

Article

Exploring Quantum Machine Learning and Feature Reduction Techniques for Wind Turbine Pitch Fault Detection

Camila Correa-Jullian ^{1,*}, Sergio Cofre-Martel ¹, Gabriel San Martin ¹, Enrique Lopez Droguett ^{1,2}, Gustavo de Novaes Pires Leite ^{3,4} and Alexandre Costa ⁴

¹ Garrick Institute for the Risk Sciences, University of California, Los Angeles, CA 90095, USA; scofre@umd.edu (S.C.-M.); gsanmartin@ucla.edu (G.S.M.); eald@ucla.edu (E.L.D.)

² Department of Civil and Environmental Engineering, University of California, Los Angeles, CA 90095, USA

³ Federal Institute of Science, Education and Technology Pernambuco (IFPE), Recife 50740-540, PE, Brazil; gustavonovaes@recife.ifpe.edu.br

⁴ Center for Renewable Energy from the Federal University of Pernambuco (CER-UFPE), Recife 50740-540, PE, Brazil; alexandre.acosta@ufpe.br

* Correspondence: ccorreaj@ucla.edu

Abstract: Driven by the development of machine learning (ML) and deep learning techniques, prognostics and health management (PHM) has become a key aspect of reliability engineering research. With the recent rise in popularity of quantum computing algorithms and public availability of first-generation quantum hardware, it is of interest to assess their potential for efficiently handling large quantities of operational data for PHM purposes. This paper addresses the application of quantum kernel classification models for fault detection in wind turbine systems (WTSs). The analyzed data correspond to low-frequency SCADA sensor measurements and recorded SCADA alarm logs, focused on the early detection of pitch fault failures. This work aims to explore potential advantages of quantum kernel methods, such as quantum support vector machines (Q-SVMs), over traditional ML approaches and compare principal component analysis (PCA) and autoencoders (AE) as feature reduction tools. Results show that the proposed quantum approach is comparable to conventional ML models in terms of performance and can outperform traditional models (random forest, k-nearest neighbors) for the selected reduced dimensionality of 19 features for both PCA and AE. The overall highest mean accuracies obtained are 0.945 for Gaussian SVM and 0.925 for Q-SVM models.

Keywords: quantum machine learning; quantum kernels; wind turbine systems; SCADA system; pitch fault diagnostics; feature reduction; principal component analysis; autoencoders; machine learning; prognostics and health management



Citation: Correa-Jullian, C.; Cofre-Martel, S.; San Martin, G.; Lopez Droguett, E.; de Novaes Pires Leite, G.; Costa, A. Exploring Quantum Machine Learning and Feature Reduction Techniques for Wind Turbine Pitch Fault Detection. *Energies* **2022**, *15*, 2792. <https://doi.org/10.3390/en15082792>

Academic Editors: Mohamed Benbouzid, Sinisa Durovic and Xiandong Ma

Received: 22 March 2022

Accepted: 8 April 2022

Published: 11 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The role of wind turbine systems (WTSs) in decarbonizing the electrical grid has steadily increased in recent years [1]. In 2020, the estimated global cumulative capacity of both onshore and offshore installations was over 743GW. These numbers are expected to grow further in the quest to supply renewable and sustainable energy [2]. A key factor to reduce the levelized cost of energy (LCOE) of wind power is to increase the performance and reliability of these systems [3]. In this context, the implementation of preventive maintenance techniques for WTSs aims to reduce operating expenditures (OPEX) related to unexpected maintenance events, expected to be of critical importance in the case of offshore installations [4]. These variable OPEX can account for 11–30% of the LCOE of onshore installations, up to 30% in offshore installations, and 20–25% of the total LCOE wind power systems [5].

The operation of WTSs depends on a multitude of elements, including external factors such as wind availability and grid stability. Hence, addressing health diagnostics and

prognostics requirements in WTSs is a complex task that depends on system behavior, component degradation, and varying environmental conditions [6]. Several issues may lead to system downtime, including mechanical, electrical, and connectivity failures. A breakdown of common WTS faults per component is presented by Liton Hossain et al. [7]. In particular, pitch system failures may account for up to 20% of total turbine downtime [8]. Determining the cause and identifying early signs of system degradation have proven to be key when developing adequate planning and scheduling of actions to maintain grid stability. Here, the development of prognostics and health management (PHM) frameworks designed for WTS operations can play a major role in deriving comprehensive maintenance policies and increasing system reliability [9]. Complementing traditional reliability methods based on statistical failure event analysis, PHM leverages the collection and analysis of sensor monitoring data to provide diagnostics and prognostics models, aimed at detecting, localizing, and/or predicting future failures and faulty states.

Most WTSs use the supervisory control and data (SCADA) system for their monitoring data collection [10–12]. SCADA is a computer-based system that gathers and processes monitoring data from multiple sensors. Even though the standard sampling rate of SCADA data is 1 s, the reported data correspond to average values over 10 min time intervals, essentially converting the data acquisition into low-frequency measurements [13]. SCADA also records anomalous behavior or system failures detected by a built-in rule-based alarm system. However, due to the high number of false positive alerts, plus the noisy and intractable nature of the generated alarm logs, most researchers have focused on using either the low-frequency SCADA sensor measurements or additional component-specific local high-frequency measurements to develop data-driven diagnostics and prognostics models [14,15]. Indeed, few published works explicitly perform a joint analysis of sensor measurements, alarm logs, and maintenance records [8,14,16]. This is exacerbated by the lack of standardized maintenance reporting procedures in the industry [17,18].

The number of collected sensor signals varies from system to system depending on the WTS, manufacturer, and operators [12,18]. The available sensors and their data quality will determine the possible anomaly detection and diagnostics models that can be trained and their performance. As such, although most of these diagnostics models are based on data collected through the SCADA system, the data preprocessing used in different architectures will likely converge to system-specific solutions. Hence, there is a need for a systematic preprocessing methodology that can be implemented to different systems requiring no or minor adjustments. In this regard, previous studies have addressed some of the global challenges presented in data collected from the SCADA system. For instance, [15] analyzed highly imbalanced SCADA data with the purpose of designing a more accurate alarm system. Here, principal component analysis (PCA) is used to preprocess SCADA sensor data, oversampling techniques are implemented to address class imbalance, and time splits were employed to avoid class contamination.

In WTSs, fault detection and diagnostics tasks have mainly been addressed through three different approaches: model-based, signal-based, and knowledge-based methods [14,19]. In this case, model-based approaches refer to statistical or data-driven models (DDMs). Among these, machine learning (ML) and deep learning (DL) have become powerful alternatives to train diagnostic and prognostic models in the context of PHM frameworks. Popular ML models include support vector machines (SVM), k-nearest neighbors (k-NN), and random forest (RF) algorithms, and they are frequently used for diagnostics tasks in a variety of settings [20–23]. For instance, in [15], a comparative analysis of fault diagnostics with k-NNs and SVM is presented, achieving F1 scores over 0.95 for both models after balancing the datasets for healthy and degraded classes. Furthermore, Stetco et al. [23] presented a comprehensive review of ML models applied for both diagnostics and prognostics in WTSs. It was identified that, for diagnostics tasks, class imbalance and noisy features can hinder model performance, and significant attention must be given to the feature selection and reduction process. Deep learning models have also been implemented to take advantage of their hierarchical structure to extract abstract features from the available data.

Chen et al. [24] implemented an unsupervised anomaly detection model for WTSs based on long short-term memory (LSTM) autoencoders (AEs) using data from the SCADA system. The model is trained based on data considered to be operating at normal states to then be evaluated for new unseen data. Depending on the network's reconstruction error, an adaptive threshold is defined to determine whether the system is operating under normal or anomalous conditions. Encalada-Davila et al. [25] presented another anomaly detection method based on the prediction of one of the sensor variables (i.e., the quantity of interest) based on other selected variables, where a residual is defined as the difference between the sensor reading and the model's predicted value. The model is trained on healthy data, defined as long operational periods where no failures were observed. A faulty state is then defined based on the model's prediction error, where it is expected that faulty states will produce a higher prediction error than a healthy state.

Two important challenges are thus identified in the literature for WTS diagnostics models based on data from the SCADA system. On the one hand, most algorithms are focused on unsupervised or semi-supervised anomaly and fault detection models. This is due to the difficulties of acquiring robust and reliable labels from the system. It is observed that the number of failures is negligible with respect to the available nominal or healthy data (i.e., normal operation) [8]. Therefore, new methodologies to acquire health state labels are required. On the other hand, the selection and implementation of ML and DL approaches have been shown to be difficult. SVM can be unstable when analyzing large multidimensional datasets, while RF tends to overfit over the training sets. DL models are highly complex and require large amounts of data to be trained. Determining the structure of a DL model is challenging given the high number of hyperparameters, and models tend to overfit. However, although ML models also tend to overfit, these may outperform more complex DL architectures under limited data regimes [26]. In this regard, feature extraction and reduction techniques have proven key to analyze smaller datasets.

In this context, quantum computing has been presented as a new computational paradigm, in which computations are performed based on two-state quantum systems, denoted as qubits, instead of traditional bits. Qubits allow for quantum mechanics properties such as interference, entanglement, and superposition to be used in computation routines, in some cases obtaining exponential gains in terms of algorithmic complexity (number of iterations required to perform a certain task). While quantum hardware is still in development, early quantum computers are becoming available for the general public through cloud computing services such as IBM's Quantum Experience. Additionally, specialized software providing high-level APIs to develop quantum algorithms using traditional languages (e.g., Python) have also been released, such as PennyLane [27], Qiskit [28], or Cirq [29]. With these two key developments, researchers and practitioners have been able to test algorithms designed when quantum computing was a theoretical field. A good example of this is Shor's algorithm, proposed originally by Peter Shor in 1994 [30], for efficiently computing the prime factors of integers, which has been recently further explored in the area of cryptography given its relevance as a way to break modern encryption techniques [31].

Recently, the focus of quantum computing research has been shifted into three main areas. The first is the usage of quantum computing to improve existing established algorithms, such as query search or decryption algorithms. Examples of this can be found in [32,33]. The second is the use of quantum properties to accelerate general optimization problems such as the quantum approximate optimization algorithm (QAOA), originally proposed in 2014 by Farhi et al. [34] to solve combinatorial optimization problems. The third area is the use of quantum computing to either improve or accelerate ML models, where research has focused on two different topics: designing quantum circuits that can be identified as neural networks [35] and developing quantum circuits that can be used as kernels for traditional algorithms such as SVMs [36].

Given this context, two research gaps are identified. Firstly, how to objectively select feature reduction techniques when training ML diagnostic and prognostic models based

on SCADA data. Secondly, given the recent advances in quantum machine learning (QML) algorithms, how can quantum kernels be used for PHM purposes and how these compare to traditional ML techniques. This paper discusses the potential of quantum SVMs (Q-SVMs) for system prognostics through a WTS case study. Details on the preprocessing methodology of SCADA sensors data and alarm logs are presented to train a quantum-enabled prognostics model, which is compared with traditional ML algorithms. Here, special attention is given to the feature reduction process through PCA and deep AE. Challenges, advantages, and prospects are discussed when using QML models in PHM.

The main contributions of this paper are the following:

- Development and implementation of a quantum kernel-based fault prognostics model in WTSs;
- Provide a comparative analysis of PCA and AE as feature reduction tools;
- Methodology to obtain health state labels based on SCADA alarm logs;
- Comparison with traditional ML models used for classification tasks.

The remainder of the paper is structured as follows. Section 2 presents a review of current research and challenges in the application of PHM to WTSs. Section 3 presents a detailed introduction to QML and quantum kernels. Section 4 describes in greater detail the WTS case study. Section 5 presents the development of the proposed prognostics models employed in this study and the obtained results, comparing the performance between classical and quantum approaches. Finally, Section 6 presents the main conclusions of this work.

2. Prognostics and Health Management in Wind Turbine Systems

Maintenance activities can be addressed through three different approaches: corrective, preventive, and condition based. Corrective maintenance corresponds to a reactive approach, where failed components are repaired or replaced after they have failed. Preventive maintenance is a more conservative strategy, where maintenance is performed before the component's estimated failure time, frequently based on fixed schedules. Here, maintenance scheduling is based on the component's statistical study, where it is ensured that a certain percentage of the failures are prevented. This approach significantly reduces the number of failures when compared to corrective maintenance; however, it is costly since it frequently results in unnecessary stoppages to perform maintenance in equipment that does not need it. In WTS farms, having these unprofitable stoppages is undesired. In this regard, condition-based maintenance (CBM) uses information from monitoring data collected from sensor networks to infer the health state of the system. This is a dynamic and proactive approach that allows integrating the health state of the system into the optimization of maintenance policies. Integrating the CBM health assessment to the decision-making process is known as PHM.

Prognostics and health management is an approach derived from CBM developed to aid the optimization of maintenance policies. PHM seeks to implement end-to-end frameworks that integrate sensor monitoring data into the decision-making processes, including everything from data acquisition and preprocessing to the training of diagnostics and prognostics models. As it is shown in Figure 1, most PHM frameworks are broadly divided into four different stages: data acquisition, data preprocessing, diagnostics and prognostics, and decision making [37].

In the last decade, research works have focused on obtaining diagnostics and prognostics models to assess the system's state of health. These models are traditionally physics-based models (PBMs), DDMs, or hybrid (i.e., a combination of PBM and DDM). On the one hand, mathematical models are highly accurate and provide interpretability. However, PBMs are rarely available to describe the degradation processes in complex systems. On the other hand, DDMs such as ML and DL techniques, have gained interest since they do not require prior knowledge on the data or system under study and present great generalization capabilities. This comes at the cost of low interpretability due to their black-box behavior and lower precision in their prediction when compared to PBMs. ML applications can be

adapted to study degradation processes at a local scale in components for which mathematical models for the physics of degradation are not available. These require highly precise and localized sensors. These applications are common in additive manufacturing [38]. Another approach considers the discovery of general degradation behavior from operational sensor measurements. This is more suitable for CESs, since knowledge on the degradation behavior is scarce and sensor networks are designed to monitor the operation of the asset rather than for diagnostics or prognostic purposes. In this case, extracting degradation data is a challenging task, since the degradation can occur at any location in the system and not necessarily where the sensors are placed. Sensor networks are also designed to simultaneously monitor several components; thus, the resulting diagnostic models usually focus on system-level degradation rather than local phenomena. Furthermore, hardware development in the last decade has allowed the training of powerful models with millions of data points using graphical power units (GPUs). As such, implementing ML and DL algorithms to obtain diagnostics and prognostics models has become the center of research in PHM. Examples of this are variational autoencoders (VAE) for fault detection [39], deep convolutional neural networks (CNNs) for damage detection and quantification [40,41], deep LSTM and recurrent neural networks (RNN) for quantity of interest prediction and anomaly detection [42,43], and physics-informed neural networks (PINNs) for remaining useful life estimation (RUL) [44].

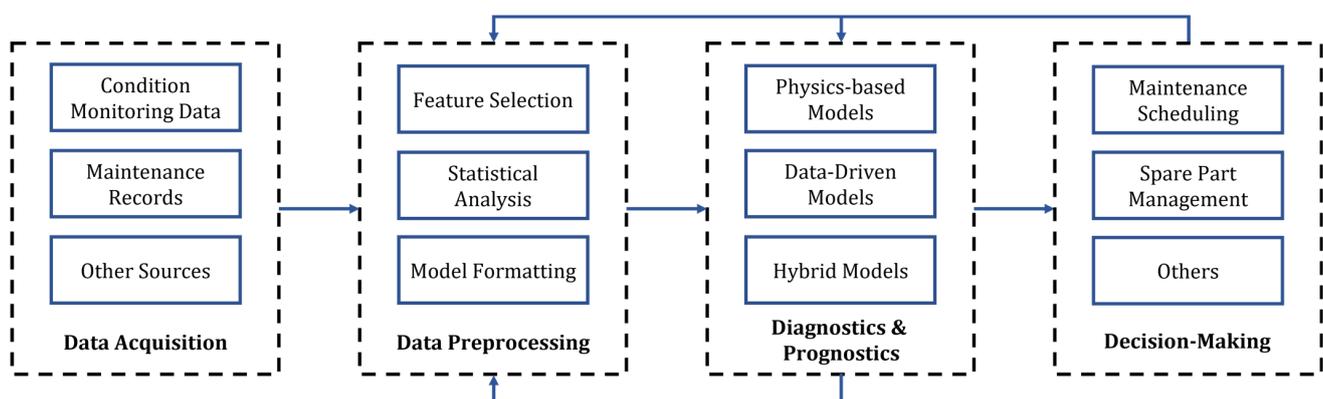


Figure 1. Example of the main stages in a PHM framework.

Multiple works have been published exploring different data-driven techniques employed for both diagnostics and prognostics in WTSs. Due to the lack of reliable labels and the difficulties presented when training prognostics algorithms, most of these DDM-PHM architectures in WTSs are used for anomaly detection and fault diagnostic tasks. These are mostly trained on data collected through the SCADA systems [12]. Among these models, SVMs, RF, and neural networks (NNs) are the most popular [45–47]. More complex DL architectures have also been implemented for anomaly detection [48]. For instance, Wu et al. [49] proposed a methodology to diagnose gearbox bearings and generator faults using SCADA sensor measurements based on a hybrid statistical–ML approach, combining LSTM and Kullback–Leibler divergence. LSTM models have also been used with AEs to develop an adaptive anomaly detection method, which then was employed with support vector regression as an adaptive threshold of performance index [24]. A hybrid approach was proposed in [50], where NNs were combined with statistical proportional hazard models for real-time performance and stress condition assessment.

Given the model-agnostic nature of ML and DL algorithms, their performance heavily relies on data availability and quality. Data preprocessing has been identified as a fundamental stage in PHM frameworks [37]. The importance of feature selection and outlier detection for diagnostic and prognostic models in WTSs have been studied thoroughly by Marti-Puig et al. [12,51]. The outlier detection process provides features with more representative domains, which in turn yields models with better generalization capabilities.

ML techniques such as SVMs tend to perform better for small input dimensions; thus, feature extraction and selection play an important role in generating smaller and representative datasets. Hence, for these models, a smaller dataset results in shorter model training and evaluation times and models with high performance. This is key to enable the online deployment of these models for WTSs. In this regard, PCA and AE have been implemented to train diagnostic and prognostic DDMs, as well as for feature reduction techniques [37,52]. Regarding WTS analysis, PCA has been implemented for different applications, including a data visualization tool [53], feature selection and reduction [54,55], and fault detection methods [56,57]. In this regard, utilizing and comparing AE and PCA as effective feature reduction tools have not been as widely studied in WTS settings.

3. Theoretical Background: Quantum Computing

This section discusses the required background for quantum computing and quantum machine learning. Section 3.1 introduces quantum computing and presents the concepts of qubit, quantum gates, and encoding schemas. Finally, Section 3.2 presents a brief revision on quantum machine learning and describes the quantum kernel circuit.

3.1. Quantum Computing

In the traditional paradigm of computation, the most basic unit of information is represented as a deterministic two-state artifact known as a bit. At a conceptual level, a bit is only able to represent, in a deterministic manner, one of two possible states: 0 or 1. At a hardware level, modern computers implement bits in microcircuits where the presence or absence of current determines the state of the bit. These bits can be utilized in logical operations to construct logical gates, such as the well-known AND and OR gates. These gates can also be combined to generate more complex artifacts, such as arithmetic circuits, memory components, and basically everything else that conforms to what is known today as a modern computer. In this regard, since the early 1950's the concept of a bit (both at a theoretical level and at a hardware level) has been used to develop and test the current understanding of modern computing.

Quantum computing is a new paradigm in which quantum mechanics phenomena are leveraged to perform computation. At its core, quantum computing proposes to replace the concept of bit with a more flexible quantum substitution called a quantum bit or qubit. While the traditional bit is limited to deterministically represent one of two possible states, the qubit is a two-state quantum system and therefore can be placed into superposition, encoding a probability distribution between the two possible states. Mathematically, the qubit is a vector in a 2D complex space and therefore can be represented as a complex linear combination of two basis vectors $|0\rangle = [1\ 0]^T$ and $|1\rangle = [0\ 1]^T$ in the form shown in Equation (1) [58]:

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle, \quad c_0, c_1 \in \mathbb{C}, \quad (1)$$

where the *ket* notation ($|\cdot\rangle$) is used to represent the basis vectors, following the nomenclature adopted by quantum mechanics. As c_0 and c_1 represent quantum state's probability amplitudes, then $|c_0|^2 + |c_1|^2 = 1$ must be satisfied. This normalization condition results from the fact that, when a physical quantum system is measured, it can only collapse into one of its possible states with a probability proportional to its amplitude. Multiple qubits can be operated together to form more complex quantum states. Consequently, multi-qubit systems can be represented mathematically using the outer tensor product, as shown in Equation (2) [58]:

$$\Psi = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_N\rangle, \quad (2)$$

where Ψ represents the quantum state formed by qubits denoted as $\{\psi_i\}_1^N$. The output of this operation is a quantum system of 2^N possible states, each with a complex probability

amplitude c_i . As in the case of individual qubits, the normalization condition still holds for multi-qubit systems as well, indicated by Equation (3) [58]:

$$\sum_{i=0}^N |c_i|^2 = 1. \quad (3)$$

While traditional systems composed of N bits can still represent a total of 2^N possible states, due to the deterministic nature of bits, only one of those states can be expressed by the system at a given time. In quantum computing, and in particular for an N qubits system, a superposition of all those states can be represented simultaneously. This alternative form of state representation is what fundamentally motivates the interest in quantum computing and its potential applications in lowering the algorithmic complexity of certain tasks.

As in the case of traditional bit systems, qubits systems can also be operated using quantum gates, which are represented by unitary matrices and represent the fundamental ways in which the qubits' states can be modified to perform a certain computational task. The gates applied to a multi-qubit system and the order in which they are applied is commonly known as a quantum circuit. In what follows, the relevant quantum gates are described [58].

3.1.1. Hadamard Gate

The Hadamard gate is a single qubit gate used to induce superposition into a system. Mathematically, the matrix form for the Hadamard gate is depicted in Equation (4):

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (4)$$

When H is applied over a qubit in the basal state $|0\rangle$, the resulting system has equal probability of collapsing to either state (i.e., $|0\rangle$ or $|1\rangle$) when measured, as it is demonstrated in Equation (5) by left multiplying the gate and the basal state qubit:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} [1 \ 0]^T = \frac{1}{\sqrt{2}} [1 \ 1]^T = \frac{1}{\sqrt{2}} [1 \ 0]^T + \frac{1}{\sqrt{2}} [0 \ 1]^T, \quad (5)$$

where $c_0 = c_1 = \frac{1}{\sqrt{2}}$; therefore, the condition $|c_0|^2 + |c_1|^2 = 1$ is fulfilled.

3.1.2. Controlled Not Gate

Entanglement is another important quantum mechanics property that is leveraged in quantum computing for constructing dependencies between qubits. The controlled not gate (C-NOT gate) induces entanglement into a two-qubit system, which is a two-qubit gate defined as shown in Equation (6):

$$C \ NOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (6)$$

When applied over a two-qubit system, the C-NOT gate generates the following control scheme: assuming that both qubits are in absolute basal states (i.e., either $|0\rangle$ or $|1\rangle$), the first qubit acts as the control qubit, while the second is the one under control; if the first qubit is $|0\rangle$, then the system is not affected by the C-NOT gate; if the first qubit is $|1\rangle$, then the second qubit is inverted to the opposite state. More generally, if the qubits are not in their basal states, the C-NOT gate inverts the probability of the third and fourth states of the combined system.

3.1.3. Rotation Gates

The Hadamard and C-NOT gates can be classified as non-parametric gates, since they operate directly on qubits without the need for the specification of external parameters. On the other hand, rotation gates are single qubit parametric gates, as their effect on a qubit can be fine-tuned externally. To visualize their effect, it is necessary to introduce the spherical representation of qubits. As shown before, qubits are vectors in a 2D complex space. If both complex coefficients are represented in their polar form, the expression depicted in Equation (7) is obtained:

$$|\psi\rangle = r_0 e^{i\phi_0} |0\rangle + r_1 e^{i\phi_1} |1\rangle. \quad (7)$$

Equation (7) shows that four parameters are needed to represent a qubit: two amplitudes (r_0 and r_1) and two phases (ϕ_0 and ϕ_1) to form the original complex probability amplitudes. Nevertheless, using the normalization condition and the fact that a qubit will not physically change when amplified by a complex factor of unitary norm (regardless of the phase, so it is admissible to apply a complex factor of phase $-\phi_0$ without loss of generality) [58], it is possible to reduce the required number of coefficients to represent the qubit to two, as shown in Equation (8):

$$|\psi\rangle = e^{-i\phi_0} (r_0 e^{i\phi_0} |0\rangle + r_1 e^{i\phi_1} |1\rangle) = r_0 |0\rangle + r_1 e^{i(\phi_1 - \phi_0)} |1\rangle = \cos \theta |0\rangle + \sin \theta e^{i\varphi} |1\rangle, \quad (8)$$

where the normalization condition is used to reduce the amplitudes r_0 and r_1 to a single parameter θ by defining $r_0 = \cos \theta$ and $r_1 = \sin \theta$. Additionally, only one phase term survives after multiplying by the unitary complex factor with phase $-\phi_0$; therefore, $\phi_1 - \phi_0$ can be replaced by a second independent parameter φ . Hence, Equation (8) shows that a qubit can be represented by two parameters, θ and φ . These parameters can be interpreted as angles in a unitary sphere, also called a Bloch Sphere. Consequently, a qubit can be understood as a point on the surface of such a sphere. This qubit representation is depicted in Figure 2.

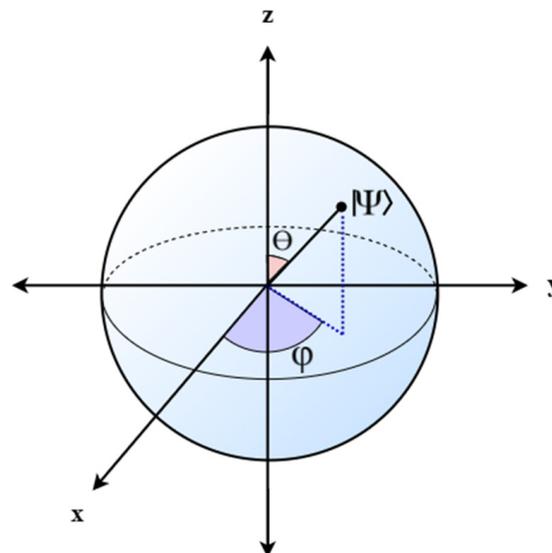


Figure 2. A qubit $|\psi\rangle$ represented as a point in the unitary Bloch Sphere. Note from Equation (8) that, for $\theta = 0$ or $\theta = \pi$, the qubit is located at the north and south poles, which coincides with the $|0\rangle$ or $|1\rangle$ states, respectively. Another important case is $\theta = \frac{\pi}{2}$, where the qubit is located exactly in the equatorial surface of the sphere, representing a perfect superposition of both basal states. (Adapted from [59]).

The Bloch Sphere representation allows a straightforward interpretation of the effect of the rotational gate operation. Each operation rotates the qubit about a main axis by a

certain number of radians specified by the external parameter ζ . The matrices for these gates are presented in Equations (9)–(11):

$$R_x(\zeta) = \begin{bmatrix} \cos \frac{\zeta}{2} & -i \sin \frac{\zeta}{2} \\ -i \sin \frac{\zeta}{2} & \cos \frac{\zeta}{2} \end{bmatrix}, \quad (9)$$

$$R_y(\zeta) = \begin{bmatrix} \cos \frac{\zeta}{2} & -\sin \frac{\zeta}{2} \\ \sin \frac{\zeta}{2} & \cos \frac{\zeta}{2} \end{bmatrix}, \quad (10)$$

$$R_z(\zeta) = \begin{bmatrix} e^{-i\frac{\zeta}{2}} & 0 \\ 0 & e^{i\frac{\zeta}{2}} \end{bmatrix}. \quad (11)$$

3.1.4. Encoding Schemas

One of the first challenges to overcome when applying quantum computing techniques to real valued data is the encoding process. This refers to how real value data should be encoded into a multi-qubit system to perform computations. While at first glance this may not seem different from any other type of encoding, in the quantum computing setting an important limitation is the hardware feasibility. Every encoding schema needs to be conducted on real quantum hardware; therefore, the encoding operations need to adhere to the restrictions imposed by quantum mechanics. For this reason, encoding schemas are an active area of research that lies in the line between quantum software and hardware. In the following section, two of the most common encoding schemas are presented.

Angular Encoding

In angular encoding, a parametric circuit is applied prior to the circuit that will manipulate the data to produce the desired output. This parametric circuit uses one rotational gate per qubit to encode real numbers in the phase angle of every qubit. By performing this operation, angular encoding needs N qubits to represent an N -dimensional real valued vector. Mathematically, the operation is depicted in Equation (12) [60]:

$$\vec{x} \rightarrow |\Psi\rangle = R_x(x_0) \otimes R_x(x_1) \dots R_x(x_{N-1}) \otimes R_x(x_N), \quad (12)$$

where R_x is the rotation gate with respect to the X axis in the Bloch Sphere (as depicted by Equation (9)), applied independently to every qubit, accepting the real values $\{x_i\}_1^N$ as its parameters, corresponding to each original dimension. While angular encoding is not as efficient in terms of encoding capacity as other encoding schemes, it is one of the simpler ones to configure and therefore is one of the most used.

Amplitude Encoding

In this type of encoding, the classical information is encoded into the amplitudes of each of the possible states, requiring $\log_2(N)$ qubits to encode an N -dimensional real value vector, which makes it more efficient than angle encoding. The mathematical operation for angular encoding is presented in Equation (13) [60]:

$$\vec{x} \rightarrow |\Psi\rangle = \sum_{i=0}^{2^N} x_i |i\rangle, \quad (13)$$

where $|i\rangle$ represent the possible basal states for an N -qubit system. It is important to note that the real values x_i need to be normalized prior to the encoding application to ensure that the resulting quantum state is valid.

3.2. Quantum Machine Learning

Quantum machine learning is a new field of research that lies in the juxtaposition between traditional ML and quantum computing. The general objective is to leverage the

theoretical advantages of quantum computing to either create new data-driven algorithms or to enhance existing ones. Two approaches have been treated with interest by the research community. The first one is a parametric approach, in which a quantum circuit composed by parametric gates is treated as a trainable model, updating the parameters to minimize a given objective function. The advantage of this approach is that it allows researchers to draw clear similarities and parallels with traditional NNs, as both models perform prediction tasks based on observational data undergoing an iterative learning phase. Parameterized quantum circuits (PQC) [61] have already been used in the PHM context to classify health states in rotatory machinery with similar results as traditional approaches [62]. Nevertheless, while useful and simpler to understand, PQC methods fall short in their flexibility, as they can only be used for classification or regression tasks. On the other hand, quantum kernel approaches can also be used for prediction tasks; additionally, they can be extended to other tasks such as clustering or dimensionality reduction.

Quantum Kernels

In this section, a particular quantum circuit used to generate a quantum kernel-like function will be introduced. Quantum kernels [63] are specialized quantum circuits that effectively perform the same operation as traditional kernels. That is, they perform an internal dot product between the image of two vectors as shown in Equation (14):

$$\kappa(\phi(x_i), \phi(x_j)) = |\langle \phi(x_i) | \phi(x_j) \rangle|^2, \quad (14)$$

where x_i and x_j are two N -dimensional vectors and $\phi(x) : R^N \rightarrow R^M$ is a feature map that transforms the vectors from R^N space to R^M space. In the quantum context, a kernel is a circuit that performs a similar operation, with the notable difference that $\phi(x) : R^N \rightarrow C^M$ is now a feature map that encodes the data into a quantum state; therefore, the dot product is performed according to the complex space rules. In terms of implementation, the circuit is composed of two parametric encoding blocks that can receive as inputs the input classical data. For example, these encoding blocks could be angular or amplitude encoding circuits. These two blocks are applied successively to the same multi-qubits system prepared in a basal state. Then, a measurement operation is applied. The final state of the circuit can be computed as shown in Equation (15) [63]:

$$\begin{aligned} \langle 0..0 | S(x') S(x)^\dagger M S(x')^\dagger S(x) | 0..0 \rangle \\ = \langle 0..0 | S(x') S(x)^\dagger | 0..0 \rangle \langle 0..0 | S(x')^\dagger S(x) | 0..0 \rangle, \end{aligned} \quad (15)$$

where $S(\cdot)$ is an encoding operation applied over the real data, and M is a measurement operation. The right-hand side of Equation (15) represents a squared norm, and it can be rewritten as depicted in Equation (16):

$$\langle 0..0 | S(x') S(x)^\dagger M S(x')^\dagger S(x) | 0..0 \rangle = |\langle 0..0 | S(x') S(x) | 0..0 \rangle|^2, \quad (16)$$

where the feature map function is identified as $\phi(x) = S(x) | 0..0 \rangle$; therefore the expression shown in Equation (17) can be identified as a kernel function between two datapoints x and x' :

$$\langle 0..0 | S(x') S(x)^\dagger M S(x')^\dagger S(x) | 0..0 \rangle = |\langle \phi(x') | \phi(x) \rangle|^2 = \kappa(x, x'). \quad (17)$$

Figure 3 portrays a diagram of this quantum circuit.

The quantum kernel portrayed in Figure 3 can, in principle, be used to replace any traditional kernel. This requires the close interaction of a quantum computer and a traditional computer, making this algorithm a hybrid approach. The quantum kernel for the entire dataset would be computed in the quantum computer and then utilize those results in a traditional algorithm, usually executed in a classical computer. The fact that quantum kernels can be seen as replacements to traditional kernels, maintaining the same basic properties, gives this approach immense flexibility in its range of applications. In this

paper, the attention is centered on the application of quantum kernels to SVM classification algorithms, which is presented in Section 5.

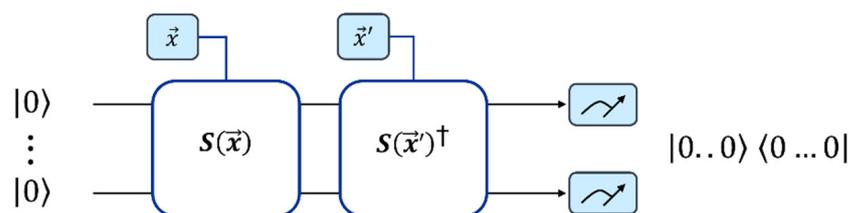


Figure 3. Kernel circuit composed of two encoding blocks applied over the necessary qubits in the basal state. The second encoding block is defined as the complex conjugate of the first one.

4. Wind Turbine Data Analysis

As in multiple complex engineering systems, one of the main challenges when analyzing data collected through SCADA is related to the lack of formal data preprocessing methodologies. Improper data manipulation can cause a significant performance drop in the diagnostics and prognostics models. Further, as SCADA data entries are recorded every 10min, this low temporal resolution is poorly complemented by the SCADA rule-based alarm systems. In this data collection regime, multiple alarms communicating the deviation of nominal operational conditions may be logged automatically in the same 10min window. Additionally, after an alarm has been triggered and the WT controller has initiated the corresponding corrective action (e.g., yaw control and grid disengagement), the same alarm may be triggered again once the alarm reset time has been surpassed, independently of whether the event or condition which caused it has not been corrected or avoided by the control system. Additionally, it has been noted that individual alarms do not necessarily indicate that a fault has occurred [18]. These issues lead to noisy, overlapping, and intractable alarm logs. Hence, data-driven implementations for diagnostics and prognostics have focused on directly analyzing the SCADA data entries or localized sensors with higher acquisition frequency rather than extracting useful information from the alarm log system. For this case study, both data sources are considered to develop diagnostic models.

4.1. Case Study

The WTS data collected correspond to a period between 2015–2019 from an onshore wind farm. In this paper, the analysis focuses on a single turbine, utilizing both the available SCADA sensor measurements data and recorded alarms. Considering a sampling time of 10 min, this period amounts to 251,164 temporal entries of sensor measurements. In this period, 337,448 alarms were registered in the analyzed turbine. The details of this dataset are discussed below.

The recorded SCADA data consist of various sensor measurements and event logs recorded with a sampling time of 10 min. Each sensor variable is described by its mean, maximum, minimum, and variance of its values in the 10min time window. For convenience, the recorded variables can be categorized in electrical, mechanical, temperature, and environmental types, as shown in Table 1. Monitored components include turbine blades, rotor, nacelle, gearbox, bearings, and cooling system, as well as multiple controllers and indicators of the generator, grid, and WTS states [7,49].

Table 1. Internal categorization of WTS sensors.

Variable Type	Sensor Measurements
Component status	Engagements, system checks
Electrical	Power, current, voltage, frequency
Mechanical	Blade position, RPM, yaw brake
Temperature	Mechanical/electrical components
Environmental	Pressure, humidity, wind speed, wind direction

Internal SCADA alarm codes are triggered under a variety of circumstances, including operational and communication logs, detected faults, start-ups, and cool-downs. Of a total of 369 alarm codes, these can be categorized internally as: mechanical issues and temperature anomalies (93), electrical anomalies (82), control actions (82), sensor malfunctions (77), operational signals (20), test codes (11), and environmental conditions (4). In the period 2015–2019, the number of alarm logs categorized by their severity level is shown in Table 2.

Table 2. Breakdown of SCADA alarm logs per severity.

Event Severity	Number of Alarm Logs
Alarm	242,401
Miscellaneous	51,208
Warning	43,839
Total	337,448

The breakdown of the most relevant alarm logs, excluding warnings and miscellaneous codes, is shown in Figure 4. As it can be observed, a high number of alarm logs corresponds to pitch faults (identified as pitch fault 1 and 2), followed by operational control actions (e.g., powering up the central controller, receiving a remote command to stop, or to stop based on safety concerns) and external faults related to grid stability, sensor faults, or control system communication faults. The pitch fault alarms are triggered when the angle between the blades surpasses a certain threshold. If this threshold is infringed for more than 60 s, an alarm is triggered. This activates the corresponding sections of the WT central control system and thus initiates corrective actions. For the studied pitch faults, these alarms result in an automatic shutdown. If the monitoring sensors detect that after the alarm reset time the conditions are still anomalous, the SCADA alarm system registers a new log. Given the number of WTS stops induced by the pitch fault alarms, it is of interest to develop DDMs to detect them. Further details on the importance of the pitch fault alarm and its relatively high failure rate when compared to other failure modes can be found in [4,8,57,64–66].

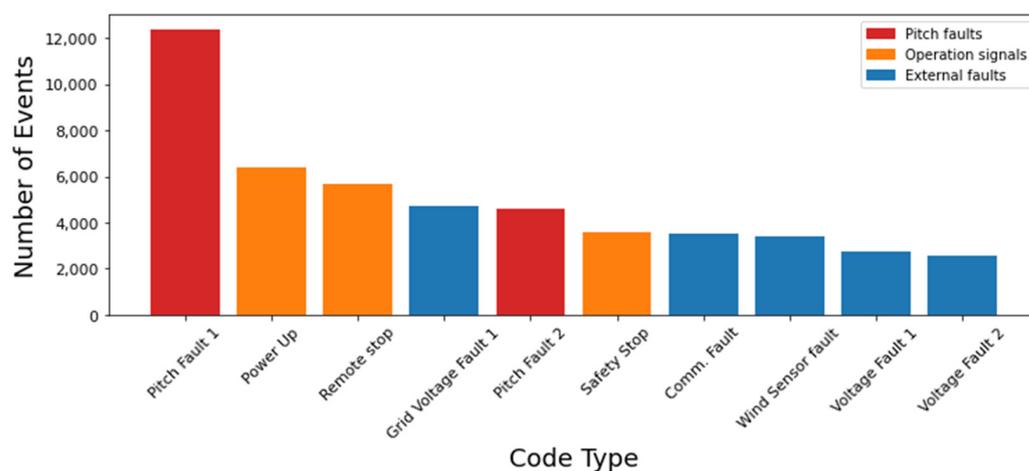


Figure 4. Breakdown of top alarm events recorded in the WTS.

4.2. Data Preprocessing

The data preprocessing stage is focused on mapping the alarm logs to the sensor data and obtaining a representative dataset of the WTS's pitch faults. The analysis is performed via the classification of the WTS health state, since obtaining robust and reliable RUL labels from the SCADA system for regression approaches is currently a difficult task. Hence, the data are preprocessed for the diagnosis (i.e., classification) of an unbalanced pitch faults failure mode. The preprocessing stage also includes feature reduction analysis through PCA and AE. This stage yields the selected principal components and latent space

representation of the sensor data that will be used as the input features to train and test the diagnostic models.

4.2.1. Alarm Logs and Label Generation

Regarding the WTS's SCADA alarm logs, the analysis focuses on the detection of unbalanced pitch faults. This alarm is triggered when the angle between two blades of the wind turbine differs more than the specified setpoint. These faults are recorded under three different alarm codes depending on the operational stage of the WTS (i.e., normal operation, shutdown, stationary). Each alarm log entry delivers the alarm code, the time at which the anomaly was detected, and the time at which the alarm is reset (i.e., fixed time). For instance, Figure 5 shows the distribution of alarm duration for different alarm codes. Here, it is shown that the majority of the continuously occurring pitch fault alarms have a duration of 1 h. Hence, a naive approach is selected to simplify the pitch fault detection task, such that faults are detected in the WTS 1 h prior to their occurrence. Consequently, the SCADA sensor data are averaged in time windows of an hour. Each sensor data entry is labeled as "healthy" or "faulty" according to whether any pitch fault alarm was triggered at any point during the previous hour.

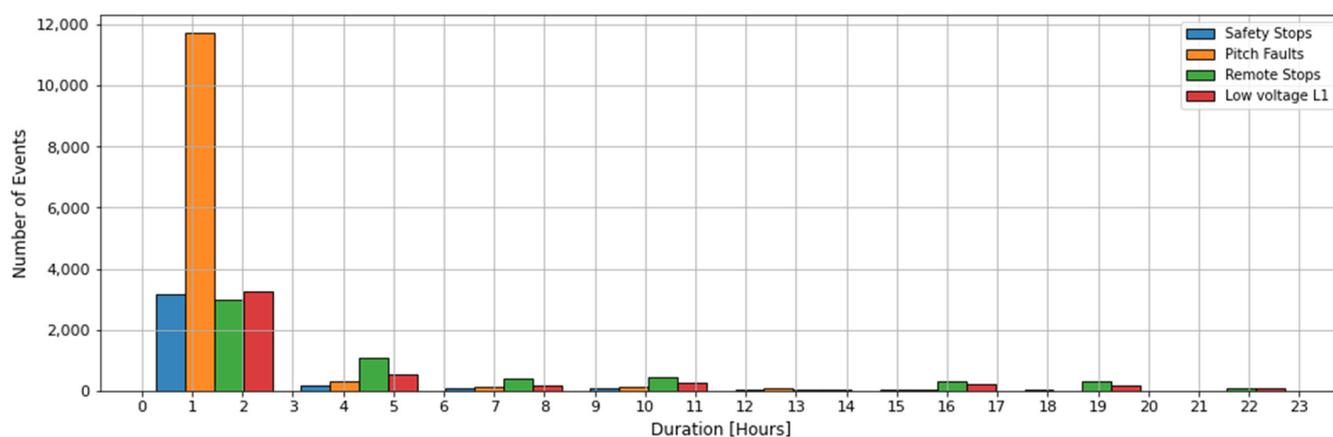


Figure 5. Histogram of alarm durations (detection–reset) for selected faults.

4.2.2. Feature Reduction Analysis

An important step in the data preprocessing stage for ML diagnostic and prognostic models is feature selection and reduction. A WTS is instrumented with hundreds of different sensors, monitoring a wide variety of variables. Although all these sensors could potentially contain valuable information on the system's state of health, ML models are known to struggle when trained on large input dimensionalities. Therefore, manual variable selection and statistical dimensionality reduction methods are commonly used to create diagnostic datasets from multi-sensor systems. These methods are also useful for analyzing systems with multiple sensors to identify representative features extracted from the original sensor data and thus disregarding uninformative sensor variables. Furthermore, unrelated variables can be identified and discarded based on expert knowledge [37].

The original SCADA dataset for this case study consists of 385 sensors. The information quality provided by these variables is measured based on the percentage of useful data they can provide and whether they provide numerical values or not. For instance, sensors with missing information are discarded based on the number of NaN (not-a-number) values they present, where columns and rows with more than 5% void entries are excluded. Figure 6a shows the original distribution of void entries, where almost 100 columns contain no information, while over 200,000 rows contain at least 36% of void entries. Figure 6b shows the resulting distribution after filtering under 5% void entries per column and rows, respectively [37]. Further, non-informative variables reporting event counts are excluded from the analysis, as well as variables that only indicate the current state of a sensor

(e.g., connected, online, and failed communication). This results in a selection of 168 of variables related to physical sensors in the system, covering temperature, vibration, and electrical measurements. This reduces the dataset size from around 96.7 M to 42.2 M of useful data entries. Table 3 compares the resulting dataset sizes when a 5% and a 0% void entry threshold is used on the original dataset.

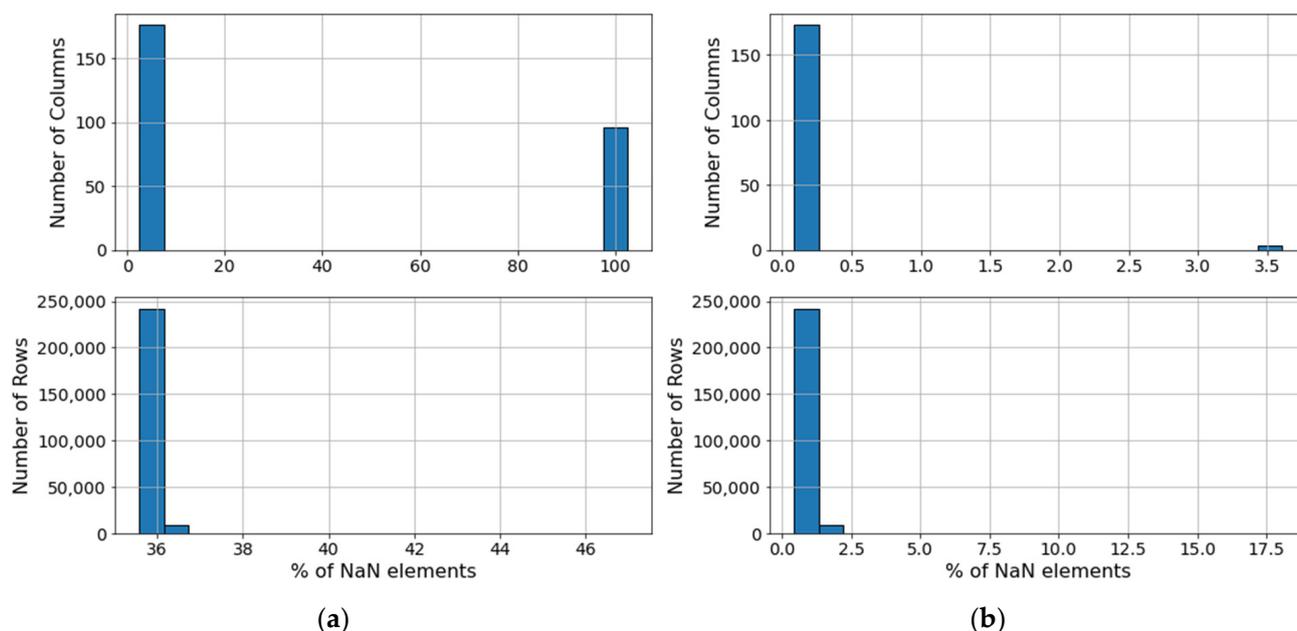


Figure 6. Distribution of NaN entries in recorded sensor data: (a) original distribution; (b) processed distribution.

Table 3. Dataset reduction by void entry threshold.

	Void Entries	Number of Features	Dataset Size
Original	24,141,390	385	96,698,140
Processed (5% NaN threshold)	29,646	168	42,195,552
Processed (0% NaN threshold)	0	142	35,665,228

Further dimensionality reduction can be achieved mainly through two methods. On the one hand, statistical tests can be performed to assess the correlation between the input features and the output labels (e.g., ANOVA test). To this end, Python libraries such as scikit-learn have implemented packages that automatically select a specified number of features based on scores obtained from a determined statistical test [67]. One example is the SelectKBest, which has proven to be an effective tool for diagnostic models [68]. However, using these tools requires manually selecting an appropriate test as well as a score threshold to determine what is considered to be an informative feature. Furthermore, the obtained scores are individual for each feature and do not have a cumulative information metric from which to construct this threshold. On the other hand, feature reduction techniques that map the original input data to a reduced dimensionality space, such as PCA and AE, allow one to make an informed decision for all features simultaneously. Although these techniques do not retain the original features of the data, they reduce the dimensionality without losing as much information as by manually discarding features from the original data. Additionally, both PCA and AE have been shown to exhibit denoising properties, which can be beneficial when training diagnostic and prognostics models [23].

In this work, both AE and PCA are employed to obtain nonlinear and linear representations of the original dataset in a lower dimensionality space. On the one hand, the cumulative explained variance (CEV) is used to evaluate the PCA's performance and

determine the adequate number of principal components that make the reduced dimensionality representative [69]. The CEV is shown in Figure 7 for the first 32 principal components extracted from the sensor dataset. It is expected that, at a higher CEV, the dataset is more representative while reducing the number of correlated variables. Although there is no general rule of thumb to what minimum CEV is required to faithfully represent a dataset, a 90% CEV is considered an acceptable threshold to select the number of principal components. For instance, in [50], PCA was used as a feature reduction tool to obtain a smaller and representative dataset, which accounted for 90% of the CEV.

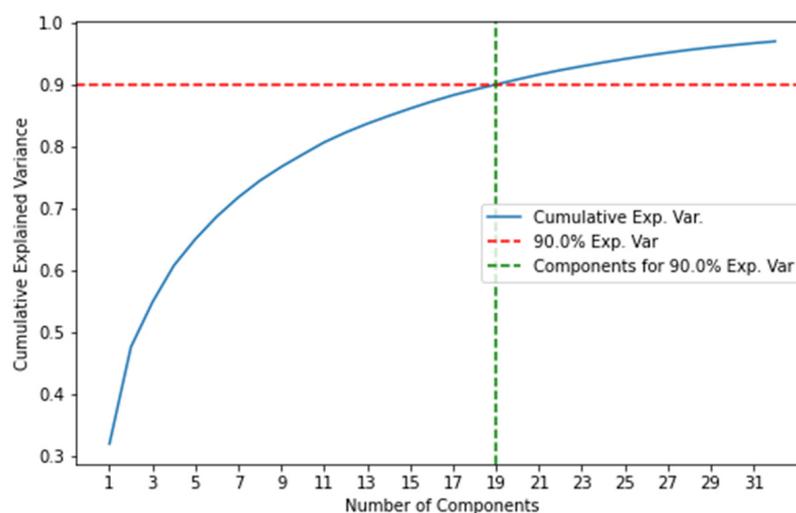


Figure 7. Cumulative explained variance by number of principal components employed.

Table 4 shows that 19 principal components correspond to a 90% CEV for this case study, which is also illustrated in Figure 7.

Table 4. Cumulative explained variance per number of principal components.

Number of PCs	CEV	Number of PCs	CEV
1	0.321	17	0.884
2	0.477	18	0.893
3	0.549	19	0.901
4	0.609	20	0.909
5	0.651	21	0.917
6	0.688	22	0.924
7	0.719	23	0.930
8	0.746	24	0.937
9	0.768	25	0.942
10	0.788	26	0.947
11	0.808	27	0.952
12	0.824	28	0.957
13	0.838	29	0.961
14	0.850	30	0.964
15	0.862	31	0.968
16	0.873	32	0.971

On the other hand, AEs are deep NNs that are trained to replicate their input values. That is, the network input corresponds to a vector X , and its output corresponds to the estimation of that same input \hat{X} . The network consists of two phases: an encoder and a decoder. For dimensionality reduction purposes, the encoder maps the input value into a smaller latent space, and then the decoder reconstructs the latent space into the original dimension. Both the encoder and the decoder consist of NNs with nonlinear activation functions. In theory, an AE with linear activation functions should be equivalent to a PCA.

However, an NN-based AE with nonlinear activation functions is expected to obtain a smaller and more accurate latent space representation of the data when compared to a PCA. It should be noted that these feature reduction tools do not allow one to trace what specific features have been selected to represent the data at a lower dimensionality.

To select the most representative latent space dimensionality, a sensibility analysis on the AE's reconstruction error must be performed. Figure 8 presents the AE reconstruction mean squared error (MSE) for different latent space dimensions. The vertical green line is used as a reference corresponding to the 19 principal components, and it can be observed that the reconstruction MSE starts to converge around 15–16 features latent space dimensionality. Note that each training of the AE will yield different results, even if the architecture and data remain untouched. As such, Figure 8 would present a smooth behavior (as the PCA in Figure 7) if multiple models were trained for each architecture (i.e., latent space dimensionality).

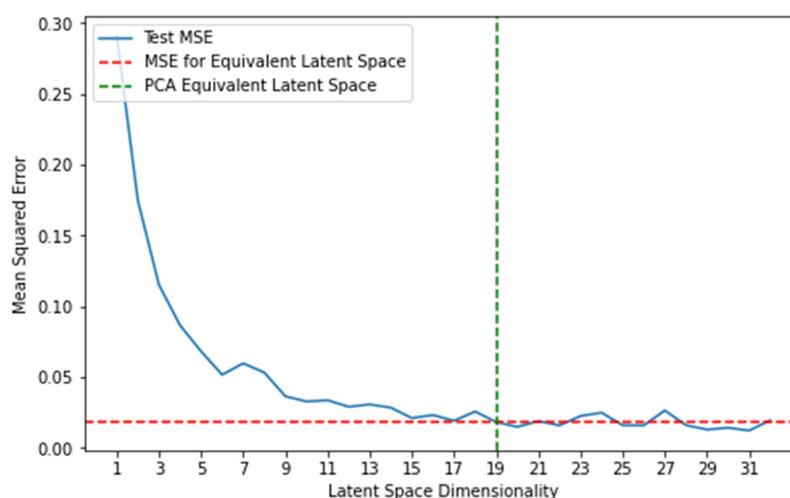


Figure 8. Reconstruction MSE for different latent space dimensionalities.

Given the results presented in Figures 7 and 8, it is expected that both dimensionality reduction techniques will present similar performance when used to train diagnostic models. It should be noted that, although a better representation is expected from the AEs, they tend to be more computationally demanding than the PCA due to the deep NN architecture. Indeed, the training processes for the AE take an average of 60 s, while training the PCA takes an average of 1 s on the same hardware. Furthermore, unlike the CEV metric, a low reconstruction MSE value does not provide an interpretable metric on the latent space's contained information. Other AEs' drawbacks include inflexibility to manage void entries (i.e., NaN values), limiting the available data that can be used to train the model, and the new unseen data that can be evaluated once the model is online. In this case, the void entry reduction technique is applied with a 0% threshold, reducing the useful dataset to 142 features and 35.6 M entries (Table 3), potentially affecting the system's representability of the obtained data. This is of great importance in WTSs, since void entries are commonly encountered in real datasets. Hence, both PCA and AE techniques for feature reduction should be explored and compared simultaneously when used to process the input data to train diagnostic models.

5. Quantum-Based Wind Turbines' Pitch Fault Prognostics

This section describes the computational experimental setup and results for the quantum-enabled and classical diagnostic approaches. The data are split into healthy and degraded state classes. Based on the previously discussed PCA and AE feature reduction process, the ML model training and testing process are described. The results of the classification models and the effect of the feature reduction techniques employed are also discussed.

5.1. Computational Experimental Setup

In this work, the performance of all the ML models is compared based on datasets of reduced dimensionalities. This dimensionality reduction is performed with both the PCA and AE techniques, as described in Section 4.2. A sensibility analysis to assess the impact of the feature reduction on the models' performance is presented. The tested dimensionalities are 4, 8, 16, 19, and 32 features. These dimensions are chosen based on two criteria. First, the dimensions 4, 8, 16, and 32 are chosen based on the encoding techniques available for quantum algorithms. Secondly, experiments with 19 features are included to represent the threshold of the PCA's 90% explained variance and where the AE's reconstruction MSE starts to converge. The resulting dataset sizes are reported in Table 5.

Table 5. Dataset size by number of features used in the reduction process.

Number of Features	Dataset Size
4	1,004,656
8	2,009,312
16	4,018,624
19	4,772,116
32	8,037,248

With respect to the quantum classification approach, both angular and amplitude encoding were utilized as the feature map function for the quantum kernel for the datasets including four and eight principal components. For the cases where 16 and 32 principal components were utilized, only amplitude encoding was tested due to limitations in quantum circuit simulation. Given the exponential increase of possible states that a quantum model can represent as the number of qubits increases, simulating a system with over 12 qubits is generally not possible on modern classical computers running quantum simulators. For the special dataset including 19 principal components, zero-padding was used to augment the dimension of each datapoint to 32 features, which is the closest number to a power of two (i.e., 2^5). As this operation is performed on each datapoint after the initial preprocessing and division into training and testing sets of the data, it does not affect the balance or fairness of the experiments nor leak training data into the testing set.

The reduced datasets are separated into balanced training and test sets [15,70]. That is, the training datasets present the same number of entries labeled as "healthy" and "faulty" states, with the purpose of reducing the model's bias toward the most observed state (i.e., the healthy state). As shown in Table 6, out of the original 251,164 temporal entries, only 779 of these correspond to faulty states. Hence, to create the balanced training and test sets, 779 entries labeled as healthy are randomly selected. The resulting 1558 entries are then divided into training and test sets, considering a 20% split to test the models' performance after these have been trained.

Table 6. Resulting size of balanced classes for healthy and faulty states.

Healthy	Degraded	Balanced Training Set Size	Balanced Test Set Size
250,384	779	1244	308

The quantum kernels are compared to traditional ML techniques, namely: SVM with both linear and RBF kernels, RF, and k-NN. The models' performances are compared based on the same dataset. A stratified k-fold (10) is used for the hyperparameter selection. The ML models are trained on Python 3.8 and the Pycaret library. Ten different models are independently trained for each classical algorithm, whereas five different models are independently trained for each quantum-based algorithm. Reported classification metrics include the average and standard deviation of accuracy, precision, recall, and F1 score. The utilized hardware consists of an NVIDIA RTX 3060 GPU, an 8-core AMD Ryzen 7 5800X CPU, and 32 GB of RAM memory.

5.2. Results and Discussion

This section is structured as follows. First, results regarding the classification task using a feature reduction strategy based on PCA and AE are presented in Table 7. This table reports the models' average accuracy and F1 score achieved for the training and test sets of five different reduced dimensionalities. Additional metrics, including averages of precision and recall, as well as the standard deviation for all metrics, are presented in Appendix A (Tables A1–A5). Then, the performance of the PCA and AE as feature extractors is discussed, followed by a comparison of the classical ML models. The performance of the Q-SVM model is discussed for the two types of encoding presented. Finally, the performance between classical ML and Q-SVM is discussed.

Table 7. Diagnostic model metrics using data preprocessed through PCA and AE.

Model	Dimension	PCA				AE			
		Train		Test		Train		Test	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1
RF	4	0.868	0.870	0.871	0.877	0.907	0.910	0.877	0.879
	8	0.898	0.905	0.887	0.889	0.897	0.902	0.885	0.887
	16	0.910	0.914	0.909	0.911	0.895	0.896	0.899	0.901
	19	0.926	0.928	0.916	0.919	0.920	0.921	0.902	0.905
	32	0.923	0.925	0.906	0.908	0.906	0.912	0.894	0.898
k-NN	4	0.871	0.876	0.869	0.875	0.915	0.920	0.888	0.891
	8	0.901	0.909	0.882	0.884	0.921	0.922	0.904	0.906
	16	0.914	0.916	0.904	0.906	0.912	0.912	0.912	0.913
	19	0.924	0.928	0.910	0.912	0.924	0.925	0.905	0.908
	32	0.920	0.923	0.906	0.909	0.915	0.921	0.904	0.907
SVM-Linear	4	0.824	0.832	0.819	0.827	0.771	0.765	0.747	0.742
	8	0.878	0.884	0.854	0.862	0.907	0.911	0.894	0.898
	16	0.912	0.915	0.890	0.894	0.903	0.907	0.883	0.888
	19	0.897	0.900	0.898	0.900	0.925	0.929	0.919	0.923
	32	0.914	0.916	0.902	0.903	0.923	0.927	0.927	0.931
SVM-RBF	4	0.858	0.864	0.846	0.852	0.887	0.891	0.865	0.870
	8	0.897	0.902	0.887	0.892	0.931	0.933	0.917	0.920
	16	0.925	0.928	0.920	0.923	0.927	0.930	0.909	0.917
	19	0.932	0.934	0.897	0.900	0.954	0.955	0.945	0.947
	32	0.938	0.939	0.921	0.923	0.939	0.942	0.912	0.919
Q-SVM	4 [Angle Enc.]	0.854	0.859	0.811	0.817	0.846	0.851	0.827	0.832
	4 [Amplitude Enc.]	0.824	0.830	0.819	0.824	0.810	0.812	0.784	0.779
	8 [Angle Enc.]	0.890	0.895	0.888	0.893	0.930	0.933	0.894	0.898
	8 [Amplitude Enc.]	0.866	0.872	0.856	0.862	0.868	0.876	0.860	0.867
	16 [Amplitude Enc.]	0.896	0.899	0.897	0.900	0.875	0.887	0.874	0.885
	19 [Amplitude Enc.]	0.895	0.898	0.898	0.901	0.931	0.935	0.924	0.928
32 [Amplitude Enc.]	0.910	0.912	0.886	0.889	0.911	0.918	0.910	0.915	

In Section 4, a sensibility analysis regarding the optimal reduced feature dimensionality of the data was presented based on the CEV and MSE metrics. However, when implementing data-driven prognostics models, it is also desired to obtain the lowest possible dimensionality that does not hinder the model's performance to minimize the computational burden. In practice, this allows for simpler model selection and maintenance for effective online deployment given their on-site hardware requirements. As such, a sensibility analysis on the model's performance is required to assess the optimal feature space dimensionality, which might not necessarily coincide with the number of features indicated by the CEV and MSE thresholds. In general, Table 7 suggests that the prognostics models perform better when analyzing the data preprocessed with AE than with PCA. Additionally, model accuracies tend to increase with a higher dimensionality space, converging to values

above 0.90. F1 scores follow a similar trend as accuracy, indicating an adequate balance between false negatives and false positives. It should be noted that most ML models achieve a peak test accuracy over 0.90 for a reduced dimensionality of 19 features. This is the case for most models trained on the AE's latent space, although this behavior is only exhibited by the RF and k-NN models when trained on the PCA data. While it may be intuitive that more training features should increase the performance of the tested models, from an information point of view, this is not always the case. In this study, 19 features represent a threshold for which most of the information from the original dataset is encoded into the PCA or AE features, as shown in Figures 7 and 8. Adding extra information in the form of additional features may become detrimental to the learning process, as they are likely to add more noise than useful information; nonetheless, the algorithms are forced to interpret them. Moreover, it is known that ML models, such as SVM, are often affected by what is known as the “the curse of dimensionality”, where datasets containing a large number of features are not suitable for efficient training and thus require further feature extraction procedures to increase their prediction performance. In general, the performance of all trained models suffers the most when a lower number of features are used as input data, as expected by the loss of information for both the PCA and AE preprocessing approaches at lower dimensionalities. This behavior can be observed in Figure 9, where the test accuracies for the models are compared based on the corresponding number of features used for the data reduction process. This is consistent with the dimensionality reduction analysis performed for both the PCA and AE (see Figures 7 and 8). However, it should be noted that unless the AE's performance is compared to that of an interpretable metric, such as the CEV obtained through PCA, there is no guarantee that a representative dataset is efficiently obtained. Results indicate that the latent space representation obtained from the PCA and AE are not directly comparable, which confirms the value of simultaneously analyzing these two feature reduction techniques.

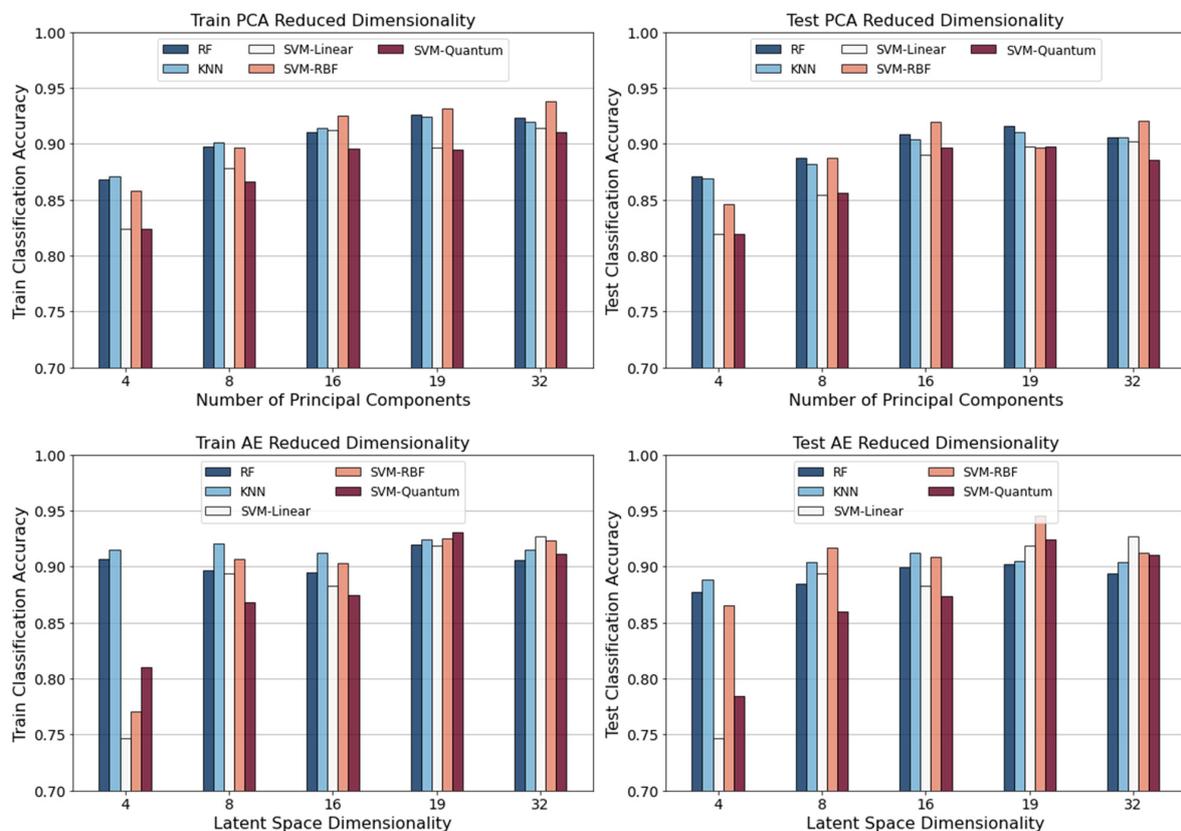


Figure 9. Comparison of models' train (left) and test (right) accuracy for reduced dimensionality using PCA (above) and AE (below).

The presented results indicate that the overall highest test accuracy is achieved by the SVM-RBF model at 0.945 and 0.921 through AE and PCA reduction, respectively. It should be noted that, although both RF and k-NN present a comparable performance, these show noticeable differences between the accuracy reported for the training and test sets, which is an indication of overfitting. Still, k-NN generally performs better than the RF using the same number of input features. It can also be observed that, for low dimensionality, the RF and k-NN models with AE feature reduction outperform the rest, while for higher dimensionalities the models trained on PCA features show better results. This is an important result to choose which dimensionality reduction technique should be used. Regardless of the input dimensionality, the standard deviation for the accuracy does not surpass 5% for all models, except for the SVM with linear kernel (see Tables A1–A5). These low standard deviations values are expected, given that the tested ML algorithms present a stable behavior during the training process. A small standard deviation also shows a consistent performance of the models, which can be related to the preprocessing methodology employed. Indeed, the average accuracy standard deviation obtained for data preprocessed with both PCA and AE does not surpass 1.87% and 2.39%, respectively. This behavior can also be observed for the Q-SVM model, for which the average accuracy standard deviation obtained is 2.26% and 2.56% for datasets preprocessed with PCA and AE, respectively. Unfortunately, none of these algorithms allow one to quantify the prediction uncertainty, which is an ongoing challenge in the PHM community.

One of the main limitations of this work lies in the simplified approach taken towards label generation. The detection time window of one hour prior to a pitch fault alarm is a naïve approach based on the distribution of alarm duration shown in Figure 5. The main drawback of using a fixed time window is that it does not consider that overlapping alarms may be recorded. Hence, further work is required to produce more robust datasets from alarm logs, considering various alarms of interest. Other label generation methodologies defined from maintenance and operational logs for prognostics purposes based on time windows have been proposed by Cofre-Martel et al. [37,70]. The approaches outlined in these articles would enable the use of longer time windows, expanding the prediction horizons to more than 1 h in the future. Additionally, a sensibility analysis is further required to obtain the optimal prediction horizon based on expert knowledge and model performance.

It can be seen from Table 7 that angle encoding results in a better model performance than angular encoding for both PCA and AE (refer to Q-SVM results for four and eight features). While both encoding techniques are lossless operations to translate information from a classical to a quantum setting, using more qubits results in more expressive kernels, as it is the case for angle encoding. This expressiveness allows the algorithm to perform better in downstream tasks, such as classification. For the rest of the cases, quantum-based fault prognostics results are comparable between preprocessing approaches (PCA and AE), and no significant differences are observed between them in terms of final accuracy, indicating the stability of the quantum kernel-based approach against widely different feature extraction techniques. Note that the performance of Q-SVM models is more sensitive to the number of features for the AE than the PCA preprocessing. Figure 9 shows that, for most cases, more features in the original dataset effectively result in better fault prognostics performance, as it is expected from an information point of view. Nevertheless, for the PCA feature reduction approach, a decay in performance is observed when the algorithm is trained using 32 principal components, indicating that the extra components contain a low amount of explained variance and therefore are detrimental to the classification task. On the other hand, for the AE feature reduction approach, no loss in performance is observed when using 19 or 32 features. This may be explained by the fundamental differences between the PCA and the AE feature reduction process, the latter being a nonlinear function optimized based on a reconstruction metric that allows the AE to encode useful information even past the threshold of 19 features. Note that no significant decay in performance is observed when the zero-padding technique is used to allow the application of amplitude encoding to the case of 19 features. This is important since it motivates further exploration in quan-

tum kernel-based fault detection models without the limitation of having to synchronize the number of features to the closest power of two. With respect to the implementation itself, while software libraries allow for a relatively straightforward interface to program simulations of quantum circuits, the execution time vastly surpasses the time necessary to train and evaluate the classical approaches tested in this paper. For example, the Q-SVM training time is in the order of four hours, while the training of traditional SVM approaches takes seconds in modern hardware. The increased amount of training time for Q-SVM models is likely to be due to the simulation process being performed in a classical computer, which is not specialized for quantum operations. The real-time requirements for quantum algorithms will need to be further assessed by the research community once quantum hardware becomes readily accessible. In this regard, this situation is comparable to the origins of other data-driven techniques such as DL before the general availability of GPUs and custom-made software to accelerate the execution of such models (e.g., CUDA).

Comparing the performance of Q-SVM with traditional approaches, it is evident that while the results are within close range with the current available quantum processors and quantum simulators, slightly better performance is obtained using some of the classical techniques for most cases, notably RF models. An outlier in this trend is the case in which the dataset with 32 features generated with an AE was used, where the Q-SVM performance surpasses the RF classifier. Nevertheless, the Q-SVM technique can achieve satisfactory pitch fault imbalance prognostics results. This indicates that the quantum kernel is effectively transforming the original data into a higher dimensional space that is rich enough to allow for the identification of pitch imbalance faults. In addition, the lower performance of the Q-SVM technique could be at least partially attributed to the current state of the art of quantum hardware and quantum simulators, which does not allow for the generation of large encoding circuits capable of leveraging on the whole range of available feature information. Given the state of current quantum technology and algorithms, the fact that the Q-SVM presents a comparable performance with classical counterparts is an important indication of the potential benefits achievable in the near future. This is a key motivation to explore these algorithms as quantum hardware and software are further developed.

To formally assess the statistical difference between the models' performance and compare the ML models with the quantum kernels, a difference of means hypothesis test is performed for the best-performing data reduction configuration. For each model using 19 features processed with AEs, 10 different instances are trained and then tested to obtain a mean and a standard deviation of the models' accuracy. The hypothesis is that the test accuracy distributions are not statistically significant. Thus, the null hypothesis is that the mean test accuracy is the same for each pair of models. The null hypothesis H_0 and the alternative hypothesis H_1 are presented in Equations (18) and (19):

$$H_0 : \bar{x}_1 = \bar{x}_2, \quad (18)$$

$$H_1 : \bar{x}_1 \neq \bar{x}_2 \quad (19)$$

where \bar{x}_1 and \bar{x}_2 are the sample mean for the first and second population samples, respectively. Equation (20) shows the standard error (SE):

$$SE = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}, \quad (20)$$

where s_1, s_2 are the sample standard deviations corresponding to \bar{x}_1 and \bar{x}_2 , respectively. The degrees of freedom (DOF) are then computed as shown in Equation (21):

$$DOF = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1-1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2-1}}. \quad (21)$$

Then, the test statistic is given by Equation (22):

$$t = \frac{\bar{x}_1 - \bar{x}_2}{SE}. \quad (22)$$

The null hypothesis H_0 is rejected if $p < \alpha$, where α is the significance level, which is normally set to a value between 0.05 and 0.10, and p is the corresponding p -value.

Table 8 shows the mean and standard deviation values for each model. Table 9 shows the obtained p -value for each pair of models.

Table 8. Test accuracy sample mean and standard deviation for 10 models using 19 latent space AE.

	Sample Mean	Sample Std
RF	0.902	0.013
k-NN	0.905	0.010
SVM-L	0.919	0.015
SVM-RBF	0.945	0.007
Q-SVM	0.925	0.017

Table 9. p -value for each pair of models using 19 latent space AE.

	k-NN	SVM-L	SVM-RBF	Q-SVM
RF	0.286	0.008	0.000	0.002
k-NN	-	0.013	0.000	0.003
SVM-L		-	0.000	0.207
SVM-RBF			-	0.003

Considering a significance level of $\alpha = 0.05$, the obtained p -values indicate that the Q-SVM outperforms both the k-NN and RF models with a statistically significant advantage. Although further testing is required to confirm the robustness of the Q-SVM approach, this is an interesting result considering the widespread use of k-NN and RF for data-driven fault diagnostics and prognostics tasks. Furthermore, the null hypothesis cannot be rejected when comparing the Q-SVM and SVM-L models. Thus, the performance difference between these two models is inconclusive, and they can be considered as comparable. The SVM-RBF is the only approach that presents a statistically significant higher performance than the Q-SVM. This result, in conjunction with the fact that quantum kernel approaches have just begun to be expanded and tailored by the machine learning research community, encourages further exploration of the technique itself and ways to improve it. Indeed, the presented Q-SVM approach uses a reduced number of qubits, limited by the current maturity of the quantum hardware and simulators available. Yet, Q-SVM obtained comparable and competitive results in terms of performance, even outperforming popular algorithms such as RF and k-NN, particularly when tested with a reduced space of 19 features (See Figure 9 and Table 7). However, the computational complexity of the calculations performed to encode and process the data in Q-SVM results in prohibitive model training and evaluation times when compared to traditional ML models. Nevertheless, the results obtained for Q-SVMs are encouraging, considering QML implementations are expected to improve and become more advantageous from a practical point of view as the quantum hardware evolves, enabling the use of more qubits and therefore enabling the exploration and construction of more complex and representative quantum states.

6. Conclusions

This paper presented a methodology to include the SCADA alarm information into data-driven diagnostic tasks in WTSs focused on detecting pitch fault failures. The number of features in the SCADA sensor data was reduced through two methods: PCA and AE. Following this, several data-driven diagnostics approaches were explored: traditional ML algorithms and quantum kernel ML algorithms. A sensibility analysis on the models' performance was presented regarding the reduced dimensionality of the dataset and feature reduction method.

Overall, the highest performance was achieved with the SVM-RBF model (mean test accuracy of 0.945), while most models present over 0.9 accuracy when 19 features are used. It was also observed that when more than 19 features are used, the overall accuracy of the classification does not improve further, and in some cases it decreases. This is consistent with the CEV and MSE analysis presented for the PCA and AE methods, respectively, where it was shown that, around 19 features, almost all the statistically significant information was extracted from the original dataset. Hence, these results suggest that the optimal feature dimensionality obtained from the feature reduction analysis coincides with the optimal performance of the prognostic models. In this regard, while the fault prognostics models tend to exhibit a slightly higher performance when using AE-based data reduction methods, PCA provides an explainable metric (CEV); therefore, it is an interesting point of comparison. These results highlight the importance of considering alternative metrics other than model performance when selecting the appropriate feature reduction procedures. Ultimately, the chosen method would depend on the application at hand and the user's interpretability requirements.

In general, the results obtained for the quantum kernel show comparable performance levels when compared with classical approaches. Indeed, when comparing the classical and quantum-based diagnostic methods using 19 features preprocessed with the AE, the Q-SVM presents a statistically significant advantage over the k-NN and RF models ($\alpha = 0.05$). Further, while the performance of the SVM-RBF models surpasses the Q-SVM, the results of the latter were comparable with those obtained with the SVM-L model. Regarding the practical implications of these results, comparable results were achieved between the proposed Q-SVM method and established approaches. As quantum hardware evolves and becomes readily available, it is expected for QML algorithms to increase in complexity and representational capacity, possibly surpassing traditional ML models. QML has just recently begun to be explored outside the quantum computing research community, so its early testing in practical applications, such as the case study presented in this work, allows the PHM community to assess its potential and identify future research paths within the field. The authors believe that, based on the results obtained for this case study, quantum kernel-based fault prognostics algorithms merit further research in the advent of the further development and general availability of quantum computers and quantum simulators, which is expected to occur during this decade.

Author Contributions: Conceptualization, C.C.-J., S.C.-M., G.S.M. and E.L.D.; methodology, C.C.-J., S.C.-M., G.S.M. and E.L.D.; software, C.C.-J., S.C.-M. and G.S.M.; validation, C.C.-J., S.C.-M. and G.S.M.; formal analysis, C.C.-J., S.C.-M., G.S.M. and E.L.D.; investigation, C.C.-J., S.C.-M., G.S.M. and E.L.D.; resources, E.L.D., G.d.N.P.L. and A.C.; data curation, C.C.-J. and S.C.-M.; writing—original draft preparation, C.C.-J., S.C.-M. and G.S.M.; writing—review and editing, E.L.D.; visualization, C.C.-J., S.C.-M. and G.S.M.; supervision, E.L.D.; project administration, E.L.D.; funding acquisition, E.L.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: An adapted version of the data presented in this study is available on request from the corresponding author. The original raw data are not publicly available due to their proprietary nature.

Acknowledgments: Sergio Cofré Martel would like to thank the Agencia Nacional de Investigación y Desarrollo (ANID—Doctorados Becas Chile-72190097).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

AE	Autoencoder
CBM	Condition-based Maintenance
CES	Complex Engineering System
CEV	Cumulative Explained Variance
DDM	Data-driven Model
DL	Deep Learning
k-NN	k-Nearest Neighbors
LCOE	Levelized Cost of Energy
LSTM	Long Short-Term Memory
ML	Machine Learning
MSE	Mean Squared Error
NN	Neural Network
PC	Principal Component
PCA	Principal Component Analysis
PHM	Prognostics and Health Management
QML	Quantum Machine Learning
RBF	Radial Basis Function
RF	Random Forest
SCADA	Supervisory Control and Data
SVM	Support Vector Machines
WTS	Wind Turbine System

Appendix A

Table A1. Binary classification metrics for RF per number of PCs and AE latent space dimensionality.

Features	Model	PCA				AE			
		Accuracy	Recall	Prec.	F1	Accuracy	Recall	Prec.	F1
4	Mean Train	0.868	0.898	0.847	0.870	0.907	0.921	0.902	0.910
	SD Train	0.036	0.046	0.047	0.035	0.032	0.045	0.048	0.031
	Mean Test	0.871	0.903	0.853	0.877	0.877	0.911	0.849	0.879
	SD Test	0.022	0.030	0.026	0.022	0.017	0.020	0.028	0.017
8	Mean Train	0.898	0.936	0.877	0.905	0.897	0.928	0.879	0.902
	SD Train	0.022	0.034	0.026	0.021	0.041	0.045	0.045	0.039
	Mean Test	0.887	0.934	0.849	0.889	0.885	0.912	0.864	0.887
	SD Test	0.019	0.008	0.036	0.018	0.022	0.026	0.024	0.020
16	Mean Train	0.910	0.936	0.894	0.914	0.895	0.912	0.882	0.896
	SD Train	0.033	0.038	0.043	0.032	0.035	0.040	0.034	0.035
	Mean Test	0.909	0.937	0.886	0.911	0.899	0.924	0.879	0.901
	SD Test	0.015	0.025	0.031	0.017	0.015	0.019	0.018	0.014
19	Mean Train	0.926	0.937	0.921	0.928	0.920	0.940	0.904	0.921
	SD Train	0.031	0.049	0.033	0.031	0.019	0.045	0.019	0.021
	Mean Test	0.916	0.941	0.897	0.919	0.902	0.931	0.881	0.905
	SD Test	0.015	0.015	0.020	0.015	0.013	0.009	0.022	0.014
32	Mean Train	0.923	0.943	0.909	0.925	0.906	0.942	0.886	0.912
	SD Train	0.015	0.029	0.030	0.014	0.040	0.034	0.057	0.038
	Mean Test	0.906	0.923	0.894	0.908	0.894	0.936	0.863	0.898
	SD Test	0.017	0.019	0.033	0.017	0.013	0.018	0.014	0.014

Table A2. Binary classification metrics for k-NN per number of PCs and AE latent space dimensionality.

Features	Model	PCA				AE			
		Accuracy	Recall	Prec.	F1	Accuracy	Recall	Prec.	F1
4	Mean Train	0.871	0.916	0.842	0.876	0.915	0.946	0.897	0.920
	SD Train	0.030	0.040	0.052	0.028	0.030	0.025	0.044	0.027
	Mean Test	0.869	0.901	0.851	0.875	0.888	0.934	0.853	0.891
	SD Test	0.018	0.027	0.022	0.017	0.013	0.020	0.025	0.012
8	Mean Train	0.901	0.951	0.872	0.909	0.921	0.919	0.926	0.922
	SD Train	0.021	0.031	0.027	0.019	0.029	0.048	0.023	0.029
	Mean Test	0.882	0.930	0.844	0.884	0.904	0.933	0.881	0.906
	SD Test	0.019	0.022	0.034	0.020	0.017	0.019	0.024	0.014
16	Mean Train	0.914	0.936	0.897	0.916	0.912	0.917	0.909	0.912
	SD Train	0.019	0.020	0.029	0.019	0.024	0.031	0.037	0.023
	Mean Test	0.904	0.940	0.876	0.906	0.912	0.922	0.905	0.913
	SD Test	0.016	0.025	0.031	0.018	0.012	0.021	0.020	0.011
19	Mean Train	0.924	0.948	0.911	0.928	0.924	0.940	0.912	0.925
	SD Train	0.032	0.037	0.053	0.029	0.013	0.037	0.019	0.014
	Mean Test	0.910	0.937	0.890	0.912	0.905	0.934	0.884	0.908
	SD Test	0.016	0.015	0.026	0.016	0.010	0.025	0.031	0.013
32	Mean Train	0.920	0.952	0.896	0.923	0.915	0.960	0.887	0.921
	SD Train	0.024	0.030	0.033	0.022	0.046	0.033	0.059	0.043
	Mean Test	0.906	0.932	0.887	0.909	0.904	0.937	0.880	0.907
	SD Test	0.020	0.023	0.035	0.020	0.010	0.022	0.028	0.011

Table A3. Binary classification metrics for SVM-linear kernel per number of PCs and AE latent space dimensionality.

Features	Model	PCA				AE			
		Accuracy	Recall	Prec.	F1	Accuracy	Recall	Prec.	F1
4	Mean Train	0.824	0.869	0.798	0.832	0.771	0.746	0.786	0.765
	SD Train	0.019	0.009	0.024	0.016	0.026	0.051	0.030	0.031
	Mean Test	0.819	0.862	0.795	0.827	0.747	0.731	0.755	0.742
	SD Test	0.042	0.048	0.040	0.040	0.072	0.087	0.068	0.076
8	Mean Train	0.878	0.927	0.844	0.884	0.907	0.953	0.872	0.911
	SD Train	0.009	0.006	0.015	0.007	0.017	0.013	0.020	0.016
	Mean Test	0.854	0.909	0.821	0.862	0.894	0.931	0.868	0.898
	SD Test	0.034	0.030	0.041	0.030	0.025	0.022	0.027	0.023
16	Mean Train	0.912	0.943	0.889	0.915	0.903	0.949	0.869	0.907
	SD Train	0.012	0.016	0.020	0.011	0.022	0.020	0.024	0.021
	Mean Test	0.890	0.924	0.866	0.894	0.883	0.928	0.852	0.888
	SD Test	0.003	0.022	0.018	0.003	0.036	0.051	0.033	0.035
19	Mean Train	0.897	0.927	0.874	0.900	0.925	0.976	0.886	0.929
	SD Train	0.008	0.005	0.012	0.007	0.009	0.004	0.013	0.009
	Mean Test	0.898	0.922	0.879	0.900	0.919	0.975	0.877	0.923
	SD Test	0.016	0.020	0.014	0.016	0.015	0.011	0.025	0.013
32	Mean Train	0.914	0.942	0.893	0.916	0.923	0.981	0.879	0.927
	SD Train	0.014	0.019	0.019	0.013	0.023	0.015	0.030	0.021
	Mean Test	0.902	0.915	0.892	0.903	0.927	0.971	0.895	0.931
	SD Test	0.014	0.020	0.024	0.013	0.031	0.019	0.050	0.027

Table A4. Binary classification metrics for SVM-RBF kernel per number of PCs and AE latent space dimensionality.

Features	Model	PCA				AE			
		Accuracy	Recall	Prec.	F1	Accuracy	Recall	Prec.	F1
4	Mean Train	0.858	0.902	0.829	0.864	0.887	0.930	0.856	0.891
	SD Train	0.018	0.028	0.015	0.018	0.020	0.020	0.020	0.019
	Mean Test	0.846	0.885	0.822	0.852	0.865	0.894	0.848	0.870
	SD Test	0.023	0.021	0.036	0.018	0.042	0.027	0.060	0.038
8	Mean Train	0.897	0.939	0.867	0.902	0.931	0.959	0.908	0.933
	SD Train	0.012	0.007	0.020	0.011	0.009	0.004	0.016	0.008
	Mean Test	0.887	0.938	0.851	0.892	0.917	0.950	0.892	0.920
	SD Test	0.022	0.021	0.025	0.020	0.028	0.029	0.031	0.026
16	Mean Train	0.925	0.961	0.897	0.928	0.927	0.975	0.889	0.930
	SD Train	0.015	0.008	0.023	0.014	0.015	0.005	0.022	0.014
	Mean Test	0.920	0.955	0.893	0.923	0.909	0.989	0.857	0.917
	SD Test	0.009	0.017	0.009	0.009	0.039	0.017	0.057	0.033
19	Mean Train	0.932	0.955	0.914	0.934	0.954	0.985	0.927	0.955
	SD Train	0.014	0.010	0.017	0.013	0.004	0.002	0.005	0.003
	Mean Test	0.897	0.927	0.875	0.900	0.945	0.978	0.918	0.947
	SD Test	0.027	0.033	0.030	0.025	0.007	0.006	0.013	0.006
32	Mean Train	0.938	0.957	0.922	0.939	0.939	0.988	0.901	0.942
	SD Train	0.008	0.011	0.013	0.007	0.015	0.005	0.024	0.013
	Mean Test	0.921	0.947	0.900	0.923	0.912	0.983	0.865	0.919
	SD Test	0.006	0.022	0.015	0.007	0.040	0.009	0.062	0.034

Table A5. Binary classification metrics for SVM-quantum kernel per number of PCs and AE latent space dimensionality.

Features	Model	PCA				AE			
		Accuracy	Recall	Prec.	F1	Accuracy	Recall	Prec.	F1
4 [Angle Encoding]	Mean Train	0.854	0.884	0.834	0.859	0.846	0.872	0.832	0.851
	SD Train	0.015	0.017	0.014	0.015	0.045	0.038	0.060	0.040
	Mean Test	0.811	0.842	0.794	0.817	0.827	0.861	0.806	0.832
	SD Test	0.028	0.027	0.041	0.024	0.037	0.060	0.040	0.039
4 [Amplitude Encoding]	Mean Train	0.824	0.858	0.803	0.830	0.810	0.816	0.808	0.812
	SD Train	0.015	0.019	0.013	0.015	0.036	0.022	0.050	0.032
	Mean Test	0.819	0.847	0.803	0.824	0.784	0.763	0.802	0.779
	SD Test	0.032	0.050	0.023	0.034	0.022	0.059	0.059	0.018
8 [Angle Encoding]	Mean Train	0.890	0.939	0.854	0.895	0.930	0.963	0.904	0.933
	SD Train	0.019	0.019	0.019	0.018	0.025	0.020	0.033	0.024
	Mean Test	0.888	0.938	0.853	0.893	0.894	0.931	0.868	0.898
	SD Test	0.017	0.027	0.027	0.016	0.037	0.050	0.040	0.036
8 [Amplitude Encoding]	Mean Train	0.866	0.913	0.835	0.872	0.868	0.932	0.826	0.876
	SD Train	0.014	0.014	0.016	0.013	0.015	0.025	0.015	0.014
	Mean Test	0.856	0.899	0.828	0.862	0.860	0.915	0.825	0.867
	SD Test	0.023	0.033	0.022	0.023	0.021	0.044	0.031	0.020
16	Mean Train	0.896	0.931	0.870	0.899	0.875	0.979	0.812	0.887
	SD Train	0.008	0.011	0.010	0.008	0.018	0.018	0.025	0.014
	Mean Test	0.897	0.929	0.873	0.900	0.874	0.974	0.812	0.885
	SD Test	0.018	0.018	0.020	0.018	0.025	0.022	0.031	0.020

Table A5. Cont.

Features	Model	PCA				AE			
		Accuracy	Recall	Prec.	F1	Accuracy	Recall	Prec.	F1
19	Mean Train	0.895	0.928	0.871	0.898	0.932	0.979	0.895	0.935
	SD Train	0.011	0.010	0.012	0.010	0.005	0.007	0.009	0.005
	Mean Test	0.898	0.929	0.876	0.901	0.925	0.976	0.886	0.928
	SD Test	0.012	0.006	0.018	0.011	0.017	0.007	0.024	0.015
32	Mean Train	0.910	0.938	0.888	0.912	0.911	0.986	0.859	0.918
	SD Train	0.014	0.019	0.014	0.014	0.021	0.016	0.029	0.018
	Mean Test	0.886	0.913	0.867	0.889	0.910	0.967	0.870	0.915
	SD Test	0.028	0.030	0.034	0.027	0.020	0.038	0.025	0.020

References

- International Energy Agency (IEA). *Renewable Energy Market Update 2021*; International Energy Agency: Paris, France, 2021.
- Global Wind Energy Council GWEC. *Global Wind Report 2021*; Global Wind Energy Council: Brussels, Belgium, 2021.
- Moghadam, F.K.; Nejad, A.R. Online Condition Monitoring of Floating Wind Turbines Drivetrain by Means of Digital Twin. *Mech. Syst. Signal Process.* **2022**, *162*, 108087. [\[CrossRef\]](#)
- McKinnon, C.; Carroll, J.; McDonald, A.; Koukoura, S.; Plumley, C. Investigation of Isolation Forest for Wind Turbine Pitch System Condition Monitoring Using SCADA Data. *Energies* **2021**, *14*, 6601. [\[CrossRef\]](#)
- Carroll, J.; McDonald, A.; McMillan, D. Failure Rate, Repair Time and Unscheduled O&M Cost Analysis of Offshore Wind Turbines. *Wind Energy* **2016**, *19*, 1107–1119. [\[CrossRef\]](#)
- Jia, X.; Han, Y.; Li, Y.; Sang, Y.; Zhang, G. Condition Monitoring and Performance Forecasting of Wind Turbines Based on Denoising Autoencoder and Novel Convolutional Neural Networks. *Energy Rep.* **2021**, *7*, 6354–6365. [\[CrossRef\]](#)
- Hossain, M.; Abu-Siada, A.; Muyeen, S. Methods for Advanced Wind Turbine Condition Monitoring and Early Diagnosis: A Literature Review. *Energies* **2018**, *11*, 1309. [\[CrossRef\]](#)
- Leahy, K.; Gallagher, C.; O'Donovan, P.; Bruton, K.; O'Sullivan, D.T.J.J. A Robust Prescriptive Framework and Performance Metric for Diagnosing and Predicting Wind Turbine Faults Based on SCADA and Alarms Data with Case Study. *Energies* **2018**, *11*, 1738. [\[CrossRef\]](#)
- Do, M.H.; Söffker, D. State-of-the-Art in Integrated Prognostics and Health Management Control for Utility-Scale Wind Turbines. *Renew. Sustain. Energy Rev.* **2021**, *145*, 111102. [\[CrossRef\]](#)
- Marti-Puig, P.; Bennásar-Sevillá, A.; Blanco, M.A.; Solé-Casals, J. Exploring the Effect of Temporal Aggregation on SCADA Data for Wind Turbine Prognosis Using a Normality Model. *Appl. Sci.* **2021**, *11*, 6405. [\[CrossRef\]](#)
- Bailey, D.; Wright, E. *Practical SCADA for Industry*; Elsevier: Amsterdam, The Netherlands, 2003.
- Marti-Puig, P.; Blanco-M, A.; Cárdenas, J.; Cusidó, J.; Solé-Casals, J. Feature Selection Algorithms for Wind Turbine Failure Prediction. *Energies* **2019**, *12*, 453. [\[CrossRef\]](#)
- Lebranchu, A.; Charbonnier, S.; Bérenguer, C.; Prévost, F. A Combined Mono- and Multi-Turbine Approach for Fault Indicator Synthesis and Wind Turbine Monitoring Using SCADA Data. *ISA Trans.* **2019**, *87*, 272–281. [\[CrossRef\]](#)
- Qiu, Y.; Feng, Y.; Infield, D. Fault Diagnosis of Wind Turbine with SCADA Alarms Based Multidimensional Information Processing Method. *Renew. Energy* **2020**, *145*, 1923–1931. [\[CrossRef\]](#)
- Velandia-Cardenas, C.; Vidal, Y.; Pozo, F. Wind Turbine Fault Detection Using Highly Imbalanced Real SCADA Data. *Energies* **2021**, *14*, 1728. [\[CrossRef\]](#)
- Reder, M.D.; Gonzalez, E.; Melero, J.J. Wind Turbine Failures—Tackling Current Problems in Failure Data Analysis. *J. Phys. Conf. Ser.* **2016**, *753*, 072027. [\[CrossRef\]](#)
- Van Kuik, G.A.M.; Peinke, J.; Nijssen, R.; Lekou, D.; Mann, J.; Sørensen, J.N.; Ferreira, C.; Van Wingerden, J.W.; Schlipf, D.; Gebraad, P.; et al. Long-Term Research Challenges in Wind Energy—A Research Agenda by the European Academy of Wind Energy. *Wind Energy Sci.* **2016**, *1*, 1–39. [\[CrossRef\]](#)
- Leahy, K.; Gallagher, C.; O'Donovan, P.; O'Sullivan, D.T.J. Issues with Data Quality for Wind Turbine Condition Monitoring and Reliability Analyses. *Energies* **2019**, *12*, 201. [\[CrossRef\]](#)
- Habibi, H.; Howard, I.; Simani, S. Reliability Improvement of Wind Turbine Power Generation Using Model-Based Fault Detection and Fault Tolerant Control: A Review. *Renew. Energy* **2019**, *135*, 877–896. [\[CrossRef\]](#)
- Panda, A.K.; Rapur, J.S.; Tiwari, R. Prediction of Flow Blockages and Impending Cavitation in Centrifugal Pumps Using Support Vector Machine (SVM) Algorithms Based on Vibration Measurements. *Measurement* **2018**, *130*, 44–56. [\[CrossRef\]](#)
- Cakir, M.; Guvenc, M.A.; Mistikoglu, S. The Experimental Application of Popular Machine Learning Algorithms on Predictive Maintenance and the Design of IIoT Based Condition Monitoring System. *Comput. Ind. Eng.* **2021**, *151*, 106948. [\[CrossRef\]](#)
- Lim, I.S.; Park, J.Y.; Choi, E.J.; Kim, M.S. Efficient Fault Diagnosis Method of PEMFC Thermal Management System for Various Current Densities. *Int. J. Hydrogen Energy* **2021**, *46*, 2543–2554. [\[CrossRef\]](#)

23. Stetco, A.; Dinmohammadi, F.; Zhao, X.; Robu, V.; Flynn, D.; Barnes, M.; Keane, J.; Nenadic, G. Machine Learning Methods for Wind Turbine Condition Monitoring: A Review. *Renew. Energy* **2019**, *133*, 620–635. [[CrossRef](#)]
24. Chen, H.; Liu, H.; Chu, X.; Liu, Q.; Xue, D. Anomaly Detection and Critical SCADA Parameters Identification for Wind Turbines Based on LSTM-AE Neural Network. *Renew. Energy* **2021**, *172*, 829–840. [[CrossRef](#)]
25. Encalada-Dávila, Á.; Puruncajas, B.; Tutivén, C.; Vidal, Y. Wind Turbine Main Bearing Fault Prognosis Based Solely on SCADA Data. *Sensors* **2021**, *21*, 2228. [[CrossRef](#)] [[PubMed](#)]
26. Fink, O.; Wang, Q.; Svensén, M.; Dersin, P.; Lee, W.J.; Ducoffe, M. Potential, Challenges and Future Directions for Deep Learning in Prognostics and Health Management Applications. *Eng. Appl. Artif. Intell.* **2020**, *92*, 103678. [[CrossRef](#)]
27. Bergholm, V.; Izaac, J.; Schuld, M.; Gogolin, C.; Alam, M.S.; Ahmed, S.; Arrazola, J.M.; Blank, C.; Delgado, A.; Jahangiri, S.; et al. PennyLane: Automatic Differentiation of Hybrid Quantum-Classical Computations. *arXiv* **2018**, arXiv:1811.04968. [[CrossRef](#)]
28. Aleksandrowicz, G.; Alexander, T.; Barkoutsos, P.; Bello, L.; Ben-Haim, Y.; Bucher, D.; Cabrera-Hernández, F.J.; Carballo-Franquis, J.; Chen, A.; Chen, C.-F.; et al. Qiskit: An Open-Source Framework for Quantum Computing. *Zenodo* **2019**. [[CrossRef](#)]
29. Developers, C. Cirq. *Zenodo* **2021**. [[CrossRef](#)]
30. Shor, P.W. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134. [[CrossRef](#)]
31. Fernandez-Carames, T.M.; Fraga-Lamas, P. Towards Post-Quantum Blockchain: A Review on Blockchain Cryptography Resistant to Quantum Computing Attacks. *IEEE Access* **2020**, *8*, 21091–21116. [[CrossRef](#)]
32. Zhang, K.; Rao, P.; Yu, K.; Lim, H.; Korepin, V. Implementation of Efficient Quantum Search Algorithms on NISQ Computers. *Quantum Inf. Process.* **2021**, *20*, 1–27. [[CrossRef](#)]
33. Zhang, J.; Huang, Z.; Li, X.; Wu, M.; Wang, X.; Dong, Y. Quantum Image Encryption Based on Quantum Image Decomposition. *Int. J. Theor. Phys.* **2021**, *60*, 2930–2942. [[CrossRef](#)]
34. Farhi, E.; Goldstone, J.; Gutmann, S. A Quantum Approximate Optimization Algorithm. *arXiv* **2014**, arXiv:1411.4028. [[CrossRef](#)]
35. Benedetti, M.; Lloyd, E.; Sack, S.; Fiorentini, M. Parameterized Quantum Circuits as Machine Learning Models. *Quantum Sci. Technol.* **2019**, *4*, 043001. [[CrossRef](#)]
36. Schuld, M. Supervised Quantum Machine Learning Models Are Kernel Methods. *arXiv* **2021**, arXiv:2101.11020. [[CrossRef](#)]
37. Cofre-Martel, S.; Droguett, E.L.; Modarres, M.; Antonino-Daviu, J.A.; Caesarendra, W.; Lopez Droguett, E.; Modarres, M. Big Machinery Data Preprocessing Methodology for Data-Driven Models in Prognostics and Health Management. *Sensors* **2021**, *21*, 6841. [[CrossRef](#)] [[PubMed](#)]
38. Nasiri, S.; Khosravani, M.R. Machine Learning in Predicting Mechanical Behavior of Additively Manufactured Parts. *J. Mater. Res. Technol.* **2021**, *14*, 1137–1153. [[CrossRef](#)]
39. San Martin, G.; López Droguett, E.; Meruane, V.; das Chagas Moura, M. Deep Variational Auto-Encoders: A Promising Tool for Dimensionality Reduction and Ball Bearing Elements Fault Diagnosis. *Struct. Health Monit.* **2019**, *18*, 1092–1128. [[CrossRef](#)]
40. Cofre-Martel, S.; Kobrich, P.; Lopez Droguett, E.; Meruane, V. Deep Convolutional Neural Network-Based Structural Damage Localization and Quantification Using Transmissibility Data. *Shock Vib.* **2019**, *2019*, 9859281. [[CrossRef](#)]
41. Barraza, J.F.; Droguett, E.L.; Naranjo, V.M.; Martins, M.R. Capsule Neural Networks for Structural Damage Localization and Quantification Using Transmissibility Data. *Appl. Soft Comput.* **2020**, *97*, 106732. [[CrossRef](#)]
42. Correa-Jullian, C.; Cardemil, J.M.; López Droguett, E.; Behzad, M. Assessment of Deep Learning Techniques for Prognosis of Solar Thermal Systems. *Renew. Energy* **2020**, *145*, 2178–2191. [[CrossRef](#)]
43. Correa Jullian, C.; Cardemil, J.; López Droguett, E.; Behzad, M. Assessment of Deep Learning Algorithms for Fault Diagnosis of Solar Thermal Systems. In Proceedings of the ISES Solar World Congress 2019, Santiago, Chile, 4–7 November 2019; International Solar Energy Society: Freiburg, Germany, 2019; pp. 1–12.
44. Cofre-Martel, S.; Lopez Droguett, E.; Modarres, M. Remaining Useful Life Estimation through Deep Learning Partial Differential Equation Models: A Framework for Degradation Dynamics Interpretation Using Latent Variables. *Shock Vib.* **2021**, *2021*, 9937846. [[CrossRef](#)]
45. Zaher, A.; McArthur, S.D.J.; Infield, D.G.; Patel, Y. Online Wind Turbine Fault Detection through Automated SCADA Data Analysis. *Wind Energy* **2009**, *12*, 574–593. [[CrossRef](#)]
46. Marvuglia, A.; Messineo, A. Monitoring of Wind Farms' Power Curves Using Machine Learning Techniques. *Appl. Energy* **2012**, *98*, 574–583. [[CrossRef](#)]
47. Bangalore, P.; Tjernberg, L.B. An Artificial Neural Network Approach for Early Fault Detection of Gearbox Bearings. *IEEE Trans. Smart Grid* **2015**, *6*, 980–987. [[CrossRef](#)]
48. Xiang, L.; Wang, P.; Yang, X.; Hu, A.; Su, H. Fault Detection of Wind Turbine Based on SCADA Data Analysis Using CNN and LSTM with Attention Mechanism. *Measurement* **2021**, *175*, 109094. [[CrossRef](#)]
49. Wu, Y.; Ma, X. A Hybrid LSTM-KLD Approach to Condition Monitoring of Operational Wind Turbines. *Renew. Energy* **2022**, *181*, 554–566. [[CrossRef](#)]
50. Mazidi, P.; Tjernberg, L.B.; Sanz Bobi, M.A. Wind Turbine Prognostics and Maintenance Management Based on a Hybrid Approach of Neural Networks and a Proportional Hazards Model. *J Risk Reliab.* **2017**, *231*, 121–129. [[CrossRef](#)]
51. Marti-Puig, P.; Blanco-M, A.; Cárdenas, J.J.; Cusidó, J.; Solé-Casals, J. Effects of the Pre-Processing Algorithms in Fault Diagnosis of Wind Turbines. *Environ. Model. Softw.* **2018**, *110*, 119–128. [[CrossRef](#)]

52. Mishra, V.; Rath, S.K. Detection of Breast Cancer Tumours Based on Feature Reduction and Classification of Thermograms. *Quant. Infrared Thermogr. J.* **2021**, *18*, 300–313. [[CrossRef](#)]
53. Castellani, F.; Astolfi, D.; Natili, F. Scada Data Analysis Methods for Diagnosis of Electrical Faults to Wind Turbine Generators. *Appl. Sci.* **2021**, *11*, 3307. [[CrossRef](#)]
54. Wang, Y.; Ma, X.; Qian, P. Wind Turbine Fault Detection and Identification Through PCA-Based Optimal Variable Selection. *IEEE Trans. Sustain. Energy* **2018**, *9*, 1627–1635. [[CrossRef](#)]
55. Zhao, Y.; Li, D.; Dong, A.; Kang, D.; Lv, Q.; Shang, L. Fault Prediction and Diagnosis of Wind Turbine Generators Using SCADA Data. *Energies* **2017**, *10*, 1210. [[CrossRef](#)]
56. Pozo, F.; Vidal, Y.; Serrahima, J.M. On Real-Time Fault Detection in Wind Turbines: Sensor Selection Algorithm and Detection Time Reduction Analysis. *Energies* **2016**, *9*, 520. [[CrossRef](#)]
57. Pozo, F.; Vidal, Y. Wind Turbine Fault Detection through Principal Component Analysis and Statistical Hypothesis Testing. *Energies* **2015**, *9*, 3. [[CrossRef](#)]
58. Yanofsky, N.S.; Mannucci, M.A. *Quantum Computing for Computer Scientists*; Cambridge University Press: Cambridge, UK, 2008; pp. 1–384, ISBN 9780521879965. [[CrossRef](#)]
59. Sutor, R.S. *Dancing with Qubits: How Quantum Computing Works and How It Can Change the World*; Packt Publishing Ltd.: Birmingham, UK, 2019.
60. Larose, R.; Coyle, B. Robust Data Encodings for Quantum Classifiers. *Phys. Rev. A* **2020**, *102*, 032420. [[CrossRef](#)]
61. Stoudenmire, E.M.; Schwab, D.J. Supervised Learning with Quantum-Inspired Tensor Networks. *Adv. Neural Inf. Processing Syst.* **2016**, *29*, 4799. [[CrossRef](#)]
62. San, G.; Silva, M.; Droguett, E.L. Quantum Machine Learning for Health State Diagnosis and Prognostics. *arXiv* **2021**, arXiv:2108.12265. [[CrossRef](#)]
63. Schuld, M.; Killoran, N. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* **2018**, *122*, 040504. [[CrossRef](#)]
64. Cho, S.; Choi, M.; Gao, Z.; Moan, T. Fault Detection and Diagnosis of a Blade Pitch System in a Floating Wind Turbine Based on Kalman Filters and Artificial Neural Networks. *Renew. Energy* **2021**, *169*, 1–13. [[CrossRef](#)]
65. Cho, S.; Gao, Z.; Moan, T. Model-Based Fault Detection, Fault Isolation and Fault-Tolerant Control of a Blade Pitch System in Floating Wind Turbines. *Renew. Energy* **2018**, *120*, 306–321. [[CrossRef](#)]
66. Qiu, Y.; Feng, Y.; Tavner, P.; Richardson, P.; Erdos, G.; Chen, B. Wind Turbine SCADA Alarm Analysis for Improving Reliability. *Wind Energy* **2012**, *15*, 951–966. [[CrossRef](#)]
67. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
68. Liu, T.; Wang, Z.; Zeng, J.; Wang, J. Machine-Learning-Based Models to Predict Shear Transfer Strength of Concrete Joints. *Eng. Struct.* **2021**, *249*, 113253. [[CrossRef](#)]
69. Partridge, M. Fast Dimensionality Reduction and Simple PCA. *Intell. Data Anal.* **1998**, *2*, 203–214. [[CrossRef](#)]
70. Cofre-Martel, S.; Correa-Jullian, C.; López Droguett, E.; Groth, K.M.; Modarres, M.M. Defining Degradation States for Diagnosis Classification Models in Real Systems Based on Monitoring Data. In Proceedings of the 31st European Safety and Reliability Conference (ESREL 2021), Angers, France, 19–23 September 2021; Research Publishing Services: Singapore, 2021; pp. 1286–1293.