Early Fault Detection in the Main Bearing of Wind Turbines Based on Gated Recurrent Unit (GRU) Neural Networks and SCADA Data

Ángel Encalada-Dávila, Luis Moyón, Christian Tutivén, Bryan Puruncajas, and Yolanda Vidal, *Senior Member, IEEE*

***Abstract*—Failures in the main bearings of wind turbines are critical in terms of downtime and replacement cost. Early diagnosis of their faults would lower the levelized cost of wind energy. Thus, this work discusses a gated recurrent unit (GRU) neural network, which detects faults in the main bearing some months ahead (when the event that initiates/develops the failure releases heat) the actual fatal fault materializes. GRUs feature internal gates that govern information flow and are utilized in this study for their capacity to understand whether data in a time series is crucial enough to preserve or forget. It is noteworthy that the proposed methodology only requires healthy Supervisory Control and Data Acquisition (SCADA) data. Thus, it can be deployed to old wind parks (nearing the end of their lifespan) where specific high frequency condition monitoring sensors are not installed and to new wind parks where faulty historical data do not exist yet. The strategy is trained, validated, and finally tested using SCADA data from an in-production wind park composed of nine wind turbines.**

***Index Terms*—early fault detection, wind turbine, main bearing, gated recurrent unit neural network, anomaly detection, SCADA data.**

I. Introduction

THE global energy system is undeniably in flux. Renewable energy uptake and utilization are critical to combating climate change and ensuring a long-term future. Renewable electricity will be key to Europe reaching climate neutrality by 2050, according to the European Commission's projections [1]. Wind energy is a critical component in achieving this goal, as it accounts for 50% of the European Union's power mix, with renewables accounting for 81%. The core of the problem in the progress of the wind business, however, is a decrease in the levelized cost of electricity (LCOE). The LCOE of a wind park includes many parameters such as total installed costs, lifetime capacity factor, operation and maintenance (O&M) expenses, project economic lifespan, and cost of capital. While all these criteria play a role in calculating a project's LCOE, some have a greater influence. Wind energy O&M expenses, particularly, account for a major portion of the LCOE (20-25 percent for onshore wind parks and 25-30 percent for offshore wind parks) [2]. Hence, optimizing maintenance procedures forms a critical component of achieving low-cost wind energy.

In any industrial-scale wind park, energy output losses due to unforeseen asset maintenance, as well as component replacement costs, can amount up to millions of euros each year. Hence, it is critical to transition from corrective (replacing failed parts) and preventive maintenance (planned at regular intervals without regard for the asset's actual state) to predictive maintenance, which is based on actual and timely data collected through a network of sensors monitoring the actual asset (performed using high-frequency data of physical quantities) and warns operators in advance before the fatal break-down materializes, enabling them to program, repairs to match with weather or production windows, thus lowering costs. To better use the information in the vast quantity of data (gathered continuously or periodically, online or offline) from diverse sensors obtained from the assets, digitalization and artificial intelligence are crucial technologies. The main concept is to identify changes in the situation that suggest a growing malfunction and reflect departures from typical operational procedures.

This paper provides an early (months ahead of time) defect detection technique for the main bearing of a wind turbine (WT) based on a GRU neural network (NN) that employs just SCADA data to this framework. As SCADA data is primarily used for operation and control rather than condition monitoring (CM), using it for this purpose is a significant issue. SCADA data contains approximately 200 different variables (i.e., it is high dimensional), has a low sampling rate (recorded continuously at 10-minute averaged intervals to minimize data transmission bandwidth and storage), depends on the WT's

The authors are grateful to the Smartive firm since this study would not have been feasible without their provided wind park data. Additionally, this work has been partially funded by the Spanish Agencia Estatal de Investigación (AEI) - Ministerio de Economía, Industria y Competitividad (MINECO), and the Fondo Europeo de Desarrollo Regional (FEDER) through the research project DPI2017-82930-C2-1-R; and by the Generalitat de Catalunya through the research project 2017 SGR 388.

Corresponding author: Yolanda Vidal (yolanda.vidal@upc.edu)
Ángel Encalada-Dávila, Christian Tutivén, and Bryan Puruncajas are with the Escuela Superior Politécnica del Litoral, ESPOL, Faculty of Mechanical Engineering and Production Science, FIMCP, Mechatronics Engineering, Campus Gustavo Galindo Km. 30.5 Vía Perimetral, P.O. Box 09-01-5863, Guayaquil, Ecuador (e-mail: angaenca@espol.edu.ec, cjtutive@espol.edu.ec, and bpurunca@espol.edu.ec).
Luís Moyón is with Universidad Ecotec, Km. 13.5 Samborondón, Samborondón, EC092302, Ecuador (e-mail: amoyon@hotmail.com).
Bryan Puruncajas and Yolanda Vidal are with the Research Group of Control, Data, and Artificial Intelligence (CoDAlab), Department of Mathematics, Escola d'Enginyeria de Barcelona Est (EEBE), Universitat Politècnica de Catalunya (UPC), Campus Diagonal-Besós (CDB), Eduard Maristany, 16, 08019 Barcelona, Spain.
Yolanda Vidal is also with the Institute of Mathematics (IMTech), Universitat Politècnica de Catalunya (UPC), Pau Gargallo 14, 08028 Barcelona, Spain (e-mail: yolanda.vidal@upc.edu).

All authors contributed equally to this work.

region of operation and environmental conditions, and is a time series with strong seasonality. Furthermore, the benefit of keeping consistent maintenance work order records with full fault descriptions was unknown when SCADA systems were established (as it was not envisioned that AI could help in this application). Operators and maintenance contractors record maintenance repair operations in a number of ways, ranging from handwritten forms that detail any work done in an unstructured format to highly automated work orders. Furthermore, most of the data comes from routine operations, resulting in very imbalanced data sets. Despite these challenges, the idea of leveraging SCADA data for predictive maintenance has recently received more attention. Many obstacles, however, remain to be solved in present and future studies. The next paragraphs quickly assesses the most important studies in the field, which demonstrate its potential while also underlining the research hurdles.

For starters, supervised algorithms are used in a large percentage of articles (classification methods). For example, in [3], the main bearing fault is diagnosed using support vector machine (SVM) classifiers; and in [4], simultaneous multiple faults are again diagnosed using SVM. Despite the promising performance of supervised algorithms in research, their deployment in a real application requires domain adaptation techniques that provide the ability to train a model on one dataset (source) for which label is available and secure a good performance on another dataset (target) whose label is not available. Otherwise, it is nearly impossible to apply supervised approaches, as deriving labeled data sets from WT operational data is often difficult (because of the lack of standardization in maintenance records), time-consuming, and error-prone, resulting in a very imbalanced data set. Furthermore, domain adaptation techniques are crucial to deploy supervised approaches to wind parks where the defect has not already occurred. It is noteworthy that there is a consistent portion of the literature that fosters the use of domain adaptation techniques to deal with supervised learning for WT fault diagnosis, as in [5] where a WT gearbox and cross-bearing faults are studied through simulated data, and [6] where a bearing fault is studied through a test bench. Second, as indicated in the review study [7], a substantial number of sources confirm the conclusions using simulated SCADA data (as in [4]) or experimental data (from a test bench) as in [3]. Although this is understandable given that real SCADA data sets are frequently proprietary and not readily available to the scientific community, it is a significant disadvantage because data provided by test rigs or mathematical models may not generalize well to real-world settings [7]. Third, as indicated in the highly referenced study [8], where a detailed analysis regarding utilizing real SCADA data for WT CM is given, the bulk of the references analyzed based their conclusions on a very limited quantity of data, typically just 1 to 4 WTs. Again, this is a significant disadvantage, as it is unclear whether these tactics would scale successfully throughout the whole wind park. Forth, certain references, such as [9], offer tactics that result in a significant number of false alerts, making the contribution inconvenient in the real world, as it would result in alarm fatigue for operators. Fifth, a significant number of studies, such as [10] and [3], detect the fault with less than a week's notice, rendering them useless in a real-world situation where the plant operator needs at least months to program repair to concur with the availability of replacement parts as well as weather or production windows to minimize turbine downtime. Finally, several notable references employ completely unsupervised techniques that have been tested on real wind parks. For instance, in [11], the pitch system CM is stated using isolation forest and validated on ten WTs.

However, with advances in the field of deep learning, progressively, much of the research is focused on capturing relevant features using neural networks with deep hidden layers. Methodologies based on artificial neural networks (ANN) have been proposed, as in [12] and [13], as well as different variations of autoencoders (based on ANNs) have been widely studied. For example, in [14] a gearbox failure detection method is proposed based on a deep joint variational autoencoder. In [15] a multi-level-denoising autoencoder approach is stated and faults at the pitch system and drive train (vibration anomaly) are detected. In [16] a denoising autoencoder with temporal information is proposed to detect anomalies in the generator speed and gearbox filter. Note that in all aforementioned references time-series data is widely adopted, where long-term dependency is essential to form the classifiable features. Because SCADA data is a time series, temporal data is essential for constructing a prognosis model. Furthermore, understanding the changes and trends in variables over time is crucial to constructing the model architecture. That is, the model learns how previous data samples impact future data samples, which is the primary purpose of designing an early defect detection approach. Most basic designs, such as ANNs, do not take a prior data point into account when deciding the next; instead, the model learns from individual samples.

To better address that the traditional ANNs either rely on expert knowledge and handcrafted features or do not fully model long-term dependencies hidden in time-domain signals, recent works focus on the adoption of recurrent neural networks (RNNs). In this case, the recurrent neural network (RNN) model may be useful since it has a recurrent connection and can learn the influence of past and current inputs while forecasting the outcome. On the other hand, RNNs suffer from the problem of vanishing gradients that the LSTM and GRU models solve by employing gates to determine which information should be kept and which should be ignored. GRU networks have a few benefits over LSTM [17] such as a smaller number of parameters and a reduced computational cost, which are relevant in this application, which requires training with data over several years. In this work, the GRU neural network is selected because of its ability to forecast information from time series data (taking into account past information) and its simplicity (associated with a lower computational cost) compared to other types of RNNs. Finally, there are some relevant works related to WT fault diagnosis with RNNs. For example, in [18], fault diagnosis of WT based on long short-term memory (LSTM) networks (a particular type of RNN) is proposed. However, not only SCADA but also high-sampling vibration data is used

and validation is done only with simulated (not real) data. It is noteworthy the work in [19], where the main bearing failure (the fault of interest in the present work) is predicted based on LSTM with more than 90 days on average, but again not only SCADA data is used but also high-sampling vibration data that greatly helps to the obtained performance. Finally, the work in [20] detects the gearbox failure by means of GRU networks, but the used SCADA data is 1-minute averaged instead of the standard 10-minute averaged, thus giving extra predicting capabilities.

All that being said, the novelty of the proposed design is the following. A novel methodology to detect months in advance the main bearing failure (when the event that initiates/develops the failure releases heat) using only standard 10-minute SCADA data and based on a GRU (because of its ability to forecast information from time series data and its low computational cost in comparison to other RNNs) and a novel fault prognosis indicator (FPI) that precludes the problem of alarm fatigue. Furthermore, the stated methodology is validated with real (as opposed to simulated or experimental which greatly simplifies the problem) data from an in production wind farm. The methodology main features are the following: 1) Semi-supervised and based only on healthy data (precluding the problem of imbalanced data sets), thus expanding its range of application to any wind park (although no faults have yet occurred). 2) Reliable predictions with minimum false alarms. 3) Early warning months in advance, providing the plant operator time to program maintenance to match with replacement part availability, as well as weather or production windows to minimize turbine downtime. 4) Robust to seasonality and operating and environmental conditions. 5) Installing extra costly sensors is not needed, as only SCADA data is used. Therefore, the method can be applied to WTs already in operation for life-time extension services (this is relevant, as it is expected that 38 GW of wind parks in Europe will reach their life expectancy in the next five years). 6) Validated on real SCADA data from a wind park composed by nine WTs in production.

The following is how the rest of the article is structured. Section II provides a brief description of the wind park. Following this, the types of main bearing faults are introduced in Section III. The available SCADA data and work order records are then presented in Section IV. Section V discusses the suggested methodology. The findings and discussion in Section VI are used to analyze and convey the performance of the stated strategy. Finally, conclusions are drawn and recommendations for future works presented in Section VII.

II. Wind park

The wind park is composed of WTs with a diameter of 101 m and 2300 kW of nominal power. Figure 1 displays the main elements of the WT. The energy production starts at an initial wind speed of 3 m/s and reaches its rated power at 12 m/s. Finally, when the wind speed is 20 m/s or more, the WT automatically stops using its braking system. In Table I, the technical specifications of these turbines are summarized.

Notably, the main bearing employed by these WTs is a double-spherical roller type, which are appropriate for applications involving very low speeds (up to 25 rpm) and high loads with varying directions; therefore, they are a reliable choice for the main bearing of a WT. Other characteristics of this type are robustness to withstanding loads of variable direction and low friction, which implies a low energy loss and a longer lifespan.

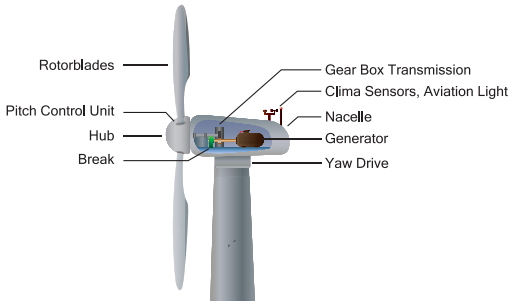

Fig. 1. Wind turbine parts.

TABLE I
Wind turbines' technical specifications.

Technical specification	Value
Number of blades	3
Nominal power	2300 kW
Rotor diameter	101 m
Cut-in wind speed	3 m/s
Rated wind speed	12 m/s
Cut-out wind speed	20 m/s

III. Main Bearing Faults

Bearings are supports or guides for machine elements such as shafts that need rotation or oscillation and can handle axial loads. Many sectors use them, including transportation, medical (surgical instruments, as well as diagnostic and laboratory equipment), and energy (wind turbines, and solar panels), among others. Typically, a bearing comprises four elements: cage, inner race, outer race, and rolling. During operations, each of these components is subjected to mechanical stress by at least one of the following forces: frictional, impact, centrifugal, and inertial. Therefore, any of them can suffer a breakdown.

The main bearing, which supports the main shaft of a WT, is a large component and this section attempts to offer an insight on the many ways of bearing failure as well as the enormous number of existing patterns, which make forecasting a bearing failure in advance difficult. Following the ISO 15243 standard, the SKF company classified a bearing's different failure modes as follows [21]: fatigue, wear, corrosion, electrical erosion, plastic deformation, and fracture and cracking. Each of these modes has different sources, behavior, and bearing damage. It is beyond the scope of this paper to elaborately explain these failure scenarios. However, a brief overview of some of the aforementioned failure modes is provided in the next paragraph to promote an understanding of the need for considering temperature variations while detecting a range of bearing defects.

First, fatigue is one of the most common failure modes in bearings. Subsurface-initiated fatigue and surface-initiated

fatigue are the two types of fatigue. Both submodes have an accumulation of residual stresses, causing the material to transition from a randomly oriented grain structure to fracture planes. This results in microcracks and beginning flaking, which emit heat. Second, the loss of a bearing surface is referred to as wear, and frictional heat is frequently present. Third, corrosion occurs due to the entry of moisture, water and aggressive contaminating liquids into the bearing. Corrosion leads to premature and extended spalling, as the material undergoes structural change and the load zone surfaces are reduced such that overloading occurs. Thus, the release of heat due to these three types of failure can be detected using temperature variables in the fault prediction model. Fourth, electrical erosion occurs because the electrical current flowing through the rings via rolling elements causes damage. The erosion can be divided into two types: excessive current erosion and current leakage erosion (produced by low-intensity current). Excessive current erosion heats up the material to temperatures between tempering and melting levels, forming craters in the rolling element and deforming the raceways. Again, the release of heat is the first symptom of this type of failure. Lastly, the bearing can fracture and crack, where bearing cracking can be caused by thermal factors. When two surfaces slide past each other, frictional heat is generated. If the sliding is considerable, the heat can crack in the perpendicular direction of the slip. Again, heat release is an initial starting point that can reveal the developing fault.

In summary, all bearing failures have a starting point from which an anomalous behavior begins until the end point, which is when the bearing eventually fails. Symptoms of anomalous behavior include a change in the main bearing temperature, among others. Heat release is a widespread symptom, so this work aims to predict the fault by detecting this first event. Thus, an early warning can be triggered, notifying that within a certain number of months, that element will seriously fail.

IV. SCADA Data

In this study, the SCADA data is obtained from nine in-production WTs. The available data spans from January 1, 2015, up to December 31, 2018, and shows the continuous operation of each WT every 10 minutes. The SCADA data contain a variety of measurements that may be divided into five categories: environmental (see Table II), electrical, component temperature (see Table III), hydraulic, and control variables. The data is obtained with a 1 Hz sampling frequency and stored on a 10-minute average. The mean, standard deviation, maximum and minimum values are available for each sensor. This work focuses only on the mean values; therefore, the description of the variables in Tables II-III refers to its mean value. Furthermore, only some exogenous variables (environmental) and some temperature variables are employed by the proposed strategy; thus, in this section, only these two sets of variables are explained in depth.

Table II displays various environmental sensors from the SCADA data. In this, the ambient temperature and the wind speed can be found. The first one affects the temperature of all subsystems that change significantly between summer and winter. Finally, given its direct impact on the machine's operation, wind speed is the most crucial exogenous variable associated with a WT.

TABLE II
Environmental variables.

Variable	Description	Units
wtc_AmbieTmp_mean	Ambient temperature	°C
wtc_PrWindSp_mean	Primary wind speed	m/s
wtc_AcWindSp_mean	Actual wind speed	m/s
wtc_PriAnemo_mean	Primary wind speed of anemometer	m/s

TABLE III
Component temperature variables.

Variable	Description	Units
wtc_A1IntTmp_mean	Internal temperature A1	°C
wtc_BrkTmpGn_mean	Brake generator temperature	°C
wtc_GenBeRTm_mean	Generator bearing temperature	°C
wtc_GeOilTmp_mean	Gearbox oil temperature	°C
wtc_HSGenTmp_mean	HS generator temperature	°C
wtc_HSRotTmp_mean	HS rotator temperature	°C
wtc_HubTemp_mean	Hub temperature	°C
wtc_NacelTmp_mean	Nacelle temperature	°C
wtc_MainBTmp_mean	Main bearing temperature	°C
wtc_HydOilTm_mean	Oil temperature for hydraulic system	°C

Table III shows some variables related to temperature. The main bearing defect is the subject of this research. As a result, it is critical to monitor the temperatures of nearby components, such as the main bearing temperature itself, the gearbox oil temperature, and the generator bearing temperature.

Finally, in addition to the SCADA data, information is available on maintenance and repair actions (work orders) for the different WTs. These data provide the following information: failure type, failure date, work order date, affected subsystems, and actions taken. This information is used in this work to model and test whether the proposed methodology can predict the appearance of main bearing faults months in advance.

V. Fault Detection Methodology

This section comprehensively describes the stated fault detection strategy. First, the data preprocess, crucial when dealing with real data, is described in Section V-A. Next, the data split is described in Section V-B. This section emphasizes the GRU model strategy and the way to deal with seasonality. Then, in Section V-C, the importance of data normalization is stated, as well as the used normalization technique. Next, in Sections V-D and V-E, the normal behavior model (NBM) -WT's normal/healthy behavior- is constructed based on a GRU neural network, and the GRU architecture and the hyperparameter set-up are comprehensively explained. When healthy data is supplied, the GRU model is trained to predict the main bearing temperature in a certain time step from the input variables, functioning as a virtual sensor. Then, the proposed fault prognosis indicator (FPI) is stated in Section V-F, which employs a moving average strategy to diminish the number of false alerts, avoiding alarm fatigue to the wind park operator.

The proposed methodology can detect any failure associated to the main bearing that initiates or develops with

TABLE IV
SELECTED SCADA VARIABLES.

Variable	Description	Units
wtc_MainBTmp_mean	Mean main bearing temperature	°C
wtc_GenBeRTm_mean	Mean generator bearing temperature	°C
wtc_GeOilTmp_mean	Mean gearbox oil temperature	°C
wtc_PrWindSp_mean	Mean primary wind speed	m/s

an associated heat release. That is because the conceived methodology relies on SCADA data associated to temperature variables to detect the failure. Note that the low sampling rate of SCADA data (10-minute average) hinders information in variables with a fast dynamic (e.g., vibrations), however as temperature variables have a slow dynamic their SCADA data still contains relevant information. Finally, as indicated in Section III, the vast majority of different main bearing failure modes are associated to heat release, thus allowing the proposed methodology to be widely applicable.

A. Data Preprocess

In Section III, the different types of main bearing failure are described, noting that a wide variety of failure modes induce an increase in temperature when the fault initiates. Consequently, in this work, the temperatures of the components near the main bearing are employed in conjunction with the ambient temperature to preclude data seasonality. The selected variables are shown in Table IV, namely, the main bearing temperature, the generator bearing temperature, the gearbox oil temperature, and the wind speed. It is noteworthy that the ambient temperature is subtracted from all variables related to temperature to avoid the problem of seasonality.

Real data is noisy; thus, a data cleaning strategy is required. Particularly, in this work, a data imputation strategy is used for missing data. Techniques based on statistical variables such as mean, median, and mode are not considered because they could generate a bias in the data's mean and deviation. [22]. Alternatively, the use of the piecewise cubic Hermite polynomial interpolation is proposed [12], as it contributes a continuous first derivative function and maintains the shape and monotonicity. Missing data at the beginning and end of the data set are filled in with their closest value.

B. Data Split: Train, Validation, and Test

To build a deep learning model, the following steps are taken: i) dividing the data into training, validation, and testing data sets; (ii) training and validating the model using the train and validation data sets, respectively; and (iii) evaluating the obtained model on the test set to investigate the performance of the model predictions when using the new data (that the model has never seen before).

The approach used for data splitting is crucial in developing a machine or deep learning model, as it has a major impact on the overall model. The following assumptions are made to separate the data in this work. To begin with, only healthy data is utilized to train and validate the model because the major goal is to build WT's NBM. Additionally, data from the WT's entire region of operation (range 1: when the wind speed is lower than the cut-in wind speed; range 2, when the wind speed is between the cut-in and rated wind speed and; range 3, when the wind speed is above the rated wind speed) is used to train and validate the model. Thus, the model generated is robust to the WT's varied operating and environmental conditions. Finally, to make the model resilient to seasonality, the training and validation data must span at least one full year. In summary, the data are split in this manner. Data from January 2015 to September 2017 (144576 samples) are used for training, data from October 2017 to December 2017 (13248 samples) are used for validation, and data from January 2018 to December 2018 (52560 samples) are used for testing.

C. Data Normalization

The data for the selected variables originate from several sources; hence, their order of magnitude differs. Scaling the data is widely advised by machine/deep learning approaches to achieve higher performance of the employed optimization algorithm. Based on SCADA data's nature for the specific wind park under study, these data have no significant outliers. Thus, the min-max scaling is directly selected (since there is no problem with outliers' sensibility that this kind of scaling can suffer) that scales data in the [0,1] range using a linear transformation from the initial set's range. Thus, this technique delivers data that are normalized exactly to the same interval for all used variables.

D. Brief introduction to GRU neural networks

The deep learning model used in this work is a GRU neural network. Although this paper does not intend to provide a detailed understanding of GRU networks, they are briefly explained below to introduce the used notation and render this paper self-contained and understandable.

Cho et al. [23] proposed GRUs to allow relationships in an adaptable manner across temporal scales [24]. The GRU workflow resembles that of a simple recurrent neural network (RNN). It processes the input vector x_t and the hidden state h_{t-1} from the previous timestamp through the gates at each timestamp t. It then produces a new hidden state h_t, which contains both short- and long-term states and is transmitted to the next timestamp. Regarding the manner in which the flow of information is regulated by gating units, the GRU is comparable to a long short-term memory (LSTM) unit and is often thought of as a simpler version of LSTM. The amount of gating units and the state vector are the main distinctions between them. The GRU only has two gating signals (update and reset), whereas LSTM has three (input, forget, and output). Therefore, the GRU has fewer parameters and lower computational cost. Note that the GRU also combines both short-term and long-term state vectors into a single vector; see [24], [25].

In Figure 2, the gated GRU architecture is shown. The input and output vectors are x_t and y_t, the reset and update gates are r_t and z_t, and the activation and candidate activation are h_t and $\tilde{h}_t$, respectively. The following are the definitions of the two gates:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \tag{1}$$

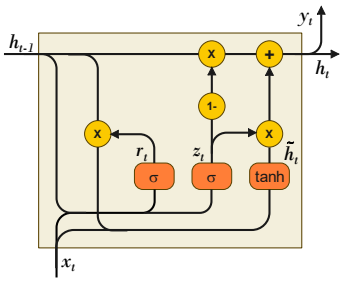

Fig. 2. Architecture of GRU cell.

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r). \tag{2}$$

The update gate, z_t, controls which bits of the long-term state should be inserted and which parts should be removed. The reset gate, r_t, determines which part of the prior state is displayed to the main layer or candidate activation, ($\tilde{h}_t$). Then, the GRU model can be formulated as follows:

$$\tilde{h}_t = \tanh\left(W_h x_t + U_h(r_t \cdot h_{t-1}) + b_h\right), \tag{3}$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t, \tag{4}$$

where the candidate activation, $\tilde{h}_t$, analyzes the input vector x_t and the previous short-term h_{t-1}, and only the most relevant parts are stored in the vector h_t. On the other hand, the activation, h_t, represents the output state of the model and, as stated before, it contains the short-term and long-term state. From the above equations,

- W_z, W_r, W_h are the weight matrices for each of the three layers in relation to the input vector x.
- U_z, U_r, U_h are the weight matrices for each of the three layers in relation to the prior short-term state, h_{t-1}.
- b_z, b_r, b_h are the bias terms for each of the three layers.

The GRU network architecture utilizes the four variables displayed in Table IV. The mean main bearing temperature is the output of the GRU model at time t, and the inputs are the mean generator bearing temperature, the mean gearbox oil temperature, and the mean primary wind speed. Thus, there are three inputs to the network and one output, i.e., the network is considered as a many-to-one structure. The inputs are given as a sequence of 144 consecutive time steps, that is, data that comprise 24 hours (recall samples are given each 10 minutes).

The GRU architecture and selected hyperparameters are comprehensively explained in the next section.

E. GRU proposed architecture

The GRU architecture requires a set of different hyperparameters for its optimal performance according to the problem to be solved. This section describes the selected hyperparameters in detail.

First, the number of hidden layers, that is, the layers between the input and output of the network, which contain the GRU cells, is stated. In this work, it is defined as one single hidden layer, as testing with two layers did not improve performance in relation to early fault detection and thus did not justify the additional computational cost.

Second, the hidden size determines the number of features in the hidden state. In other words, it defines the number of neurons or GRU cells in the hidden layer. This hyperparameter determines the learning power of the model. In this paper, 128 neurons are selected in the hidden state due to the amount of samples that the model has to process.

Third, the batch size is defined in 128 samples, where each sample contains 144-time steps. An initial learning rate of 0.001 is defined, and 50 epochs is used to train the network.

Finally, a loss function must be defined. In this work, the mean squared error (MSE) is employed, as large errors are important (may represent a fault) with respect to small errors (maybe due to model error). To sum up, Table V details the hyperparameter values for the proposed GRU architecture.

In a nutshell, the diagram shown in Figure 3 summarizes the stated approach from the initial data preprocess, through data imputation, data normalization, and, finally, the training stage of the GRU model.

TABLE V
SETUP OF GRU HYPERPARAMETERS.

Hyperparameter	Value
Number of hidden layers	1
Number of neurons in the hidden state	128
Batch size	128
Epoch size	50
Initial learning rate	0.001
Loss function	MSE

F. Fault Prognosis Indicator (FPI)

The FPIs are typically defined using a residual and establishing a decision threshold. When the residual of a sample exceeds the threshold, an alert is activated. In this work, the natural residual to be employed would be the square of the difference between the real SCADA main bearing temperature, T, and that predicted by the GRU network, $\hat{T}$, i.e., $(T - \hat{T})^2$. However, when using this residual straightforward to define a threshold, an overwhelming number of false alarms (false positives) would be triggered, leading to alarm fatigue in wind turbine operators and rendering the method useless.

To prevent this issue, the long persistence in time of the residual over the threshold has to be monitored, to discriminate false alarms from real fault detection alarms. Thus, in this section, to accomplish this aim, an FPI is stated, which filters the initial obtained residual with a moving average (MA) to smooth the original spiky residual.

Remember that a MA of k observations smooths a time series by computing the average of the k most recent observations; as a result, each new entering observation drives the oldest in the group out of the computation [26]. According to [27], the MA may generate cyclic and trend-like displays even when the source data are separate random occurrences with a fixed mean. As a result, this property limits its use as a control mechanism, and the exponentially weighted moving average (EWMA) emerges. This has the property of allocating less weight to data as it ages. A point on an EWMA chart

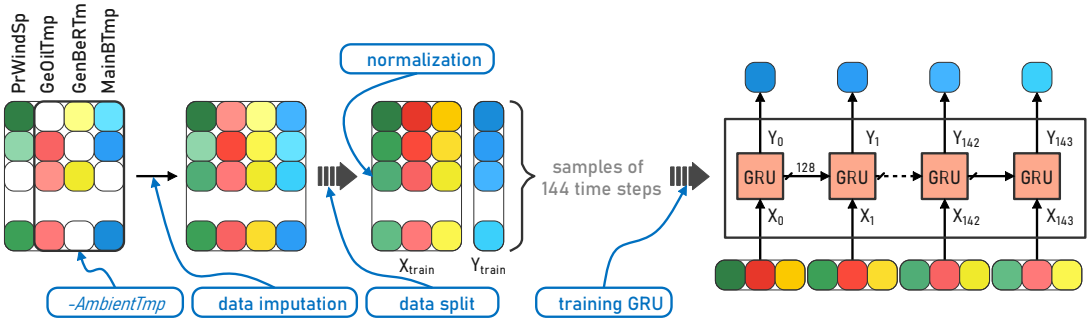

Fig. 3. Flowchart for the first phase of the proposed methodology until the training of the GRU model.

can be assigned a long or short memory. The EWMA is the best depicted one-time position ahead of the most recent observation; consequently, this statistic may be used to forecast the future observation. In this work, the EWMA is selected to be applied to the prediction errors as follows,

$$\text{EWMA} = \hat{T}_{t+1} = \hat{T}_t + \lambda e_t = \hat{T}_t + \lambda(T_t - \hat{T}_t), \quad (5)$$

where $\hat{T}_{t+1}$ is the predicted value at time $t+1$. Similarly, $\hat{T}_t$ is the predicted value at time t, T_t is the measured real SCADA value at time t, and $e_t = T_t - \hat{T}_t$ is the prediction error at time t. Finally, λ is a parameter ($0 < \lambda < 1$) that determines the memory depth of the EWMA. This parameter is empirically selected using its relation with the span ($s \geq 1$) [28]. Equation 6 shows this relation:

$$\lambda = \frac{2}{s+1}. \quad (6)$$

Finally, after averaging the initial residuals with the EWMA, a threshold is defined using the training and validation data. In this study, the threshold is the limit where a residual would be considered within normal behavior, so the mean μ_h and standard deviation σ_h (this must not be confused with the sigmoid function) of the EWMA residuals of training and validation data must be taken into account. It is important to emphasize that, since the aim is to model the WT's normal behavior, the μ_h and σ_h are obtained only from healthy data (from training and validation). Finally, the threshold is designated:

$$\text{threshold} = \mu_h + \kappa\sigma_h, \quad (7)$$

where κ is a constant which establishes the threshold value. In this work, two values of κ are used, as one defines a warning threshold and the other a fault alarm threshold. To sum up, Figure 4 shows the second phase of the proposed methodology, which details the computation of the warning and alarm thresholds after the EWMA filtering of the prediction error.

In the next section, an analysis is conducted on how the selection of the span number, s, and the κ values to determine the thresholds (warning and alarm) affects the proposed methodology.

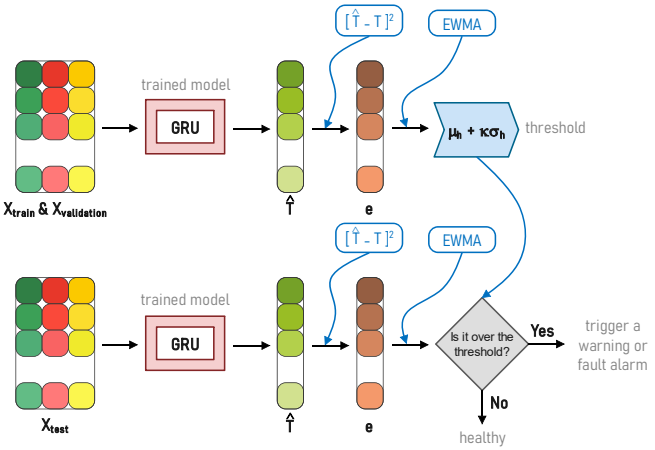

Fig. 4. Flowchart for the second phase of the proposed methodology, considering the computation of the threshold.

VI. Results and Discussion

This section provides and examines the outcomes of the suggested fault prediction approach for a real-world wind park in operation.

Figure 5 shows the GRU estimations for two different WTs over the train and test data sets. Remember that the test data set contains the data of the year 2018. Figures 5 (a) and (b) display the estimated value $\hat{T}$ and the SCADA target of WT1, a healthy WT. The estimation is close to the real SCADA value in the two figures, and very few samples have disparate values. The performance of the GRU for this WT is shown in Figures 5 (c) and (d), where the absolute error between SCADA and prediction is graphed for the training and test sets. In contrast, Figures 5 (e) and (f) present the GRU estimated value and SCADA value for WT2, a WT that suffered a main bearing fault on June 11, 2018. The prediction over the training set is close to the target, contrary to the prediction over the test set. Particularly, in the test set, the prediction does not fit well for months before the fault occurs, and after it (when the fault is corrected), this error decreases. Figures 5 (g) and (h) show the residual on the training and test data sets, confirming the good performance over the training set, while in the test set, the residual increases a month before the fault and decreases

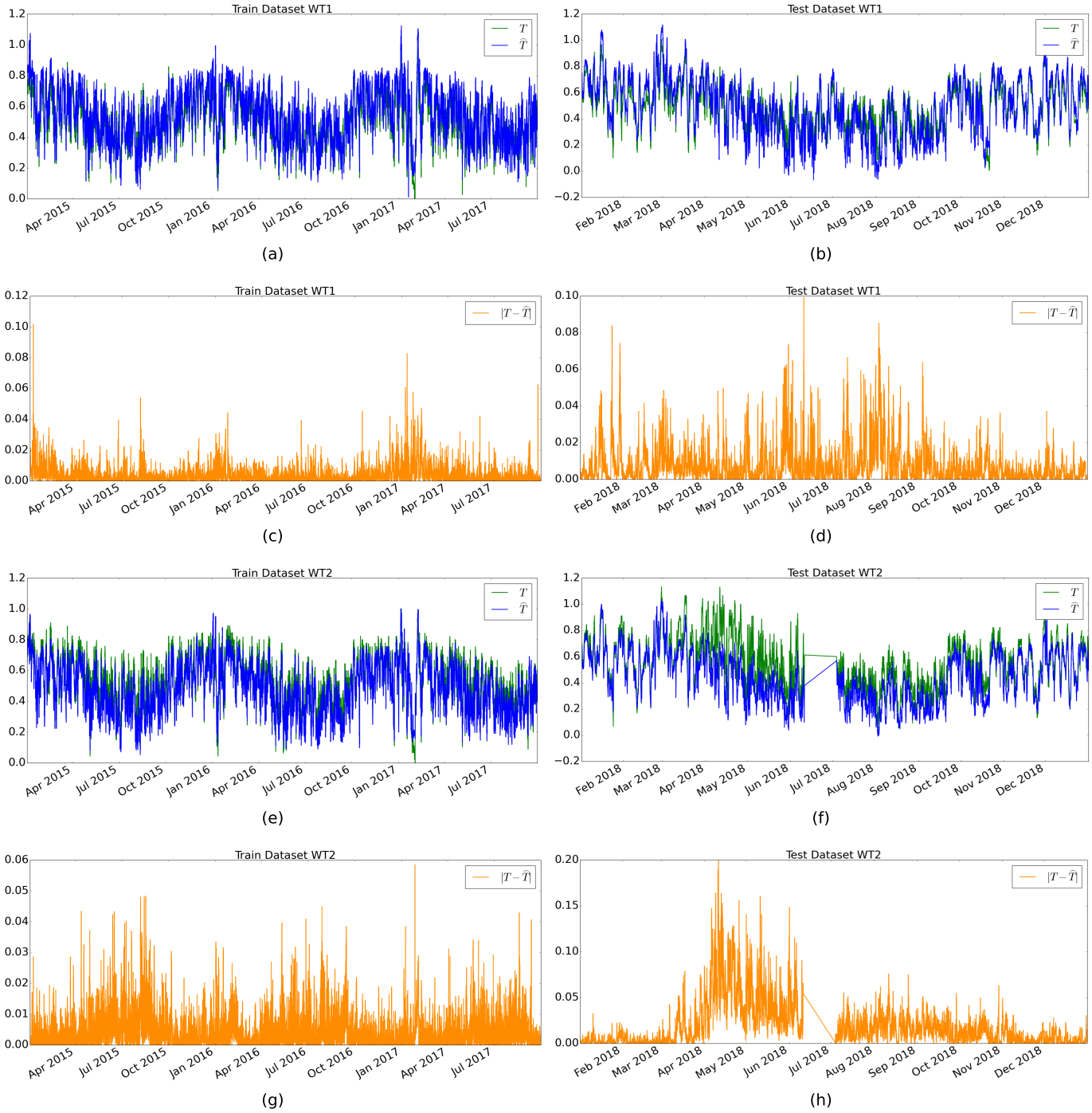

Fig. 5. (a) GRU estimated $\widehat{T}$, and SCADA real T, for WT1 over the train set. (b) GRU estimated $\widehat{T}$, and SCADA real T, for WT1 over the test set. (c) Error absolute value, $|T-\widehat{T}|$, for WT1 over the train set. (d) Error absolute value, $|T-\widehat{T}|$, for WT1 over the test set. (e) GRU estimated $\widehat{T}$, and SCADA target value T, for WT2 over the train set. (f) GRU estimated $\widehat{T}$, and SCADA target T, for WT2 over the test set. (g) Error absolute value, $|T-\widehat{T}|$, for WT2 over the train set. (h) Error absolute value, $|T-\widehat{T}|$, for WT2 over the test set.

after it. As comprehensively explained in Section V-F, the EWMA is applied over the prediction errors to measure the behavior and trend of the data as a function of persistence over time and, thus, to diminish the number of false alerts. The historical data in this study corresponds to samples taken every 10 minutes during a year, and even though temperature is a slow-changing-rate variable by nature, the prediction error must be filtered with an EWMA to obtain data with better defined patterns and behaviors. As shown in Equation 5, the depth of memory must be selected and, as stated in Equation 6, that parameter is established as a function of the span, s. For this study, two spans are considered and analyzed: s =144 (one day), and s =1008 (one week). These values are chosen to measure the persistence of the data trend (successive peaks, monotony) through time. For instance, if the EWMA is weekly based, perhaps the data trend is more visible and smoother

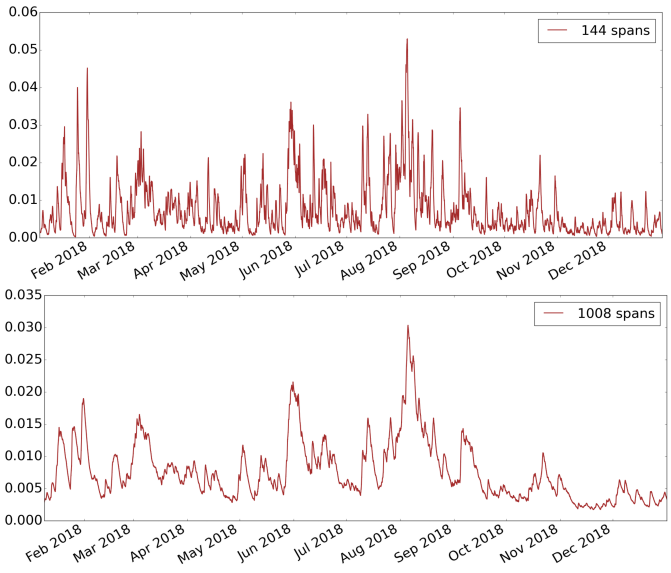

Fig. 6. EWMA filtered prediction errors for a WT, considering the span values $s = 144$, and $s = 1008$.

than if it is daily based. It helps to avoid the noise caused by isolated hourly peaks, which is very common in real SCADA data. Actually, this selection is influenced by the findings of McKinnon et al. [29]. Their research, on the influence of time history on WT failures using SCADA data, tests three distinct moving windows (MWs): daily, weekly, and monthly. In comparison to the others, the weekly MW has the best performance in identifying failures. On the one hand, a daily MW contains too much noise, leading to a large percentage of false alarms. On the other hand, a monthly MW removes much information and does not allow any specification of when an anomaly occurred.

Figure 6 shows the EWMA computation considering the established spans. After the EWMA computation, the definition of the thresholds is required, as stated in Section V. The aim is to propose two thresholds: one used as a warning and another to trigger a fault alarm. Table VI shows the detailed false-positive alarms over the training and validation dataset for each WT, considering the two proposed spans' values and two values for κ: 12 and 15. Thus, note that the selected values for κ are just based on the observation of the training and validation dataset where the WTs are healthy. The value of κ is set to minimize the number of false alarms over these datasets. Therefore, there is no information from the test set (or from the knowledge of the occurred fault on the test set) used to decide the κ values. The results demonstrate that using 1008 spans no false-positive alarms are reached over the training and validation datasets.

Finally, Figure 7 shows the results on the test dataset for the entire wind park, obtained with the proposed FPI. A warning is triggered when the prediction error crosses $\mu_h + 12\sigma_h$ (green dotted line), while a fault alarm is triggered when the error exceeds $\mu_h + 15\sigma_h$ (red dotted line). Recall that the aforementioned mean and standard deviation are computed over the filtered prediction errors from train and validation. For WT2, both the warning and fault alarm were triggered

TABLE VI
FALSE-POSITIVE ALARMS (X-MARK) OVER THE TRAINING AND VALIDATION DATASETS.

WT ID	144 spans		1008 spans	
	$\mu_h + 12\sigma_h$	$\mu_h + 15\sigma_h$	$\mu_h + 12\sigma_h$	$\mu_h + 15\sigma_h$
WT1				
WT2				
WT3	X			
WT4	X			
WT5	X	X		
WT6				
WT7	X			
WT8	X			
WT9	X	X		

in the first week of June 2018. Considering that for WT2, the main bearing fault occurred on June 11, 2018, the early detection through this methodology is accomplished roughly two months in advance. Additionally, note that there is a clear trend of residual's increasing and then a decreasing. When bearing failure initiates (or develops) there is usually a brief heat release rendered as temperature increasing. As it is stated in [21], almost all bearing failure modes (excessive current, fatigue fracture, thermal cracking, etc.) are driven by unforeseen heat release. After that, the temperature goes back to normal i.e. crack is not growing. Thus, the methodology's approach is to predict this typical heat release in advance, before the bearing is entirely damaged. The case for WT3 and WT5 is interesting, as both a warning and a fault alarm were triggered in a week, but this WTs are healthy during 2018. Thus, they represent a false alarm. However, note that the alarm has a very short period length (less than a week), in both cases, in contrast to WT2 where the alarm is activated for two months. Therefore, in short, the model successfully detects the expected fault onset on the WT that contains the main bearing fault. As it has been before-explained, the onset defines the start point of the component's degradation through time. Thus, it is crucial that despite the residuals go back under the threshold, the triggered alarm must be kept active. It allows maintenance operators to plan ahead on-site revisions and give all needed attention to these component and perform required actions to extend its lifespan and not compromise WTs uptime.

VII. CONCLUSIONS

In this work, an early fault detection approach for main bearing failures in WTs is devised and verified using a GRU neural network and just SCADA data. The model is built entirely from healthy data and is robust to all operational and environmental variations. The approach has been tested at a wind park with nine WTs. The findings show that the system produces minimal false alarms and that the defect of concern is predicted months in advance. Unfortunately, there is just one major bearing failure in the investigated wind park data, which is insufficient for statistical analysis. To investigate more extensively and draw conclusions such as a predicted time and confidence level, it is necessary to apply the model to other cases with this issue. Finally, it is convenient to study and test other sequential learning models that have been developed

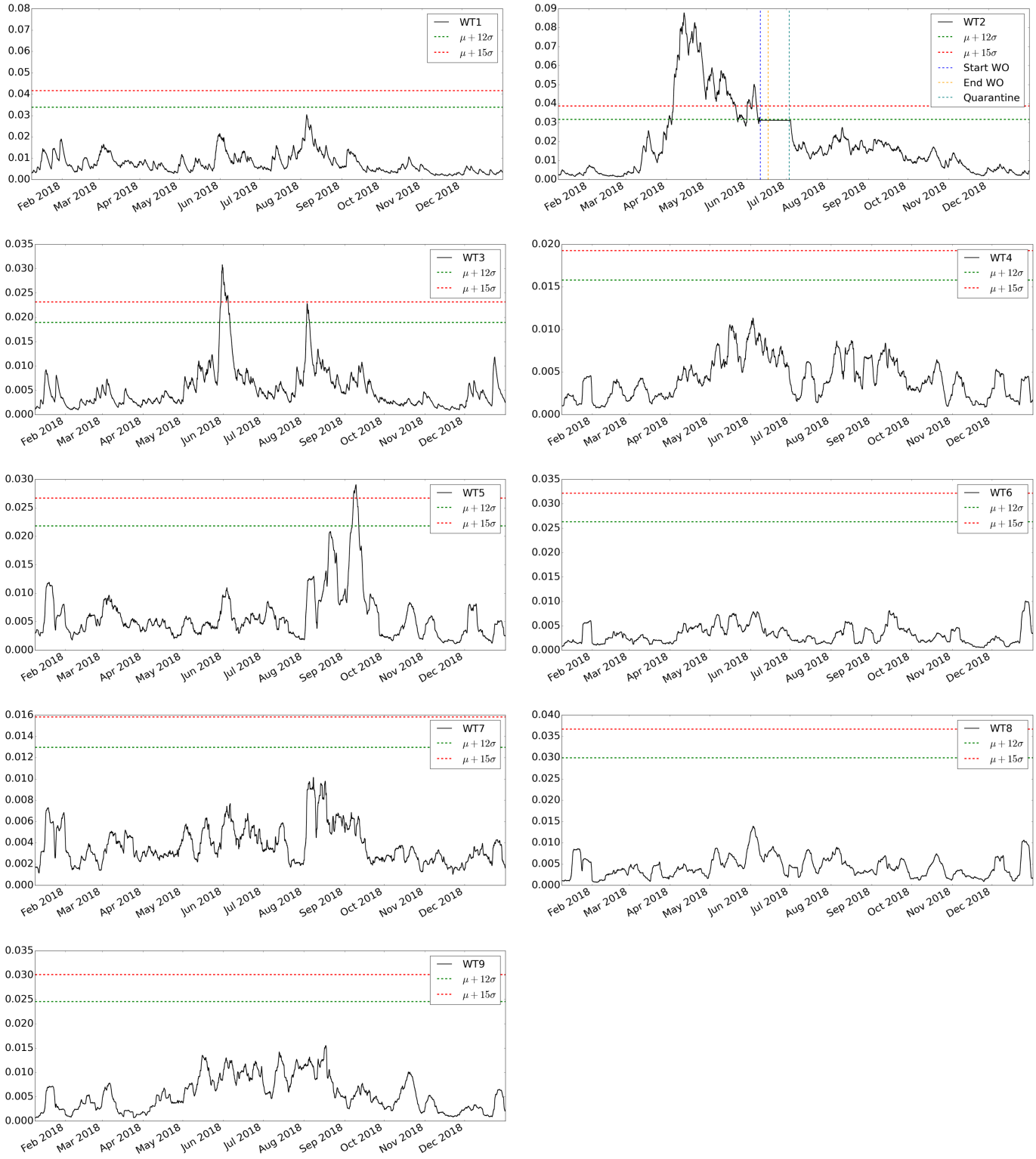

Fig. 7. EWMA on residual errors for the WT's test dataset (using 1008 spans), where the green line represents a fault warning and the red line indicates a definite fault.

in recent years. Specifically, time convolution neural networks and transformer networks, both of which have shown to be very powerful for some specific problems.

REFERENCES

[1] WindEurope, "Getting fit for 55 and set for 2050, electrifying europe with wind energy," 2021.
[2] IRENA, "Renewable power generation costs in 2020," 2021.
[3] H. Zhao, Y. Gao, H. Liu, and L. Li, "Fault diagnosis of wind turbine bearing based on stochastic subspace identification and multi-kernel

support vector machine," *Journal of Modern Power Systems and Clean Energy*, vol. 7, no. 2, pp. 350–356, 2019.
[4] Y. Fu, Z. Gao, A. Zhang, and X. Liu, "Fault classification for wind turbine benchmark model based on hilbert-huang transformation and support vector machine strategies," in *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*. IEEE, 2021, pp. 1–8.
[5] Y. Qin, Q. Qian, J. Luo, and H. Pu, "Deep joint distribution alignment: A novel enhanced-domain adaptation mechanism for fault transfer diagnosis," *IEEE Transactions on Cybernetics*, 2022.
[6] Z.-H. Liu, B.-L. Lu, H.-L. Wei, X.-H. Li, and L. Chen, "Fault diagnosis for electromechanical drivetrains using a joint distribution optimal deep domain adaptation approach," *IEEE Sensors Journal*, vol. 19, no. 24, pp. 12 261–12 270, 2019.
[7] A. Stetco, F. Dinmohammadi, X. Zhao, V. Robu, D. Flynn, M. Barnes, J. Keane, and G. Nenadic, "Machine learning methods for wind turbine condition monitoring: A review," *Renewable energy*, vol. 133, pp. 620–635, 2019.
[8] J. Tautz-Weinert and S. J. Watson, "Using scada data for wind turbine condition monitoring–a review," *IET Renewable Power Generation*, vol. 11, no. 4, pp. 382–394, 2017.
[9] J. L. Godwin and P. Matthews, "Classification and detection of wind turbine pitch faults through scada data analysis," *IJPHM Special Issue on Wind Turbine PHM*, p. 90, 2013.
[10] K. Leahy, R. L. Hu, I. C. Konstantakopoulos, C. J. Spanos, A. M. Agogino, and D. T. O'Sullivan, "Diagnosing and predictingwind turbine faults from scada data using support vector machines," *International Journal of Prognostics and Health Management*, vol. 9, no. 1, 2018.
[11] C. McKinnon, J. Carroll, A. McDonald, S. Koukoura, and C. Plumley, "Investigation of isolation forest for wind turbine pitch system condition monitoring using scada data," *Energies*, vol. 14, no. 20, p. 6601, 2021.
[12] Á. Encalada-Dávila, B. Puruncajas, C. Tutivén, and Y. Vidal, "Wind turbine main bearing fault prognosis based solely on scada data," *Sensors*, vol. 21, no. 6, p. 2228, 2021.
[13] Z. Zhang *et al.*, "Automatic fault prediction of wind turbine main bearing based on scada data and artificial neural network," *Open Journal of Applied Sciences*, vol. 8, no. 06, p. 211, 2018.
[14] L. Yang and Z. Zhang, "Wind turbine gearbox failure detection based on scada data: A deep learning-based approach," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2020.
[15] X. Wu, G. Jiang, X. Wang, P. Xie, and X. Li, "A multi-level-denoising autoencoder approach for wind turbine fault detection," *Ieee Access*, vol. 7, pp. 59 376–59 387, 2019.
[16] G. Jiang, P. Xie, H. He, and J. Yan, "Wind turbine fault detection using a denoising autoencoder with temporal information," *IEEE/Asme transactions on mechatronics*, vol. 23, no. 1, pp. 89–100, 2017.
[17] A. Flores, H. Tito-Chura, and V. Yana-Mamani, "Wind speed time series prediction with deep learning and data augmentation," in *Intelligent Systems and Applications*, K. Arai, Ed. Cham: Springer International Publishing, 2022, pp. 330–343.
[18] J. Lei, C. Liu, and D. Jiang, "Fault diagnosis of wind turbine based on long short-term memory networks," *Renewable energy*, vol. 133, pp. 422–432, 2019.
[19] J. Herp, N. L. Pedersen, and E. S. Nadimi, "A novel probabilistic long-term fault prediction framework beyond scada data-with applications in main bearing failure," in *Journal of Physics: Conference Series*, vol. 1222, no. 1. IOP Publishing, 2019, p. 012043.
[20] L. Xiang, X. Yang, A. Hu, H. Su, and P. Wang, "Condition monitoring and anomaly detection of wind turbine based on cascaded and bidirectional deep learning networks," *Applied Energy*, vol. 305, p. 117925, 2022.
[21] "Bearing damage and failure analysis," https://www.skf.com/binaries/pub12/Images/0901d1968064c148-Bearing-failures—14219_2-EN_tcm_12-297619.pdf, accessed: 2021-08-12.
[22] Z. Zhang, "Missing data imputation: Focusing on single imputation," *Annals of translational medicine*, vol. 4, 2016.
[23] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. [Online]. Available: https://aclanthology.org/D14-1179
[24] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555
[25] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (gru) neural networks," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2017, pp. 1597–1600.
[26] J. S. Hunter, "The exponentially weighted moving average," *Journal of Quality Technology*, vol. 18, no. 4, pp. 203–210, 1986. [Online]. Available: https://doi.org/10.1080/00224065.1986.11979014
[27] L. S. Nelson, "The deceptiveness of moving averages," *Journal of Quality Technology*, vol. 15, no. 2, pp. 99–100, 1983. [Online]. Available: https://doi.org/10.1080/00224065.1983.11978852
[28] P. Cisar and S. M. Cisar, "Ewma statistic in adaptive threshold algorithm," in *2007 11th International Conference on Intelligent Engineering Systems*, 2007, pp. 51–54.
[29] C. McKinnon, A. Turnbull, S. Koukoura, J. Carroll, and A. McDonald, "Effect of time history on normal behaviour modelling using scada data to predict wind turbine failures," *Energies*, vol. 13, no. 18, p. 4745, 2020.

VIII. Biography Section

Ángel Encalada-Dávila received his bachelor's degree in Mechatronics Engineering in 2021 from Escuela Superior Politécnica del Litoral. He is currently working as a Data Scientist in an international data analytics company. Moreover, he is editor of the CIDiER 2021 Conference Proceedings (Book of Series "Green Energy and Technology", Springer). He is the author of 5 journal articles, and 7 conference papers.

Luis Moyón received his bachelor's degree in Mechatronics Engineering in 2022 from Escuela Superior Politécnica del Litoral. Currently, he works in an international software company and collaborates with Universidad Ecotec. His areas of research interest are robotics, automation and control systems, artificial intelligence, and software development.

Christian Tutivén received the B.E. degree in Telecommunications in 2007 from Universidad Católica Santiago de Guayaquil, Ecuador, and the Ph.D. in 2018 from the Universitat Politècnica de Catalunya, Barcelona, Spain. Since 2019, he has been with the Faculty of Mechanical Engineering and Production Science and Mechatronics Engineering at Escuela Superior Politécnica del Litoral, where he is an associate professor. He is the author of 10 journal articles, and 28 conference papers.

Bryan Puruncajas is a Mechatronics engineer, graduated from the Universidad de las Fuerzas Armadas ESPE, Ecuador, in 2015. He was granted a scholarship by the National Taipei University of Technology, Taiwan, to pursue a master's degree in 2016. He is currently a PhD student at the Universitat Politècnica de Catalunya, Spain, and lecturer at the Escuela Superior Politécnica del Litoral, Ecuador. He is the author of 4 journal articles, and 5 conference papers.

Yolanda Vidal received the B.E. degree in mathematics in 1999 and the Ph.D. degree in 2005 from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. Since 2001, she has been with the Department of Mathematics and the Barcelona East School of Engineering (EEBE), at UPC, where she is an associate professor. She is an IEEE Senior member and is the author of 50 journal articles, and more than 100 conference papers.