a realistic and rigorous quantification of icing-related power losses, beyond simple classification metrics (like accuracy), thereby furnishing a valuable evaluation tool for wind farm operators to benchmark and compare data-driven icing monitoring solutions.

The remainder of this paper is organized as follows. Section 2 introduces the problem and presents two possible operational modes for TIGER: diagnostic and prognostic and the building blocks of TIGER, which is followed by Section 3 where evaluations, results, and discussions are presented. Finally, Section 4 concludes the paper.

2. Methodology

2.1. Problem Motivation and Definition

We assume that at the current time t, the farm operator has access to historical wind turbine data of k variables (e.g., wind power, wind speed, temperature), all the way back till $t-h+1$, where $h \in \mathbb{Z}^+$ is a time window of relevance. Specifically, we denote by $\mathbf{S} \in \mathbb{R}^{k\times h} := [\boldsymbol{s}_{t-h+1}, \ldots, \boldsymbol{s}_t]$ the multivariate time series formed by the $k \times h$ matrix for which the (l,j)th entry constitutes the jth measurement of the lth variable, such that $l \in \{1, ..., k\}$ and $j \in \{t-h+1, ..., t\}$. In addition, we assume that the wind farm operator has access to historical icing records encoded in the vector $\mathbf{y} := [y_{t-h+1}, ..., y_t]^T$, such that $y_i \in \{0, 1\}$ denotes the occurrence of an icing event at time i. Next, we present two modes of operation for TIGER: a diagnostic and a prognostic mode.

2.1.1. Diagnostic mode: Icing detection

Here, the goal is to timely detect an icing event as soon as it occurs (i.e., in real- or near-real-time). In other words, we seek to determine whether $\boldsymbol{s}_t$ (the k-dimensional vector representing the hth column of $\mathbf{S}$) corresponds to an icing event. In mathematical terms, we seek a functional mapping $f(\cdot)$, which optimally assigns labels (or probabilities) on the basis of the SCADA observations in $\mathbf{S}$ and the icing records up to $t-1$:

$$f(\boldsymbol{s}_t|\mathbf{S}, y_{t-h+1}, ..., y_{t-1}) \rightarrow y_t \in \{0, 1\}. \tag{1}$$

2.1.2. Prognostic mode: Icing prediction

Unlike the diagnostic mode, we here attempt to anticipate an icing event *before* it actually occurs. In addition to using historical observations, future forecasts about the SCADA variables are assumed to be available. In other words, we assume the operator has access to a larger input matrix $\hat{\mathbf{S}} \in \mathbb{R}^{k\times(h+u)} := [\boldsymbol{s}_{t-h+1}, \ldots, \boldsymbol{s}_t, \hat{\boldsymbol{s}}_{t+1}, \ldots, \hat{\boldsymbol{s}}_{t+u}]$, where $u \in \mathbb{Z}^+$ is the forecast horizon, and $\hat{\boldsymbol{s}}_{t+q}$ denotes a forecast of the true (but unknown) value of $\boldsymbol{s}_{t+q}$, such that $q \in \{1, ..., u\}$. In the wind industry, those forecasts are typically available to the wind farm operator and are obtained using statistical methods [47], numerical weather predictions [48], or a combination thereof [49]. The prognostic problem is similar to the diagnostic problem except that it uses $\hat{\mathbf{S}}$ instead of $\mathbf{S}$, and that it assumes y_t has already been observed:

$$f(\hat{\boldsymbol{s}}_{t+q}|\hat{\mathbf{S}}, y_{t-h+1}, ..., y_t) \rightarrow y_{t+q} \in \{0, 1\} \quad \forall q. \tag{2}$$

Next, we start by a brief overview of tensor projection notation, then proceed to introduce the TIGER framework.

2.2. Low-rank Tensor Projection

A tensor is the generalization of a matrix to a higher dimension, and is typically described in terms of its order (or rank) and mode(s). An N-order tensor can be represented as $\mathcal{T} \in \mathbb{R}^{I_1\times\ldots\times I_n\times\ldots\times I_N}$, such that I_n is the dimension of the nth mode. Since tensors, by definition, are higher-dimensional objects, tensor projection is the task of representing a tensor in terms of lower-dimensional objects (e.g., simpler tensors, vectors, scalars) that encode its most relevant information content. One of the simplest forms of tensor projection is the so-called elementary multilinear projection (EMP), which, as shown in (3), projects $\mathcal{T}$ into a scalar x through N unit projection vectors denoted by $\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \ldots, \boldsymbol{v}^{(N)}$.

$$\begin{gathered} x = \langle \mathcal{T}, \mathcal{V} \rangle = \left\langle \mathcal{T}, \boldsymbol{v}^{(1)} \circ \boldsymbol{v}^{(2)} \circ \ldots \circ \boldsymbol{v}^{(N)} \right\rangle = \\ \mathcal{T} \times_1 \boldsymbol{v}^{(1)^T} \times_2 \boldsymbol{v}^{(2)^T} \ldots \times_N \boldsymbol{v}^{(N)^T}, \\ \left\| \boldsymbol{v}^{(n)} \right\| = 1 \text{ for } n = 1, 2, \ldots, N, \end{gathered} \tag{3}$$

where $\mathcal{V} \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ is the tensor formed by the outer product of the N unit projection vectors, such that $\mathcal{V} = \boldsymbol{v}^{(1)} \circ \boldsymbol{v}^{(2)} \circ \ldots \circ \boldsymbol{v}^{(N)}$. In (3), $\|\cdot\|$ denotes the Euclidean norm, and $\langle \mathcal{T}, \mathcal{V} \rangle$ is the inner product of $\mathcal{T}$ and $\mathcal{V}$, which is defined as:

$$\langle \mathcal{T}, \mathcal{V} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N}^{I_N} t_{i_1 i_2 \ldots i_N} v_{i_1 i_2 \ldots i_N}. \tag{4}$$

In many settings, one has to analyze a set of tensor *samples* denoted by $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_M\}$. For example, in image processing, tensor samples may correspond to different images collected over time, where each image is regarded as a three-order tensor. The uncorrelated multilinear principal component analysis (UMPCA) is a tensor-to-vector (TVP) projection approach [50, 51, 52] which maps the mth tensor sample $\mathcal{T}_m$, where $m \in \{1, ..., M\}$, into a vector subspace $\mathbb{R}^L$ ($L < \prod_{n=1}^{N} I_n$) through a series of L EMPs:

$$\boldsymbol{f}_m = \mathcal{T}_m \times_{n=1}^{N} \left\{ \boldsymbol{v}_\ell^{(n)^T}, n = 1, \ldots, N \right\}_{\ell=1}^{L} \quad \forall m, \tag{5}$$

where $\boldsymbol{f}_m \in \mathbb{R}^L$ denotes the vector of feature values extracted by UMPCA, such that its ℓth entry, $f_m(\ell)$, is the projection of $\mathcal{T}_m$ via the ℓth EMP: $\left\{ \boldsymbol{v}_\ell^{(n)^T}, n = 1, \ldots, N \right\}$, as in (6).

$$f_m(\ell) = \mathcal{T}_m \times_{n=1}^{N} \left\{ \boldsymbol{v}_\ell^{(n)^T}, n = 1 \ldots, N \right\}. \tag{6}$$

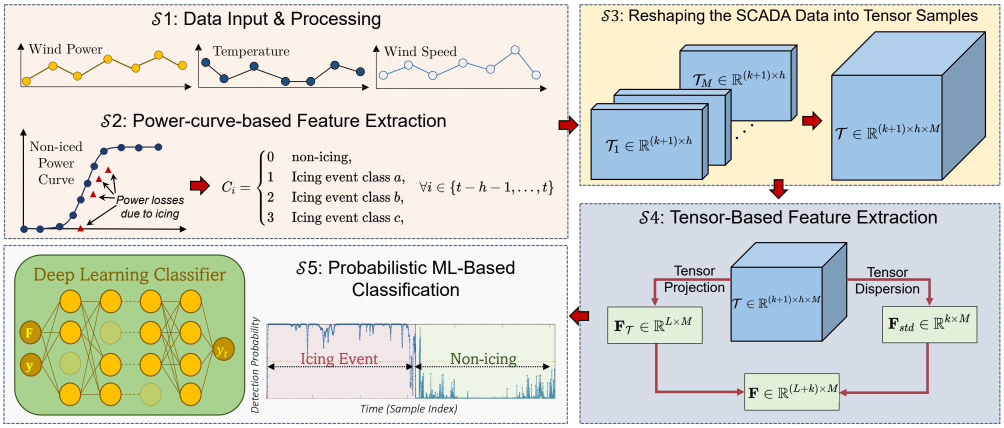
Similar in concept to principal component analysis (PCA), UMPCA finds the L most "informative" EMPs (or projection vectors) $\left\{ \boldsymbol{v}_\ell^{(n)^T}, n = 1, \ldots, N \right\}_{l=1}^{L}$ that maximize the projected variance, while ensuring the extracted features are uncorrelated:

$$\begin{aligned} \left\{ \boldsymbol{v}_\ell^{(n)^T}, n = 1, \ldots, N \right\} &= \text{argmax} \sum_{m=1}^{M} \left(f_m(\ell) - \bar{f}_\ell \right)^2 \\ \text{subject to} \quad & \boldsymbol{v}_\ell^{(n)^T} \boldsymbol{v}_\ell^{(n)} = 1 \\ & \frac{\boldsymbol{z}_\ell^T \boldsymbol{z}_j}{\|\boldsymbol{z}_\ell\| \, \|\boldsymbol{z}_j\|} = \delta_{\ell j} \\ & \ell, j \in \{1, \ldots, L\}, \end{aligned} \tag{7}$$

where $\bar{f}_\ell = \frac{1}{M} \sum_{m=1}^{M} f_m(\ell)$. In (7), $\boldsymbol{z}_\ell$ denotes the ℓth coordinate vector, such that the mth projected sample using the ℓth EMP is equivalent to the mth element of the ℓth coordinate vectors: i.e., $z_\ell(m) = f_m(\ell)$. The Kronecker delta, $\delta_{\ell j}$, is defined as $\delta_{\ell j} = \mathbb{1}(\ell = j)$, where $\mathbb{1}(\cdot)$ is the indicator function. The optimization in (7) is solved heuristically and results in L projection vectors, such that $L \leq \min\{\min_n\{I_n\}, M\}$, which define the set of tensor-based extracted features, $\boldsymbol{f}_m$.

2.3. The TIGER Framework

We introduce the overall framework of TIGER in Figure 2. The framework comprises five steps in total: ($\mathcal{S}1$) Data input and processing; ($\mathcal{S}2$) Power-curve-based feature extraction; ($\mathcal{S}3$) Reshaping the SCADA data into tensor samples; ($\mathcal{S}4$) Feature extraction via tensor projection; and ($\mathcal{S}5$) Probabilistic classification. The details of each step are outlined below.



$\mathcal{S}1$: Data Input & Processing
Wind Power
Temperature
Wind Speed
$\mathcal{S}2$: Power-curve-based Feature Extraction
Non-iced Power Curve
Power losses due to icing
$C_i = \begin{cases} 0 & \text{non-icing,} \\ 1 & \text{Icing event class } a, \\ 2 & \text{Icing event class } b, \\ 3 & \text{Icing event class } c, \end{cases} \quad \forall i \in \{t-h-1, \ldots, t\}$
$\mathcal{S}3$: Reshaping the SCADA Data into Tensor Samples
$\mathcal{T}_M \in \mathbb{R}^{(k+1)\times h}$
$\mathcal{T}_1 \in \mathbb{R}^{(k+1)\times h}$
$\mathcal{T} \in \mathbb{R}^{(k+1)\times h\times M}$
$\mathcal{S}4$: Tensor-Based Feature Extraction
Tensor Projection
Tensor Dispersion
$\mathcal{T} \in \mathbb{R}^{(k+1)\times h\times M}$
$\mathbf{F}_{\mathcal{T}} \in \mathbb{R}^{L\times M}$
$\mathbf{F}_{std} \in \mathbb{R}^{k\times M}$
$\mathbf{F} \in \mathbb{R}^{(L+k)\times M}$
Deep Learning Classifier
F
y
y_t
$\mathcal{S}5$: Probabilistic ML-Based Classification
Detection Probability
Icing Event
Non-icing
Time (Sample Index)

Figure 2: Framework of TIGER starting from data input and processing ($\mathcal{S}1$); Power-curve related feature extraction ($\mathcal{S}2$); Reshaping the SCADA data into tensor samples ($\mathcal{S}3$); Tensor-based feature extraction ($\mathcal{S}4$); and Probabilistic ML classification ($\mathcal{S}5$).

($\mathcal{S}1$) Data Input & Processing: TIGER takes as input the raw 10-min SCADA data consisting of (at least) the following $k = 3$ variables: wind speed w_s, wind power w_p, and nacelle temperature T_{site} (More variables can be added if available). For the diagnostic mode, only historical data are used. For the prognostic mode, those are augmented by a set of k-dimensional forecasts. For notational consistency, this section presents the diagnostic mode, but all steps performed therein are exactly similar to those in the prognostic mode.

We first perform a density correction for w_s, as recommended in the IEC 61400-12-1 standard [53]:

$$\begin{aligned} w_{\text{site}} &= w_s \times \left(\frac{\frac{P_{\text{std}}}{T_{\text{std}}}}{\frac{P_{site}}{T_{\text{site}}}} \right)^{\frac{1}{3}} \\ &= w_s \times \left(\frac{T_{\text{std}}}{T_{\text{site}}} \left(1 - 2.25577 \times 10^{-5} \times H\right)^{5.25588} \right)^{\frac{1}{3}}, \end{aligned} \tag{8}$$

where w_{site} is the corrected wind speed, T_{std} is the standard temperature of 15°C (288.15 K) corresponding to air density of 1.225 kg/m^3 at sea level $P_{\text{std}} = 101325$ Pa ambient air pressure, and H is the site elevation above sea level. The calibrated wind speed w_{site} now replaces w_s in the input dataset. We refer to this dataset hereinafter as $\mathbf{S} \in \mathbb{R}^{k \times h}$.

($\mathcal{S}$2) Power-curve-based feature extraction: We engineer a subset of features that encode the power curve physics. We first start by constructing a "non-iced" power curve. This is a power curve that is constructed solely using the non-icing data records (Refer to the blue curve in Figure 1 as an example). For that, we use the method of bins, as per the IEC 61400-12-1 standard's recommendation [53], although other multi-input statistical methods can equally be used [54, 55]. The qth percentile of the non-iced power within the dth bin, denoted as $\bar{w}_p^q(d)$, is determined using the non-iced records within that bin.

The IEA T19iceloss method categorizes icing events into three classes, corresponding to three physically meaningful manifestations of rotor icing. To capture those different icing categories, we define a new feature, $C_i \in \mathbb{Z}$, as in (9).

$$C_i = \begin{cases} 0 & \textit{non-icing,} \\ 1 & \textit{Icing event class a,} \\ 2 & \textit{Icing event class b,} \\ 3 & \textit{Icing event class c,} \end{cases} \quad \forall i \in \{t-h+1, ..., t\}. \tag{9}$$

The description of icing event classes a, b, and c, are defined in detail in [11], and are briefly described as follows: (A) *Icing event class a*: If $T_{site} < 0$°C and $w_p < \bar{w}_p^{10}(d)$ for at least 30 minutes, then this signals the start of an icing event class a. If $w_p \geq \bar{w}_p^{10}(d)$ for 30 minutes or longer, icing event class a ends. For all observations between the start and end of this icing event, we set $C_i = 1$. (B) *Icing event class b*: If $T_{site} < 0°\text{C}$ and $w_p < \bar{w}_p^{10}(d)$ for 10 minutes resulting in a shut down (defined as $w_p < 0.5 w_p^r$ for at least 20 minutes), then icing event class b begins, where w_p^r is the turbine's rated power. This typically corresponds with a turbine that has ceased to operate due to icing. This icing event ends when $w_p \geq \bar{w}_p^{10}(d)$ for 30 minutes or longer. For all observations between the start and end of this icing event, we set $C_i = 2$. (C) *Icing event class c*: If $T_{site} < 0°\text{C}$ and $w_p > \bar{w}_p^{90}(d)$ for 30 minutes or more, then icing event class c starts. In practice, this is correspondent to a heated anemometer, which results in overproduction measurements. If $w_p < \bar{w}_p^{90}(d)$ for 30 minutes or more, then this icing event ends. For all observations between the start and end of this icing event, we set $C_i = 3$. (D) *Non-icing event*: For all observations that do not satisfy any of the conditions listed above for classes a, b, or c, we set $C_i = 0$. The augmented dataset including the feature C is now called $\tilde{\mathbf{S}}$ and is a multivariate time series comprising $k+1$ features: w_{site}, w_p, T_{site}, and C.

($\mathcal{S}3$) Reshaping SCADA data into tensor samples: We reshape the multivariate time series $\tilde{\mathbf{S}}$ into a set of tensor samples, $\mathcal{T}_1, ..., \mathcal{T}_M$. As shown in Figure 3, two parameters are used: r which determines the "time length" of each sample (i.e. how many time stamps are included), and $\Delta\mathcal{T}$, which is the inter-sample time between two consecutive tensor samples. For demonstration, Figure 3 depicts $r = 4$ time units and $\Delta\mathcal{T} = 5$. In this paper, we set $r = 12$ intervals of 10-min each (i.e., 2 hours), since this roughly corresponds to the time frame of relevance for operational monitoring of icing events (For example, T19iceloss uses 30-min as a minimal threshold for declaring icing events). We also set $\Delta\mathcal{T} = 1$, so that two consecutive tensor samples overlap in all but one time stamp. This helps us generate the maximum number of tensor samples, M, and hence, a larger dataset for ML training.

($\mathcal{S}4$) Tensor-Based Feature Extraction: The previous step enables us to enact tensor projection approaches in order to extract low-dimensional explanatory features from the multivariate SCADA data. Here, we use UMPCA (See Section 2.1) to compress the mth tensor sample $\mathcal{T}_m$ using (7) into $L = 4$ features. This creates the feature matrix $\mathbf{F}_{\mathcal{T}} \in \mathbb{R}^{L \times M}$. We also engineer additional features related to the within-sample dispersion of each tensor sample, as estimated by the standard deviation of the jth tensor feature (for $j \in \{1, ..., k\}$; we exclude C). We find that those features are able descriptors of icing events. This generates the feature matrix $\mathbf{F}_{\text{std}} \in \mathbb{R}^{k \times M}$. Doing this for $\mathcal{T}_1, ..., \mathcal{T}_m$ forms our final $(L+k)$-feature matrix $\mathbf{F} \in \mathbb{R}^{(L+k) \times M} = [\mathbf{F}_{\mathcal{T}}, \mathbf{F}_{\text{std}}]$.

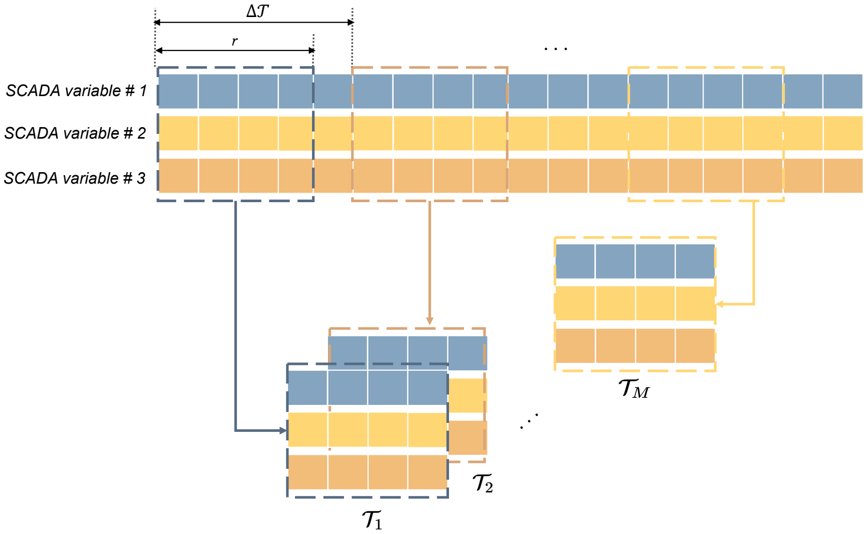


Figure 3: Reshaping $\tilde{\mathbf{S}}$ into tensor samples, $\mathcal{T}_1, ..., \mathcal{T}_M$. The parameters r and $\Delta\mathcal{T}$ dictate the number and dimension of the tensor samples. For demonstration, we have $r = 4$ and $\Delta\mathcal{T} = 5$ in the Figure, although in this work, we use $r = 12$ and $\Delta\mathcal{T} = 1$.

($\mathcal{S}$5) Probabilistic ML Classification: In the final step, we train a classifier to map the feature matrix $\mathbf{F}$ to the icing labels $\mathbf{y}$. Here, we tailor a deep-learning-based classifier, TabNet [56], for the icing detection problem. We provide here a high-level description of TabNet but refer the reader to [56] for more details. TabNet is a recently proposed architecture that consists of multiple sequential steps that feed into each other. Two integral components of TabNet are a feature transformer and an attentive transfer. The feature transformer is composed of a series of blocks (linear and non-linear transformations), each containing fully connected (FC) layers, batch normalization (BN) layers, and gated linear units (GLU) which aids the model in learning complex, non-linear representations of the data. The blocks are stacked together such that each block receives the transformed output of the previous block, allowing for a deep processing of the input features.

Leveraging the refined feature representations from the feature transformer, The attentive transformer then starts with a FC linear transformation, followed by a BN layer, and a sparsemax layer. The output of the attentive transformer then feeds into an attention mask which ensures that only a subset of "relevant" features are effectively used, increasing the learning efficiency and interpertability of the classifier. A linear mapping is used to get the final output mapping.

3. Results and Discussion

We first start by describing the dataset and experimental setup used in this work, then proceed to present and discuss the results of our numerical experiments.

3.1. Data Description and Experimental Setup

The data used in this work is a set of SCADA data that has been synthetically generated by the IEA Task 19 as part of their T19iceloss method development to accelerate and benchmark icing monitoring research [57]. The dataset spans one year and contains 10-min records of wind speed, nacelle temperature, wind direction, and wind power, along with corresponding icing occurrences. The dataset includes two icing events (a total of 1691 icing records), which occurred over a time span of 11.6 days and 3.5 hours, respectively. We used all the available variables from our raw dataset, with the exception of wind direction, that is, $k = 3$. We point out, however, that our approach is able to accommodate more variables if needed and/or available.

We divide the data into training and testing sets based on the proportion of icing records, such that our training data has 70% of the icing records (70% $\times 1691 \approxeq 1184$ icing samples). To avoid class imbalance, we select 1184 non-icing observations (the same as the number of icing observations) for inclusion in the training set. As a result, the training set contained a total of 2368 samples (1184 icing records + 1184 non-icing records). For the test set, we include the remaining icing samples (30% $\times 1691 \approxeq 507$ icing samples), and augment them by a sample of non-icing records that is three times the size of the icing test set ($3 \times 507 = 1521$ non-icing samples). That way, the test set has $n_t = 2028$ samples (507 icing records + 1521 non-icing records).

We compare TIGER to five prevalent data-driven benchmarks that are representative of the literature on data-driven icing monitoring. The first benchmark is a power curve monitoring (PCM) approach which is a variant of the T19iceloss method where we first construct a non-iced power curve, and then classify an observation as an icing record if it belongs to one of the icing classes defined by the T19iceloss method [11]. The second benchmark is an ambient temperature threshold rule, which works as follows: we calculate the 95th percentile of the ambient temperature and set that as the threshold. For a test observation, if the ambient temperature is below the threshold value, we flag it as an icing event, otherwise it is deemed as non-icing. We refer to this temperature-threshold-based benchmark as TTD.

The next three benchmarks represent prevalent ML classifiers which we directly train on the (normalized) raw SCADA data. The first classifier is TabNet [56], which has been explained in Section 2. The two other classifiers are Support Vector Machines (SVM) and Extreme Gradient Boosting (XGBoost) [58]. For SVM, we use an RBF kernel, which is a popular kernel choice, and perform randomized parameter optimization to determine the penalty coefficient, for which the best-fitting value was found to be 16.3. For training TabNet, we utilized grid search to identify the optimal hyperparameters, selecting a dimension of the attention feature layer of 6 and a dimension of the decision prediction layer of 16, with a batch size of 64. The model was trained over 10 epochs using batch gradient descent, with a learning rate of 0.01. For PCM, we construct the non-iced power curve using the full non-icing dataset (without the under-sampling step) since PCM does not suffer from class imbalance as in ML classifiers.

We compare all models via four evaluation metrics that are standard in the ML literature: accuracy, precision, recall, and F_1 score. Accuracy is the fraction of correct predictions, defined as $A(y_i, \hat{y}_i) = \frac{1}{n_t}\sum_{i=1}^{n_t} \mathbb{1}\left(\hat{y}_i = y_i\right)$, where $\mathbb{1}(\cdot)$ is the indicator function, whereas $\hat{y}_i \in \{0, 1\}$ is the ith classified output. Precision, $Pr(y_i, \hat{y}_i)$, measures the proportion of correct positive classifications, whereas recall, $R(y_t, \hat{y}_t)$, evaluates the proportion of icing occurrences that were flagged. If we let tp and fp denote the number of true and false positives, respectively, and fn denote the number of false negatives, then $Pr(y_i, \hat{y}_i) = \frac{tp}{(tp+fp)}$, whereas $R(y_t, \hat{y}_t) = \frac{tp}{tp+fn}$. The F_1 score computes the harmonic mean of precision and recall, such that $F_1(y_i, \hat{y}_i) = 2 \times \frac{Pr(y_i,\hat{y}_i)\times R(y_i,\hat{y}_i)}{Pr(y_i,\hat{y}_i)+R(y_i,\hat{y}_i)}$, to provide an aggregate evaluation of a classifier's ability to balance false and missed alarms.

3.2. Diagnostic Mode Results

Table 2 shows the results of the diagnostic mode. Few key insights can be drawn: First, it appears that ML-based approaches (SVM, XGBoost, TabNet, and TIGER) outperform those based on power curve monitoring (PCM) and the temperature threshold method (TTD). We believe that this is due to the ability of ML-based methods to learn the complex dependencies and unique icing signatures from the historical data. In terms of F_1 score, TIGER outperforms PCM and TTD by 12.7% and 12.2%, respectively.

Table 2: Diagnostic mode: Out-of-sample performance of PCM, TTD, SVM, XGBoost, TabNet, and TIGER. Bold-faced values denote best performance.

	PCM	TTD	SVM	XGBoost	TabNet	TIGER
Accuracy	.901	.898	.926	.943	.943	**.961**
Precision	.749	.724	.783	.841	.839	**.885**
Recall	.911	.959	.970	.953	.957	**.970**
F_1 score	.822	.825	.867	.894	.894	**.926**

Second, the results in Table 2 show that increasing the sophistication of the classifier invoked within TIGER leads to enhanced detection of the icing signatures. This is evident by how XGBoost and TabNet (both state-of-the-art classifers) outperform a traditional kernel-based method like SVM. Finally, the results demonstrate that TIGER, owing to the combination of its ability to extract information-rich features (both power-curve- and tensor-based), along with leveraging a powerful attention-based classifier (TabNet), is able to outperform all methods considered, whether ML- or power-curve-based. In terms of F_1 score, improvements from TIGER range from 6.8% (relative to SVM) to 3.58% relative to TabNet (its closest competitor).

A deeper look into the confusion matrices of the competing methods, shown in Figure 4, further demonstrates the effectiveness of TIGER. Here, TIGER has a significantly lower number of false positives (fp) relative to its competitors (64 FPs, 93 for TabNet, 91 for XGBoost, 136 for SVM, 155 for PCM, and 185 for TTD). In practice, a low rate of false alarms is extremely important for establishing the trust in the monitoring system. TIGER also maintains the lowest number of false negatives (fn), on par with SVM (15 FNs, relative to 22 for TabNet, 24 for XGBoost, 21 for TTD, and 45 for PCM). Understandably, a smaller number of false negatives (i.e., missed alarms) is pivotal to establish the utility of a monitoring system.

Finally, we would like to examine the improvements realized by modeling the auto- and cross-correlations, which is an integral piece of TIGER. To do so, we add an extra benchmark that is similar to TIGER, except that it replaces the tensor-based feature extraction step therein with a PCA-based one acting on the same multivariate time series SCADA data (where PCA refers to Principal Component Analysis). We call this additional benchmark: TabNet$^+$. In Figure 5, we compare two of the most influential raw features, namely, wind speed and ambient temperature (panel a), against PCA-based features used in TabNet$^+$ (panel b), and tensor-based features used in TIGER

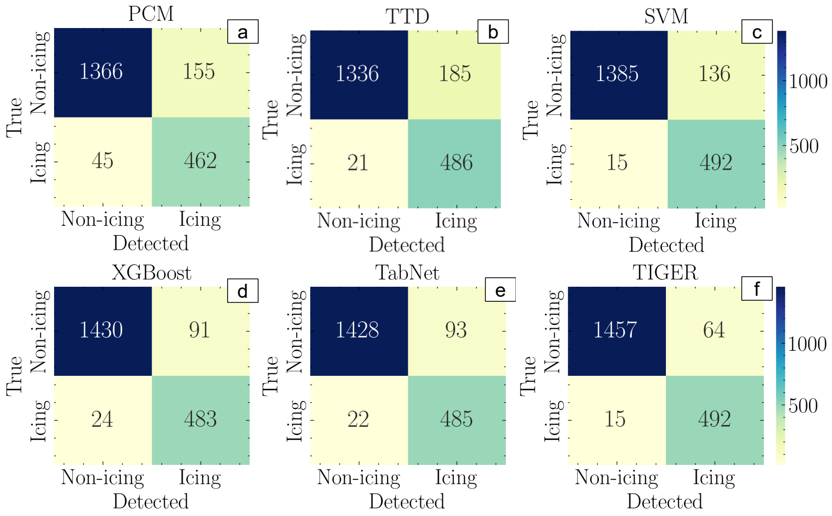


Figure 4: Diagnostic mode results: Confusion matrices for PCM (a), TTD (b), SVM (c), XGBoost (d), TabNet (e), and TIGER (f). Relative to all methods, TIGER yields the best icing detection performance as evident by the low off-diagonal entries.

(panel c). It is evident that the tensor-based features in TIGER exhibit significantly greater separability between the two classes. This distinction explains the enhanced classification performance achieved by TIGER. As a result, TabNet$^+$, which solely relies on PCA for feature extraction, leads to negligible improvement over its base version, TabNet (F_1 score = .895).

3.3. Prognostic Mode Results

For the prognostic mode, we follow the same procedure as in the diagnostic mode, except that, we use the data matrix $\hat{\mathbf{S}}$ (instead of $\mathbf{S}$) which comprises both the SCADA measurements up to time t, as well as u-hour ahead forecasts of w_s, w_p, and T_{site}. Another difference is that the definition of what constitutes an icing record is slightly different. In the diagnostic mode, we have $y_i = 1$ when an icing event occurs at the ith time stamp. However, in the prognostic mode, we have $y_i = 1$ when an icing event occurs at any time during the interval $[i, i+u]$, where $u \in \mathbb{Z}^+$ is the forecast horizon.

Here, we focus on short-term forecasting (i.e., 1-hour ahead in 10-min resolution). To mimic the impact of forecast errors, we add a zero-mean Gaussian noise to the true values (assumed to be unknown) of the SCADA variables. Specifically, we assume $\hat{\mathbf{s}}_{t+q} = \mathbf{s}_{t+q} + \mathbf{e}_{t+q} \; \forall \; q \in \{1, ..., u\}$, where

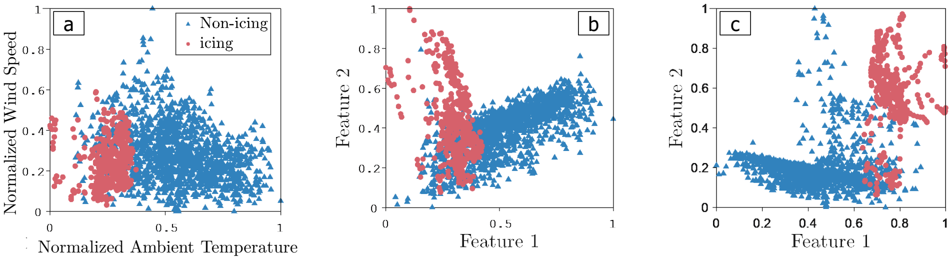


Figure 5: Illustration of normalized features. (a): Raw features; (b) PCA-based features; (c) Tensor-based Features. The latter demonstrate greater explanatory power in distinguishing icing from non-icing classes.

$\mathbf{e}_{t+q}$ is the vector of forecast errors at $t+q$, generated from $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{t+q})$, where $\mathbf{\Sigma}_{t+q} = \begin{bmatrix} 0.1\sigma_{w_{\text{site}}} & 0 & 0 \\ 0 & 0.1\sigma_{T_{\text{site}}} & 0 \\ 0 & 0 & 0.1\sigma_{w_p} \end{bmatrix}$, such that $\sigma_{w_{\text{site}}}$, $\sigma_{T_{\text{site}}}$, and σ_{w_p} are the in-sample standard deviations of w_{site}, T_{site}, and w_p, respectively. We then apply a set of post-processing steps to ensure that the non-negativity of some of the input variables, and to preserve the governing speed-to-power physical conversion relationship (in terms of cut-in, rated, and cut-out speeds). In practice, those simulated forecasts would be replaced by their real-world counterparts which are typically available to the wind farm operator at time t [59].

Table 3 shows the results of the prognostic mode for 1-hour-ahead forecasting, i.e., $u = 60$ minutes Again, TIGER is shown to noticeably outperform its competitors across all metrics, attesting to its merit in not only detecting icing events, but also anticipating them *ex-ante*. Comparing the performance in the short-term prediction task shown in Table 3 relative to that in the detection task shown in Table 2 suggests that, with the exception of TabNet, all the methods that do no encode the power curve physics (namely, TTD, SVM, XGBoost) are impacted by forecast errors, i.e., they all experienced a drop in performance in accuracy and F_1 score. In contrast, TIGER appears to be robust to forecast errors.

3.4. Estimating Icing Probabilities

A key advantage of TabNet (and other similar classifiers) is our ability to derive icing probabilities using a final sigmoid activation function in its output layer, ensuring that the model's output, $P(y \mid \mathbf{F})$, falls within the range $[0, 1]$. This output represents the probability that a given input $\mathbf{F}$

Table 3: Prognostic mode: Out-of-sample performance of PCM, TTD, SVM, XGBoost, TabNet, and TIGER. Bold-faced values denote best performance.

	PCM	TTD	SVM	XGBoost	TabNet	TIGER
Accuracy	.909	.888	.915	.942	.948	**.964**
Precision	.772	.707	.750	.824	.840	**.878**
Recall	.905	.942	.991	.974	.976	**.994**
F_1 score	.833	.807	.854	.893	.903	**.932**

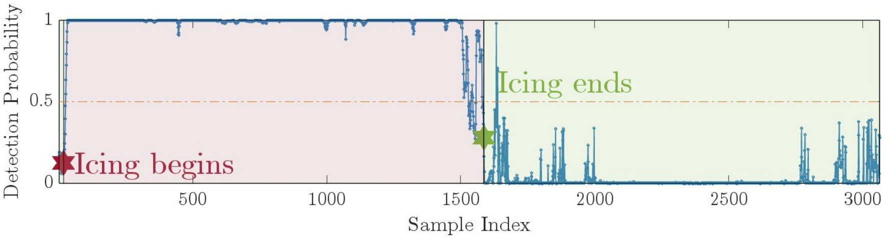


Figure 6: Diagnostic mode: Icing probabilities issued via TIGER. A video demonstrating the evolution of those probabilities is included in the supplementary material (SM-1).

belongs to the positive class y.

Figure 6 shows an illustration of the icing probabilities issued via TIGER, suggesting high confidence in its detection output, wherein the estimated probabilities are typically highest during icing periods (red-shaded background), and lowest during periods of non-icing (green-shaded background). A video demonstrating the evolution of those probabilities over time is included in the supplementary material appended with this article (SM-1). Those probabilistic predictions can be of high value to wind farm operators in informing their icing mitigation decisions. For example, given the projected icing probability, and depending on the operator's attitude towards risk (e.g., averse, tolerant), the operator can either wait on initiating icing mitigation measures until the probability of icing is high enough, or vice-versa.

3.5. Icing-related Production Loss Estimation

Quantifying icing-related production losses is highly relevant for productivity and cost analysis in wind farms [60, 19]. To do so, two relevant metrics are introduced: the detected power loss (PL) and the false power loss (FPL).

Both quantities can be computed using the predicted classes as in (10).

$$PL(\mathcal{C}) = \sum_{i=1}^{n_t} \mathbb{1}[\hat{y}_i(\mathcal{C}) = 1 | y_i = 1] |P_{\text{ref}} - P_{\text{obs}}|, \tag{10}$$

where PL is the total power loss and $\mathcal{C}$ is the classification model (e.g., PCM or TIGER) used to issue the icing detections, $\{\hat{y}_i(\mathcal{C})\}_{i=1}^{n_t}$. The quantities P_{ref} and P_{obs} denote the reference non-iced power and the observed power values, respectively. Similarly, we compute the FPL as follows:

$$FPL(\mathcal{C}) = \sum_{i=1}^{n_t} \mathbb{1}[\hat{y}_i(\mathcal{C}) = 1 | y_i = 0] |P_{\text{ref}} - P_{\text{obs}}|. \tag{11}$$

For an effective monitoring system, $PL(\mathcal{C})$ would be as close as possible to the actual power loss, denoted by PL_{truth}, and obtained by replacing $\hat{y}_i(\mathcal{C})$ by the true values, i.e., y_i $\forall i$ in (10), while $FPL(\mathcal{C})$ would be as minimal as possible (zero would correspond to no false positives).

Making use of $PL(\mathcal{C})$ and $FPL(\mathcal{C})$, we propose a new loss measure which can be used by operators to compare different icing monitoring systems. We call it the icing power loss error (IPLE) and is defined as:

$$IPLE(\mathcal{C}) = \alpha \underbrace{|PL(\mathcal{C}) - PL_{truth}|}_{\text{Detected Power Loss Error}} + \underbrace{(1-\alpha)FPL(\mathcal{C})}_{\text{False Power Loss}}, \tag{12}$$

where the first term estimates the difference between the detected power losses by the classifier versus the actual power losses, whereas the second term represents the false power losses due to incorrect detections by the classifier. The coefficient $\alpha \in [0, 1]$ dictates the weight assigned to error types I and II (false vs. missed alarm), depending on the operator's preference.

Using $\alpha = 0.5$, Figure 7 compares various icing detection methods in terms of the power loss detection error, the false power loss, and the total icing-related power loss error (IPLE). It is clear that TIGER provides significant margins of improvement relative to its competitors in accurately estimating icing-related power losses. In specific, TIGER reduces the power loss estimation error by 18.72% relative to XGBoost (its closest competitor).

4. Conclusion and Future Directions

Rotor icing causes significant power losses as well as potential reliability and safety hazards in wind farms. An ML-based icing monitoring framework,

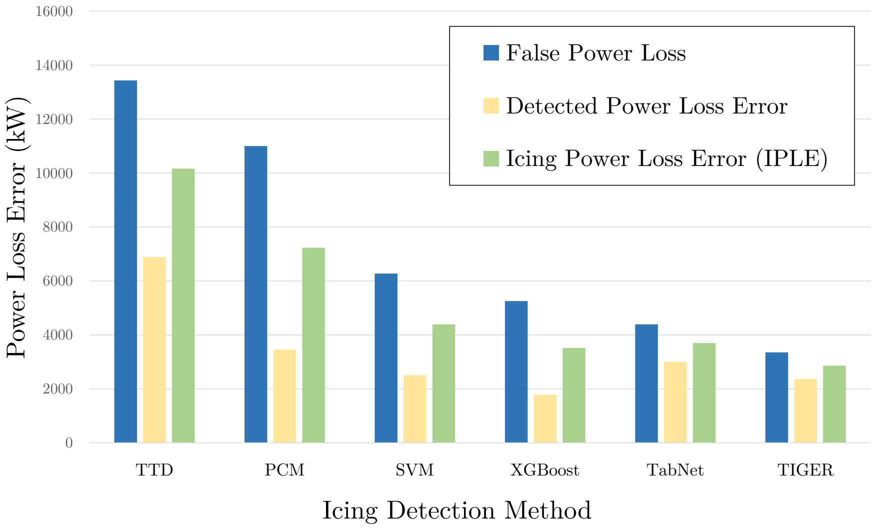


Figure 7: Estimation of power losses for various icing detection methods. TIGER achieves the lowest errors in terms of power loss detection, false power losses, and total icing-related power loss (IPLE).

called TIGER, has been proposed in this paper. By reshaping the SCADA data into tensor samples, TIGER is able to project high-dimensional multivariate time series into a subset of low-dimensional, information-rich features. When combined with other physically meaningful power-curve-derived predictors, the integrated feature set is used to drive a probabilistic deep learning model that learns the signatures of icing events and estimates icing probabilities and power losses. The superiority and robustness of TIGER in detecting and predicting rotor icing have been demonstrated using extensive numerical experiments, as well as through a newly proposed loss measure which realistically and rigorously quantifies the icing-related power losses.

This work can be extended in several ways—Two of which are briefly listed below. The first direction is to explore how our approach can be extended for longer-term icing predictions. Here, we only focused on short-term horizons but longer horizons (e.g., a day ahead) are relevant for other operational purposes like electricity markets. For example, in the 2021 TX blackout, market prices surged by $9,000$ times their pre-storm levels. Extending the forecast horizon would indeed go beyond purely data-driven approaches, and instead require the careful development of icing-adjusted physical-statistical models that make use of tailored numerical weather predictions [61, 62]. The second direction is to integrate the proposed method within a decision-theoretic framework for maintenance optimization [63] to rigorously assess the economic value of effective icing monitoring on key operational metrics, including system cost, downtime, and reliability.

Supplementary Materials

SM-1 is a video showing the evolution of rotor icing probabilities during icing and non-icing periods.

Acknowledgments

This work has been supported in part by the National Science Foundation under Grant # ECCS-2114422.

References

[1] IEA, Renewables 2023, `https://www.iea.org/reports/renewables-2023` (2024).

[2] D. Zhang, Z. Xu, C. Li, R. Yang, M. Shahidehpour, Q. Wu, M. Yan, Economic and sustainability promises of wind energy considering the impacts of climate change and vulnerabilities to extreme conditions, The Electricity Journal 32 (6) (2019) 7–12.

[3] P. Veers, K. Dykes, R. Baranowski, C. Bay, P. Bortolotti, P. Doubrawa, S. MacDonald, S. Rooney, C. L. Bottasso, P. Fleming, et al., Grand challenges revisited: Wind energy research needs for a global energy transition, Tech. rep., National Renewable Energy Laboratory (NREL), Golden, CO (United States) (2023).

[4] P. Papadopoulos, D. Coit, A. Ezzat, STOCHOS: Stochastic opportunistic maintenance scheduling for offshore wind farms, IISE Transactions (2022) 1–15.

[5] O. Fakorede, Z. Feger, H. Ibrahim, A. Ilinca, J. Perron, C. Masson, Ice protection systems for wind turbines in cold climate: characteristics, comparisons and analysis, Renewable and Sustainable Energy Reviews 65 (2016) 662–675.

[6] L. Gao, H. Hu, Wind turbine icing characteristics and icing-induced power losses to utility-scale wind turbines, Proceedings of the National Academy of Sciences 118 (42) (2021) e2111461118.

[7] A. Krenn, A. Stök, N. Weber, S. Barup, T. Weidl, A. Hoffmann, R. E. Bredesen, M. Lannic, S. Müller, N. Stoffels, et al., IEA Wind TCP Task 19: International recommendations for ice fall and ice throw risk assessments, Tech. rep. (2018).

[8] S. Kolar, A comparison of wind power production with three different de-and anti-icing systems, Master's thesis, Uppsala University (2015).

[9] J. Zhou, D. W. Coit, F. A. Felder, S. Tsianikas, Combined optimization of system reliability improvement and resilience with mixed cascading failures in dependent network systems, Reliability Engineering & System Safety 237 (2023) 109376.

[10] P. Veers, B. Kroposki, J. Novacheck, V. Gevorgian, D. Laird, Y. Zhang, D. Corbus, M. Baggu, B. Palmintier, S. Dhulipala, Examination of the extreme cold weather event affecting the power system in Texas - February 2021, Tech. rep., National Renewable Energy Lab (NREL), Golden, CO (United States) (2021).

[11] IEA, IEA Wind Task 19, `https://iea-wind.org/task19/` (2018).

[12] C. Godreau, Y. Paquet, P. Froidevaux, A. Krenn, H. Wickman, IEA Task 19: Ice detection guidelines for wind energy applications, Tech. rep. (2019).

[13] N. Davis, Z. Khadiri-Yazami, R. E. Bredesen, A. Kjeller Vindteknikk, N. G. Ronsten, H. Wickman, S. Bourgeois, Available technologies for wind energy in cold climates–report, IEA Wind Task: Paris, France 19 (2018) 38–62.

[14] METEOTEST, Evaluation of ice detection systems for wind turbines, Tech. rep. (2016).

[15] D. Brenner, 1500 operational years of icing on wind turbines—a long term study, in: Proceedings of the Winterwind, International Wind Energy Conference, Piteå, Sweden, 2015, pp. 3–4.

[16] M. Moser, T. Schlegl, H. Zangl, Retrofittable, autonomous and wireless icing and temperature monitoring on rotor blades for efficient anti-and de-icing, in: Winterwind 2014 International Wind Energy Conference, Québec, QC, Canada, Vol. 11, 2014.

[17] C. Q. Gómez Muñoz, F. P. García Márquez, J. M. Sánchez Tomás, Ice detection using thermal infrared radiometry on wind turbine blades, Measurement 93 (2016) 157–163.

[18] A. A. Jiménez, F. P. G. Márquez, V. B. Moraleda, C. Q. G. Muñoz, Linear and nonlinear features and machine learning for wind turbine blade ice detection and diagnosis, Renewable energy 132 (2019) 1034–1048.

[19] T. Karlsson, V. Turkia, T. Wallenius, J. Miettinen, Production loss estimation for wind power forecasting, Proceedings of Winterwind (2014).

[20] N. N. Davis, Ø. Byrkjedal, A. N. Hahmann, N.-E. Clausen, M. Žagar, Ice detection on wind turbines using the observed power curve, Wind Energy 19 (6) (2016) 999–1010.

[21] N. N. Davis, P. Pinson, A. N. Hahmann, N.-E. Clausen, M. Žagar, Identifying and characterizing the impact of turbine icing on wind farm power generation, Wind Energy 19 (8) (2016) 1503–1518.

[22] B. Golparvar, P. Papadopoulos, A. A. Ezzat, R.-Q. Wang, A surrogate-model-based approach for estimating the first and second-order moments of offshore wind power, Applied Energy 299 (2021) 117286.

[23] X. Dong, D. Gao, J. Li, Z. Jincao, K. Zheng, Blades icing identification model of wind turbines based on SCADA data, Renewable Energy 162 (2020) 575–586.

[24] IEA, T19icelossmethod: A standardized method for assessment of production losses due to icing from wind turbine SCADA data, `https://iea-wind.org/task19/t19icelossmethod/` (2019).

[25] A. Kusiak, H. Zheng, Z. Song, On-line monitoring of power curves, Renewable Energy 34 (6) (2009) 1487–1493.

[26] N. Yampikulsakul, E. Byon, S. Huang, S. Sheng, M. You, Condition monitoring of wind power system with nonparametric regression analysis, IEEE Transactions on Energy Conversion 29 (2) (2014) 288–299.

[27] M. Schlechtingen, I. F. Santos, S. Achiche, Using data-mining approaches for wind turbine power curve monitoring: A comparative study, IEEE Transactions on Sustainable Energy 4 (3) (2013) 671–679.

[28] J.-Y. Park, J.-K. Lee, K.-Y. Oh, J.-S. Lee, Development of a novel power curve monitoring method for wind turbines and its field tests, IEEE Transactions on Energy Conversion 29 (1) (2014) 119–128.

[29] S. A. Saleh, R. Ahshan, C. R. Moloney, Wavelet-based signal processing method for detecting ice accretion on wind turbines, IEEE Transactions on Sustainable Energy 3 (3) (2012) 585–597.

[30] M. Kreutz, A. Ait-Alla, K. Varasteh, S. Oelker, A. Greulich, M. Freitag, K.-D. Thoben, Machine learning-based icing prediction on wind turbines, Procedia CIRP 81 (2019) 423–428, 52nd CIRP Conference on Manufacturing Systems.

[31] B. Yuan, C. Wang, C. Luo, F. Jiang, M. Long, P. S. Yu, Y. Liu, WaveletAE: A wavelet-enhanced autoencoder for wind turbine blade icing detection, arXiv preprint arXiv:1902.05625 (2019).

[32] Y. Liu, H. Cheng, X. Kong, Q. Wang, H. Cui, Intelligent wind turbine blade icing detection using supervisory control and data acquisition data and ensemble deep learning, Energy Science & Engineering 7 (6) (2019) 2633–2645.

[33] X. Cheng, F. Shi, M. Zhao, G. Li, H. Zhang, S. Chen, Temporal attention convolutional neural network for estimation of icing probability on wind turbine blades, IEEE Transactions on Industrial Electronics 69 (6) (2021) 6371–6380.

[34] W. Chen, Y. Qiu, Y. Feng, Y. Li, A. Kusiak, Diagnosis of wind turbine faults with transfer learning algorithms, Renewable Energy 163 (2021) 2053–2067.

[35] Z. Wang, B. Qin, H. Sun, J. Zhang, M. D. Butala, C. Demartino, P. Peng, H. Wang, An imbalanced semi-supervised wind turbine blade

icing detection method based on contrastive learning, Renewable Energy 212 (2023) 251–262.

[36] T. Tao, Y. Liu, Y. Qiao, L. Gao, J. Lu, C. Zhang, Y. Wang, Wind turbine blade icing diagnosis using hybrid features and stacked-XGBoost algorithm, Renewable Energy 180 (2021) 1004–1013.

[37] P. Guo, D. Infield, Wind turbine blade icing detection with multi-model collaborative monitoring method, Renewable Energy 179 (2021) 1098–1105. doi:https://doi.org/10.1016/j.renene.2021.07.120.

[38] X. Bai, T. Tao, L. Gao, C. Tao, Y. Liu, Wind turbine blade icing diagnosis using RFECV-TSVM pseudo-sample processing, Renewable Energy 211 (2023) 412–419.

[39] A. A. Jiménez, F. P. García Márquez, V. B. Moraleda, C. Q. Gómez Muñoz, Linear and nonlinear features and machine learning for wind turbine blade ice detection and diagnosis, Renewable Energy 132 (2019) 1034–1048.

[40] Y. Ge, D. Yue, L. Chen, Prediction of wind turbine blades icing based on MBK-SMOTE and random forest in imbalanced data set, in: 2017 IEEE Conference on Energy Internet and Energy System Integration (EI2), IEEE, 2017, pp. 1–6.

[41] Z. Guangfei, T. Wen, Z. Da, Ice detection for wind turbine blades based on PSO-SVM method, in: Journal of Physics: Conference Series, Vol. 1087, IOP Publishing, 2018, p. 022036.

[42] L. Zhang, K. Liu, Y. Wang, Z. B. Omariba, Ice detection model of wind turbine blades based on random forest classifier, Energies 11 (10) (2018) 2548.

[43] M. Kreutz, A. A. Alla, A. Eisenstadt, M. Freitag, K.-D. Thoben, Ice detection on rotor blades of wind turbines using RGB images and convolutional neural networks, Procedia CIRP 93 (2020) 1292–1297.

[44] K. Hacıefendioğlu, H. B. Başağa, Z. Yavuz, M. T. Karimi, Intelligent ice detection on wind turbine blades using semantic segmentation and class activation map approaches based on deep learning method, Renewable Energy 182 (2022) 1–16.

[45] X. Yang, Y. Zhang, W. Lv, D. Wang, Image recognition of wind turbine blade damage based on a deep learning model with transfer learning and an ensemble learning classifier, Renewable Energy 163 (2021) 386–397.

[46] S. Scher, J. Molinder, Machine learning-based prediction of icing-related wind power production loss, IEEE Access 7 (2019) 129421–129429.

[47] A. A. Ezzat, M. Jun, Y. Ding, Spatio-temporal asymmetry of local wind fields and its impact on short-term wind forecasting, IEEE Transactions on Sustainable Energy 9 (3) (2018) 1437–1447.

[48] M. Optis, A. Kumler, G. N. Scott, M. C. Debnath, P. J. Moriarty, Validation of RU-WRF, the custom atmospheric mesoscale model of the Rutgers Center for Ocean Observing Leadership, Tech. rep., National Renewable Energy Laboratory (NREL), Golden, CO (United States) (2020).

[49] F. Ye, J. Brodie, T. Miles, A. A. Ezzat, Airu-wrf: A physics-guided spatio-temporal wind forecasting model and its application to the us mid atlantic offshore wind energy areas, Renewable Energy (2024) 119934.

[50] H. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, Uncorrelated multilinear principal component analysis through successive variance maximization, in: Proceedings of the 25th international conference on Machine learning, 2008, pp. 616–623.

[51] K. Paynabar, J. Jin, M. Pacella, Monitoring and diagnosis of multichannel nonlinear profile variations using uncorrelated multilinear principal component analysis, IIE Transactions 45 (11) (2013) 1235–1247.

[52] F. Ye, Z. Zhang, Z. Xia, Y. Zhou, H. Zhang, Monitoring and diagnosis of multi-channel profile data based on uncorrelated multilinear discriminant analysis, The International Journal of Advanced Manufacturing Technology 103 (9) (2019) 4659–4669.

[53] Wind Energy Generation Systems - Part 12-1: Power Performance Measurements of Electricity Producing Wind Turbines, IEC 61400-12-1International Electrotechnical Commission (2017).

[54] Y. Ding, Data Science for Wind Energy, CRC Press, 2019.

[55] P. Nasery, A. A. Ezzat, Yaw-adjusted wind power curve modeling: A local regression approach, Renewable Energy 202 (2023) 1368–1376.

[56] S. Ö. Arik, T. Pfister, Tabnet: Attentive interpretable tabular learning, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 35, 2021, pp. 6679–6687.

[57] IEA, Method for estimating icing losses from wind turbine SCADA data, `https://github.com/IEAWind-Task19/T19IceLossMethod` (2019).

[58] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[59] A. A. Ezzat, M. Jun, Y. Ding, Spatio-temporal short-term wind forecast: A calibrated regime-switching method, The Annals of Applied Statistics 13 (3) (2019) 1484.

[60] E. Hellström, Development of a model for estimation of wind farm production losses due to icing, Tech. rep., Uppsala University (2013).

[61] Ø. Byrkjedal, J. Hansson, H. van der Velde, Development of operational forecasting for icing and wind power at cold climate sites, in: IWAIS 2015–16th International Workshop on Atmospheric Icing of, 2015.

[62] L. Gao, T. Dasari, J. Hong, Wind farm icing loss forecast pertinent to winter extremes, Sustainable Energy Technologies and Assessments 50 (2022) 101872.

[63] P. Papadopoulos, D. Coit, A. A. Ezzat, Seizing opportunity: Maintenance optimization in offshore wind farms considering accessibility, production, and crew dispatch, IEEE Transactions on Sustainable Energy 13 (1) (2021) 111–121.

Appendix

The nomenclature for the key variables and parameters used in this paper is presented in Table 4.

Table 4: Nomenclature

Notation	**Definition**
k	Number of SCADA variables (or channels) considered
y_t	Whether or not an icing event has occurred at time t
$\mathbf{s}_t$	A vector of k SCADA variables at time t
$\hat{\boldsymbol{s}}_{t+u}$	Forecast of the true (but unknown) value of $\boldsymbol{s}_{t+u}$
$\mathbf{S}$	A matrix denoting the multivariate time series of SCADA data
$\hat{\mathbf{S}}$	A matrix denoting the multivariate time series of SCADA data and forecasts
u	Forecast horizon
t	Current time
$\mathcal{T}_m$	mth tensor sample
N	Number of tensor projection vectors
M	Number of tensor samples
$\boldsymbol{v}^{(n)}$	nth unit projection vector
L	Number of elementary multilinear projections (EMPs)
$\boldsymbol{f}_m$	Vector of feature values extracted by UMPCA
$\boldsymbol{z}_\ell$	ℓth coordinate vector
I_n	Dimension of the nth mode of a tensor
w_s	Wind speed
w_p	Wind power
T_{site}	Nacelle temperature
w_{site}	Density-corrected wind speed
T_{std}	Standard temperature
P_{std}	Sea level ambient air pressure
H	Site elevation above sea level
C_i	The value of the power-curve-derived feature at time i
$\bar{w}_p^q(d)$	The qth percentile of the non-iced power within the dth bin
r	Time length of a tensor sample
$\Delta\mathcal{T}$	Inter-sample time between two consecutive tensor samples
$\mathbf{F}_{\mathcal{T}}$	Feature matrix extracted by tensor projection
$\mathbf{F}_{\text{std}}$	Within-sample dispersion feature matrix used in TIGER
$\mathbf{F}$	Final feature matrix used in TIGER
$\mathbf{e}_{t+q}$	Vector of forecast errors at $t+q$
$\sigma_{w_{\text{site}}}$	In-sample standard deviations of w_{site}
$\sigma_{T_{\text{site}}}$	In-sample standard deviations of T_{site}
σ_{w_p}	In-sample standard deviations of w_p
P_{ref}	The reference non-iced power value
P_{obs}	The observed power value
α	Weight of error types I and II in the proposed IPLE measure