Received: 17 June 2019 | Revised: 22 February 2020 | Accepted: 17 March 2020
DOI: 10.1002/we.2510

WILEY

RESEARCH ARTICLE

Deep learning with knowledge transfer for explainable anomaly prediction in wind turbines

Joyjit Chatterjee | Nina Dethlefs

Department of Computer Science and Technology, University of Hull, Yorkshire, UK

Correspondence
Joyjit Chatterjee, Department of Computer Science and Technology, Big Data Analytics Research Group, Robert Blackburn Building, University of Hull, Cottingham Road, Hull HU6 7RX, UK.
Email: j.chatterjee-2018@hull.ac.uk

Abstract

The last decade has witnessed an increased interest in applying machine learning techniques to predict faults and anomalies in the operation of wind turbines. These efforts have lately been dominated by deep learning techniques which, as in other fields, tend to outperform traditional machine learning algorithms given sufficient amounts of training data. An important shortcoming of deep learning models is their lack of transparency—they operate as black boxes and typically do not provide rationales for their predictions, which can lead to a lack of trust in predicted outputs. In this article, a novel hybrid model for anomaly prediction in wind farms is proposed, which combines a recurrent neural network approach for accurate classification with an XGBoost decision tree classifier for transparent outputs. Experiments with an offshore wind turbine show that our model achieves a classification accuracy of up to 97%. The model is further able to generate detailed feature importance analyses for any detected anomalies, identifying exactly those components in a wind turbine that contribute to an anomaly. Finally, the feasibility of transfer learning is demonstrated for the wind domain by porting our "offshore" model to an unseen dataset from an onshore wind farm. The latter model achieves an accuracy of 65% and is able to detect 85% of anomalies in the unseen domain. These results are encouraging for application to wind farms for which no training data are available, for example, because they have not been in operation for long.

KEYWORDS

LSTM, SCADA, SMOTE, transfer learning, XGBoost

1 | INTRODUCTION

Wind energy has become the fastest growing renewable energy resource over the last few decades in terms of popularity and global uptake, owing to rapid technological developments[1] in areas of aerodynamics, structural dynamics and so on. As wind turbines consist of several electrical and mechanical components, including generator and bearings, they experience irregular loads and inconsistent operational behaviour. This situation is even more complex for offshore wind farms, wherein the average wind speeds are higher and conditions at sea challenging, requiring the turbines to be more rugged.[2] Operations and maintenance (O&M) is key to monitoring such inconsistent behaviour of turbines and preventing any incipient faults. Previous studies estimate that almost 30% of the total cost of wind power generation is owed to O&M,[3] motivating research into efficient and effective automation of O&M procedures.

Generally, there have been three approaches to predicting faults in turbines:[4] (1) a data-driven approach of using supervisory control and acquisition data (SCADA) to predict future events,[5] (2) a signal processing-based approach for analysing vibrational signals[6] and (3) a numerical model-based approach wherein physical models of a turbine and its subcomponents (e.g., gearbox, blades, etc.) are used to identify anomalies in operations.[7] In this article, a data-driven approach is utilised, particularly focusing on deep learning algorithms.[8] The data-driven approach

Peer Review The peer review history for this article is available at https://publons.com/publon/10.1002/we.2510.

Abbreviations: LSTM, long short-term memory; SCADA, supervisory control and data acquisition; SMOTE, Synthetic Minority Oversampling TEchnique; XGBoost, eXtreme Gradient Boosting.

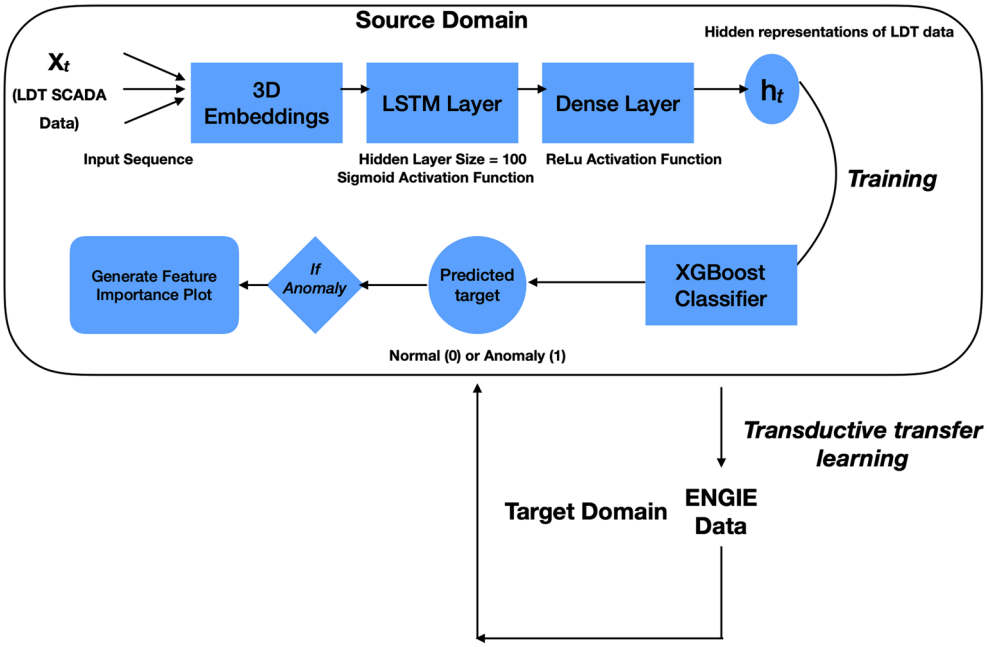

FIGURE 1 Proposed LSTM-XGBoost hybrid model. LDT, Levenmouth demonstration turbine; LSTM, long short-term memory; SCADA, supervisory control and data acquisition; XGBoost, eXtreme Gradient Boosting [Colour figure can be viewed at wileyonlinelibrary.com]

is often most cost effective and achieves very reasonable accuracy given sufficient training data. However, it can suffer from a lack of access to sufficient amounts of training data as well as frequently occurring class imbalance in existing SCADA datasets. Another key issue is the lack of transparency in deep learning models due to their black box nature—being able to predict an anomaly is of limited use if no diagnosis can be provided alongside, detailing exactly which components are affected and the possible causes. Such information can help O&M engineers to perform appropriate prognosis and maintenance operations easily.

Our primary computational model in this article is the long short-term memory network (LSTM), a special type of recurrent neural network capable of learning from long sequences and therefore particularly suited to modelling time-series data, such as SCADA. We use LSTMs to make accurate predictions of impending faults. To also address explainability and provide rationales alongside model predictions, a gradient boosted (XGBoost) decision tree classifier is integrated, which can identify the most prominent input features giving rise to an incipient anomaly. Please refer to Figure 1 for a graphical illustration of the learning model. Our prediction model achieves an accuracy of up to 97% in experiments with two datasets. As a second step, the possibility of deploying our model in a transfer learning setting is explored. Here, the aim is to port our trained model from its original (source) domain to a new unseen (target) domain and identify anomalies for unseen turbines without retraining the model. We show in experiments comparing an offshore and an onshore wind farm that our hybrid LSTM+XGBoost model is transferable across domains, achieving an accuracy of 65%, significantly higher than a simple majority baseline.

The article is organised as follows. Section 2 discusses related work on machine learning for fault prediction in wind turbines. The proposed model is presented in Section 3 and motivates the need and application of transfer learning. Datasets are introduced in Section 4, alongside details on preprocessing and sampling. Section 5 discusses our experiments and results, whereas Section 6 concludes and gives an outlook on future work.

2 | RELATED WORK

Machine learning has been used in the last decade to analyse and predict faults and anomalies in turbines. Most existing work relies on traditional methods including support vector machines, decision/regression trees or probabilistically informed heuristics. As an example, Zhao et al.[9] apply support vector machines to classify normal and faulty operations of a turbine from SCADA data with an accuracy of 94%. In a similar vein of work, Yang et al.[10] use support vector regression to develop a reconstruction-based model for fault detection in turbine subcomponents, thus aiming for a more focused output prediction in terms of error location. The authors demonstrate that support vector regression can be used for identifying anomalies in signals obtained from SCADA data based on the residual error between these signals. An alternative technique that is slightly less computationally expensive and has shown comparable results are decision trees, see Si et al.[5] and Abdallah et al.[11]

Other approaches have aimed to predict faulty turbine behaviours using idealised power curves. Du et al.[12] use Euclidean distance to identify data points falling off the ideal power curve. They demonstrate that computational expense can be reduced through dimensionality reduction and feature selection prior to classification but provide no accuracy results to help assess reliability of their technique.

Supervised methods possess the primary advantage of providing a clear relationship between input features and outputs, which system designers can inspect for plausibility in monitoring the behaviour of turbines. However, approaches like the above have been outperformed in recent years by deep learning—a family of algorithms built around neural networks that are able to learn non-linear data patterns using abstraction over multiple hidden layers.[13] Several recent studies have explored deep learning for condition monitoring in wind turbines[14] and for specific subcomponents, for example, Lu et al.,[15] who estimate the lifetime of the pitch system, gear box, rotor and generator. Other studies have looked

into automating visual inspection of turbine components, see Yu et al.[16] and H. Li et al.[17] Yu et al.[16] use a convolutional neural network to process images of turbines into a reduced set of relevant features that can be used for fault prediction. This methodology is effective for external parts of the turbine that can be monitored with drones, for example, but is less straightforward to apply internally.

The idea of identifying anomalies in turbine operation based on deviations from an ideal power curve has also been addressed with deep learning. Papatheou et al.[18] use neural networks for predicting power in a wind farm. They use the concept of a Gaussian process to show that any outlier from outside the confidence intervals of the predictive distribution can be considered a novelty, thereby a potential fault. In a similar task, Marvuglia et al.[19] provide a comparison of different types of neural nets in modelling the power curve/s of a wind farm. They show that a general regression neural network achieves good performance particularly in modelling non-linear functions with discontinuity, as, for example, a power curve. Other authors have used data from turbines to model bivariate probability distribution functions and capture any deviations from the copula models of the turbine's ideal power curve.[20,21]

Other work on analysing subcomponent behaviour includes Zaher et al.,[14] who model the normal behaviour of generator and gearbox from SCADA data and use temperature-based anomaly detection with a multilayer perceptron to identify outlier behaviour, thereby making predictions without the need for large amounts of fault data. Although this work demonstrates the potential of anomaly detection techniques, it still relies heavily on input of engineers to interpret the output and draw conclusions. Following a different methodology, Andersen et al.[22] predict faults based on vibrational signals. The authors use historical vibrational analysis data from onshore turbines, labelled with fault/no fault situations, and show that a convolutional neural network outperforms baselines to achieve over 97% of training accuracy. Wang et al.[23] demonstrate that the selection of prediction features themselves can be beneficial. The authors use gearbox lubricant pressure as the predictive signal, unlike most other works which use gearbox oil temperature for making predictions, thus demonstrating that unconventional parameters from SCADA data may well be useful in identifying anomalies in the turbine subcomponents. Other authors have aimed to predict impending faults in real time and from live turbine data, see, for example, Ibrahim et al.,[24] who achieve a notable accuracy of 98% using current signature analysis and artificial neural networks from simulated data. Finally, an active though still underexplored area of research is the use of recurrent neural networks[25] for modelling performance of turbines. Recurrent neural networks are particularly suited for modelling time-series data due to their ability to capture temporal information over multiple time steps, such as meteorological information or turbine measurements.[26] Current studies have mostly used long short-term memory models for power forecasting in wind turbines, see, for example, Liu et al.[27] or Lei et al.[28] for a study on fault diagnosis.

In summary, the above studies make important advances towards more accurate power forecasting and anomaly prediction in turbines. Although deep learning models achieve high accuracy in many cases, they largely cannot provide rationales for their decisions and therefore be of limited use in some scenarios. The models may learn to predict that a fault is likely to occur but without information on details of the fault, it will be difficult to address and avert. In this article, we hope to bridge this gap between accuracy and transparency in traditional models.

3 | LEARNING MODEL

3.1 | Long short-term memory model

Just as humans might interpret the meaning of a word given the context of a complete sentence, in a similar way, LSTMs generate predictions from a sequence of temporally ordered observations that they receive as input, for example, a sequence of SCADA logs obtained at consecutive time intervals. LSTMs are special type of recurrent neural network (RNN) that have been shown to successfully make predictions from long input sequences due to their ability to learn long-term dependencies from sequential data and overcoming the problem of vanishing and exploding gradients in standard recurrent neural networks.[29,30] First proposed by Hochreiter and Schmidhuber,[31] LSTMs are generally more successful than their standard counterparts on many real-world sequence classification tasks, such as machine translation or speech recognition.[32,33]

The general setup of our learning task is to make an output prediction y from a vector of input features X. X is defined to be a sequence of SCADA measurements, and we define our output prediction y to denote either 'normal operation' of the wind farm or 'abnomality'. Deep learning models learn to map inputs X to an output y by estimating an abstract hidden representation h, where $h = f(X)$ and f refers to a non-linear activation function, for example, tangent or sigmoid. As recurrent neural networks work by taking temporal information into account in their prediction making, h is in fact estimated recursively as $h_t = f(x_t, h_{t-1})$, where t is the current time step. The goal is to then minimise the loss between expected and generated outputs ($\mathbf{y}$, $\hat{\mathbf{y}}$), respectively, using, for example, cross-entropy:

$$(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{N} \sum_i y_i \quad \log \quad \hat{y}_i. \tag{1}$$

Figure 2 depicts the architecture of an LSTM learning model[8] where a hidden state h_t is computed from an input x_t, a previous hidden state h_{t-1} and a previous cell state c_{t-1} under consideration of several 'gates' that control updates to the cell state C_t. The cell state C_t captures information that is passed between time steps, incorporating X as well as any gate updates (f_t, i_t or o_t). Each of the gates are briefly defined below, see, for example, Graves[34] for a detailed introduction to LSTMs.

1. The forget gate f_t is modelled as a sigmoid function that decides at each time step how much of the current information in the cell state is retained or forgotten. The output of the forget gate f_t is a real valued number between 0 (all information is forgotten) and 1 (all information is retained), see Equation (2).

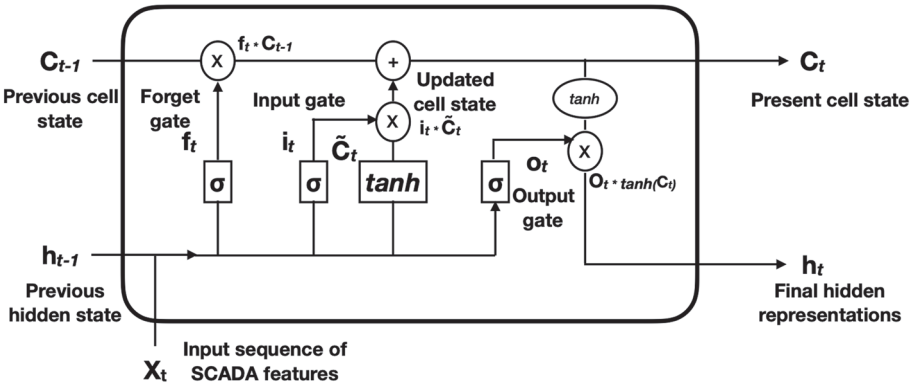

FIGURE 2 Architecture of an LSTM model with gates learning from SCADA data. LSTM, long short-term memory; SCADA, supervisory control and data acquisition

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{2}$$

2. The input gate i_t is a sigmoid layer within the network that decides how much new information to add to the cell state at each time step. A *tanh* squashing function distributes the values of the cell state between −1 and 1 to create an updated cell state $\tilde{C}_t$. See Equations (3) and (4).

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{3}$$

$$\tilde{C}_t = \tanh\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right) \tag{4}$$

3. The cell state C_t is then updated by multiplying the previously updated cell state C_{t-1} with the output of the forget gate f_t. Also, the values of the new input are added based on scaling the input gate values i_t with the updated cell state $\tilde{C}_t$, see Equation (5).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

4. Finally, the output gate o_t determines what the output of the cell state should be. A sigmoid activation function σ is firstly used to limit the output to the essential components of the cell state and to ignore the redundant values in the feature space. Then, the *tanh* squashing function is used to restrict values to the range of −1 and 1, see Equations (6) and (15).

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{6}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{7}$$

Given that SCADA data from turbines can be highly complex, non-linear and contain multivariate time series, it is expected that the LSTM will help capture any relevant long-term dependencies in the data as well as automatically identify the features required to optimally make predictions for the turbine's operating condition—thereby eliminating the need for manual feature engineering.

3.2 | XGBoost

XGBoost is a scalable and effective implementation of the popular gradient boosted decision trees algorithm first proposed by Chen and Guestrin.[35] It is a supervised learning method, which builds a prediction model using an ensemble of decision tree classifiers to produce optimal results even from sparse data samples.[36] Figure 3 illustrates an example of a decision tree in our domain, which works by checking for various conditions and making decisions based on threshold values. The ensemble scenario uses multiple such trees, each learning to make separate output predictions, which are then combined into a final joint output. In our scenario, each decision tree classifier is trained on a separate portion of the training data.

The working of the XGBoost model is briefly summarised below:

1. First, the objective (loss) function for the XGBoost model $\mathcal{L}^{(t)}$ is to be minimised. The loss function $\mathcal{L}^{(t)}$ is dependent on the actual labelled values of targets in the training data y_i and is a function of several different classification and regression tree (CART) learners at each successive iteration t, as outlined in Equation (8). Here, i denotes a particular training sample for the learner. On using second order Taylor approximation for minimising $\mathcal{L}^{(t)}$, the final version of the simplified objective function is obtained as in Equation (9), where g_i and h_i are the statistical gradient approximations of the loss function of first and second order, respectively.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t\left(\mathbf{x}_i\right)\right) + \Omega\left(f_t\right) \tag{8}$$

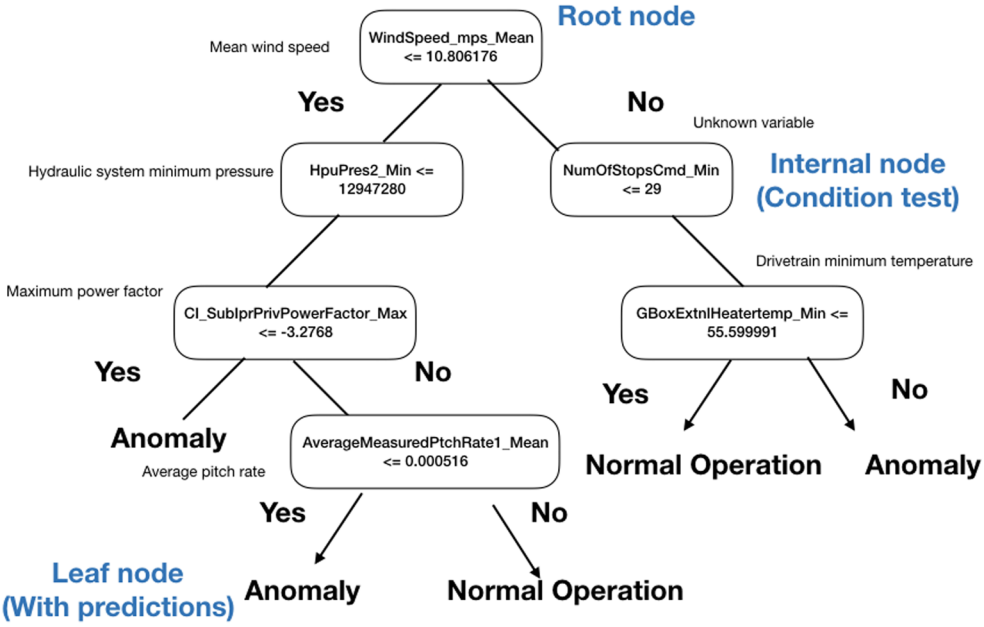

FIGURE 3 Visualisation of an example decision tree ensemble for supervisory control and data acquisition data [Colour figure can be viewed at wileyonlinelibrary.com]

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \Omega(f_t) \tag{9}$$

2. The XGBoost model then finds the successive learners for the next iterations by looking for the decision trees that ensure the minimum loss. This is achieved using a scoring function q, which evaluates the reduction in loss for each learner by iterating over all prevailing features in the training data and evaluating the loss reduction at each successive node. It achieves this using the *exact greedy algorithm* to greedily minimise the value of the objective function at time step t. See Equation (10).

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{\left(\sum_{i \in I_j} g_i \right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{10}$$

The minimal point is located by isolating the weights w_j of the classifier model using the following rule in Equation (11).

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{11}$$

The final gain obtained is then evaluated. Assuming that the algorithm splits the root node into two leaf nodes L_L and L_R for the decision-making process at any instance, the split objective function obtained is shown in Equation (12).

$$\mathcal{L}_{split} = \frac{1}{2} \left[\frac{\left(\sum_{i \in I_L} g_i \right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i \right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{\left(\sum_{i \in I} g_i \right)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma \tag{12}$$

3. Finally, the XGBoost model computes a probability score t for each successive prediction in the range of $\{0, 1\}$ by using a cross-entropy loss function for binary classification, see Equation (13).

$$y \ln(p) + (1 - y) \ln(1 - p), \quad \text{where } p = \frac{1}{(1 + e^{-x})} \tag{13}$$

The probability is computed after applying the sigmoid function to obtain importance of the features for a particular predicted target. This is the key feature allowing the XGBoost model to both classify new data in a binary context as well as evaluate the ranking of predictors for the learning model, thereby providing the transparency of feature importance we seek in our deep learning model.

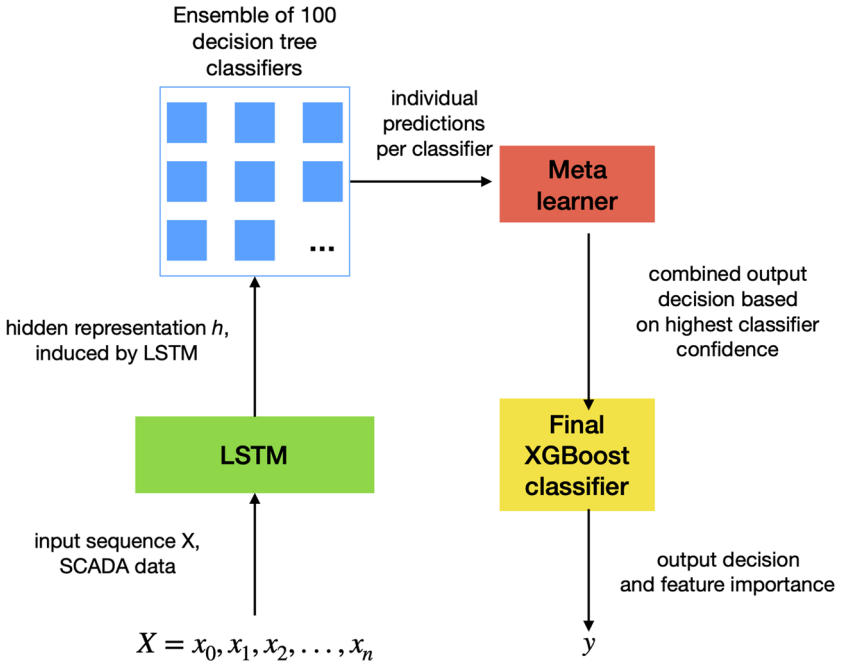

FIGURE 4 Bringing LSTM and XGBoost models together, where the final hidden layer of the LSTM feeds into the XGBoost classifier ensemble. LSTM, long short-term memory; SCADA, supervisory control and data acquisition; XGBoost, eXtreme Gradient Boosting [Colour figure can be viewed at wileyonlinelibrary.com]

3.3 | Augmentation of LSTM with XGBoost

In this section, the novel idea of bringing together the LSTM and XGBoost models is presented to provide accurate and transparent prediction making for anomaly detection in wind turbines. The step by step process for augmenting the LSTM with an XGBoost model is explained below. Figure 4 illustrates our approach, where the LSTM will initially make predictions based on SCADA inputs and the XGBoost classifier will estimate feature importance from the input based on the hidden representation of the LSTM. The latter is done for multiple decision trees in the ensemble which then get combined by a meta-learner for a final vote. A similar model was presented for predicting hypoxemia in operating room of hospitals,[25] but the earlier model is extended to the wind domain, adding support for continuous reading from SCADA logs and transfer learning.

The following steps outline our procedure:

1. Multivariate time-series SCADA data x_t are fed as input to the LSTM network alongside a desired output prediction y representing the ground truth for each data point (normal operation/anomaly). The LSTM generates values at the output gate o_t, see Equation (6).
2. The hidden state h_t (captured by o_t) is forwarded to a dense layer, see Equation (14), where y_t is an intermediate output of the LSTM, which is used to map the hidden representations to a particular subtree in the XGBoost classifier.

$$y_t = W * o_t \tag{14}$$

3. As above in Equation (15), the final hidden representation $h_t = \{h_0, h_1, h_2 \ldots \ldots, h_{t-1}\}$ is obtained as (see also Equation 15):

$$h_t = o_t * \tanh(C_t). \tag{15}$$

4. The hidden representations of the input data thus obtained are now used as training samples for the XGBoost classifier. As outlined before, the XGBoost model takes into account not just one decision tree but an ensemble of multiple trees to make optimal predictions. Consider that the model has N weak classifiers, with each being trained on a subset of the training samples h_t. In this case, the weights of all the weak classifiers are combined into a joint final classifier, which takes into account the maximum likelihood of each prediction being labelled correctly based on the ground truth training samples $y \in h_t$. The best classifier is used for making the final predictions $\hat{y}$ as per Equation (16). $\hat{y}$ is a binary value, either a 0 (normal operation) or 1 (anomaly) representing the operational behaviour of the turbine. A total of 100 classifiers were used as estimators in our ensemble.

$$\hat{y} = \underset{y \in h_t}{\arg\max} l(y; y_1, \ldots, y_N) \tag{16}$$

3.4 | Transfer learning

Conventionally, machine learning algorithms are trained for a specific task given a dedicated dataset of the target domain. In real data applications, this scenario is often limiting as models need to be retrained, often from brand new datasets for any new, even related domain,[37] making data collection expensive and often infeasible. This is also true for the wind domain, where large reliable datasets for new wind farms and turbines are not always readily available. In this section, transfer learning is introduced and further applied to the proposed hybrid LSTM +XGBoost learning model.

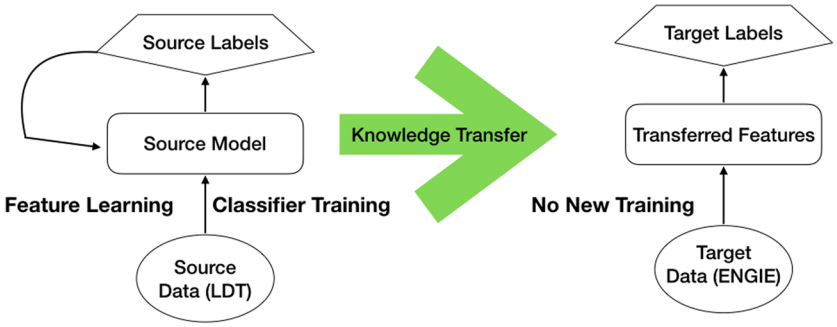

FIGURE 5 Knowledge transfer from source to target domain. LDT, Levenmouth demonstration turbine [Colour figure can be viewed at wileyonlinelibrary.com]

The essence of transfer learning is to ensure that knowledge gained in one domain (the source domain) allows optimal generalisation in a different unseen (target) domain, making the learning process more accurate and computationally efficient in cases where there is a lack of labelled training data.[37] A common distinction is made between inductive and transductive transfer learning. Inductive transfer learning is the most common type of knowledge transfer, wherein the model learns from labelled examples in the source domain to predict the labels for new examples which it has not seen before in the target domain.[38] In this article, the focus is on transductive transfer learning, wherein the source domain labels are available, but the target domain has unlabelled data. To formally define transductive transfer learning, we consider a domain $\mathcal{D}$, which is made up of the feature space $\mathcal{X}$, and the marginal probability distribution set $P(X)$.[39] Thereby, the domain can be represented by

$$D = \{\mathcal{X}, P(X)\}, \tag{17}$$

where $X = \{x_1, \ldots, x_n\}, x_i \in \mathcal{X}$ is the training data for the model with n samples. Considering a source domain $D_S = \left\{\left(x_{S_1}, y_{S_1}\right), \ldots, \left(x_{S_{n_S}}, y_{S_{n_S}}\right)\right\}$, where $x_{S_i} \in \mathcal{X}_S$ denotes a particular sample in the training dataset and $y_{S_i} \in \mathcal{Y}_S$ denotes the actual label in the dataset. We can map this to a target domain $D_T = \left\{\left(x_{T_1}, y_{T_1}\right), \ldots, \left(x_{T_{m_T}}, y_{T_{m_T}}\right)\right\}$, where $x_{T_i} \in \mathcal{X}_T$ and $y_{T_i} \in \mathcal{Y}_T$ are the actual predicted output in the target domain that is to be estimated. In this scenario, transductive transfer learning can be mathematically defined as follows:

- The source domain D_S and target domain D_T are different but closely related, that is, $D_S \neq D_T$.
- The target task to be solved in both the source domain and target domain is same, that is, $\mathcal{Y}_S = \mathcal{Y}_T$.
- The feature space distribution of the data in the source and target domains is different, that is, $\mathcal{X}_S \neq \mathcal{X}_T$.
- The marginal probability distribution sets of the source and target domains are different, that is, $P(X_S) \neq P(X_T)$.

Transductive transfer learning solves the target task $\mathcal{Y}_T$ of making predictions in the new but related domain D_T by leveraging the knowledge and experience of learning from the training data in the source domain D_S. This makes it possible to generate predictions in situations where the target domain lacks sufficient training data for a dedicated model.

In this article, the hybrid LSTM+XGBoost model is utilised in a transductive transfer learning setting to predict anomalies in operation of an onshore wind farm based on a model trained from a single offshore wind turbine. Figure 5 illustrates the process of domain to domain knowledge transfer.

4 | DATA DESCRIPTION AND PREPROCESSING

4.1 | Source domain: Levenmouth demonstration turbine

As a source domain, SCADA data are utilised, measured at 10-min intervals from the Levenmouth demonstration turbine (LDT),* a 7-MW offshore wind turbine located off the coast in Fife, Scotland. Similar to most existing SCADA measurements in modern day turbines, the features used for the study include meteorological variables (wind speed, air temperature, etc.), sensor measurements such as rotor speed, pitch angle, active power, etc., and the operational status of turbine subcomponents (gearbox bearing temperature, generator converter speed, etc.). The LDT is rated to operate at 7 MW, but due to issues of noise curtailment, it is restricted to operate at a maximum power of 6.5 MW by the turbine operators. The original SCADA data include data from the Met mast, turbine substation, alarm and control information, temperature and pressure data, electrical and mechanical data and data from the turbine itself. ‡ To facilitate analysis, all information was merged, which was available at the same timestamp across the 10-min interval.

The LDT SCADA data consist of processed events with information of faults that occurred in the turbine as well as a functional group for the fault, such as pitch system, gearbox, hydraulic system, etc. There were a total of 13 functional groups that had the faults, and the specific details are not revealed for confidentiality reasons. For the purpose of this research, an anomaly is considered to have occurred whenever there is a fault raised in the processed events data for alarms in between a specific time duration (*TimeOn*—when the alarm was started and *TimeOff*—when

* Platform for operational data (POD) disseminated by ORE Catapult: https://pod.ore.catapult.org.uk.

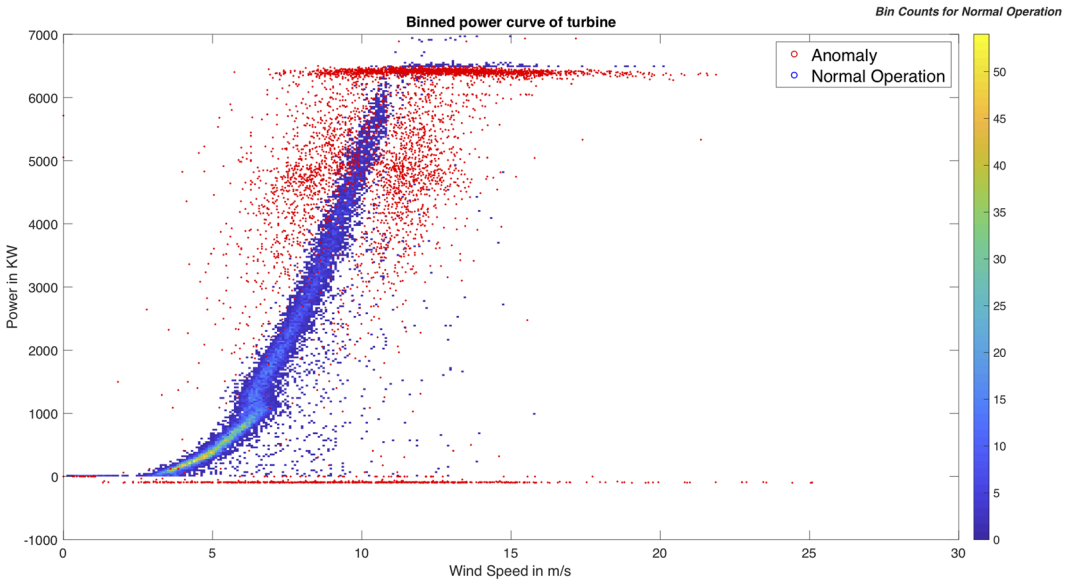

FIGURE 6 Binned power curve for the Levenmouth demonstration turbine under normal and anomaly conditions based on ground truth [Colour figure can be viewed at wileyonlinelibrary.com]

the alarm was cleared). Based on the alarm logs, all faults that had occurred in the turbine's operation were available, and the faults were labelled in the dataset within the duration of these alarms. All other circumstances, wherein there was no alarm raised was considered to be normal operation for this study. Further, to specifically ensure we only target faults that have occurred as a consequence of an actual anomaly in the turbine and not due to requested shutdowns, the forced outages in the turbine are considered based on the unique alarms list to be an indicator of an anomaly. There were several irregular measurements in the SCADA data, including negative power values, power generation at 0 wind speed, etc., which were cleaned during preprocessing. All features were scaled between 0 and 1 as a normalisation procedure, and the same step is later adopted for the target domain to facilitate transfer learning. Finally, there were 21 392 measurements at 10-min intervals for the LDT, each containing 102 features.

The power curve for the LDT under the normal and anomaly conditions is shown in Figure 6. To create a smoother curve for visualisation, the normal operations were grouped together, and a binned scatter plot was developed. It was found that the majority of anomalies are labelled as *partial performance degraded*—a situation in which the prevailing wind speed is sufficient for producing power, but the turbine is not operating at its optimal capacity due to some internal problems within its subcomponents. The power curve shown under normal conditions is based on ground truth of labelled anomaly data from the alarm log and should not be confused with a full-parabolic idealised power curve,[40] which is a reflection of the ideal operation of the turbine under ideal conditions (uniform and steady wind, zero yaw error, etc.). Specifically, ground truth refers to the actual observations (normal operation/anomaly) that had historically occurred in the turbine's operation in real-world conditions, based on the LDT alarm log, and an anomaly for this study denotes a situation wherein there was a fault in one of the 13 different functional groups (pitch system, gearbox, yaw brake, etc.) associated with the turbine.

4.1.1 | Synthetic Minority Oversampling Technique on LDT data

The LDT data used for this study consist of a total of 16 498 instances of normal operation and 4444 instances of anomaly. Considering the large imbalance in these two classes as well as huge imbalances in the types of anomalies in various subcomponents, the Synthetic Minority Oversampling Technique (SMOTE) is used[41] to oversample the minority samples and in this way balance the overall class distribution. This is to avoid creating a bias during training in favour of the majority samples.

SMOTE is a popular statistical algorithm in data analytics, which works by looking at successive samples in a given training dataset and calculating the distance between the *k*-nearest neighbours across the entire feature space.[41] The new synthetic data points are generated by multiplying the vectorial distance between the nearest neighbours across the original dataset with a random real number between 0 and 1 and summing it with the present value of the sample in the feature space. See Chawla et al.[41] for details.

The *imbalanced-learn*[42] python library is used for implementing SMOTE on 80% of the original dataset. The same portion of the data will later be used for training our learning model, leaving 20% of original data for testing. The training data used for SMOTE consisted of 17 112 data points, with 13 594 samples of normal operation and 3518 samples of anomalies. The generated synthetic data have a total of 27 188 samples with 13 594 samples of normal operation and 13 594 samples for anomalies, that is, a perfect balance between the two classes. As an evaluation of our resampling process, the *Kolmogorov–Smirnov test (KS test)* is applied,[43] a statistical non-parametric test method for comparing the equality of a set of probability distributions, in our case between the original and the synthetically generated data. Figure 7 illustrates our results in a heatmap, where a *p* value of 0 signifies completely distinct distributions, whereas 1 signifies perfect equality. As can be seen, the majority of the shared features have at least 75% similarity, and despite sudden valleys in some features, it can safely be accounted that the synthetic data are a

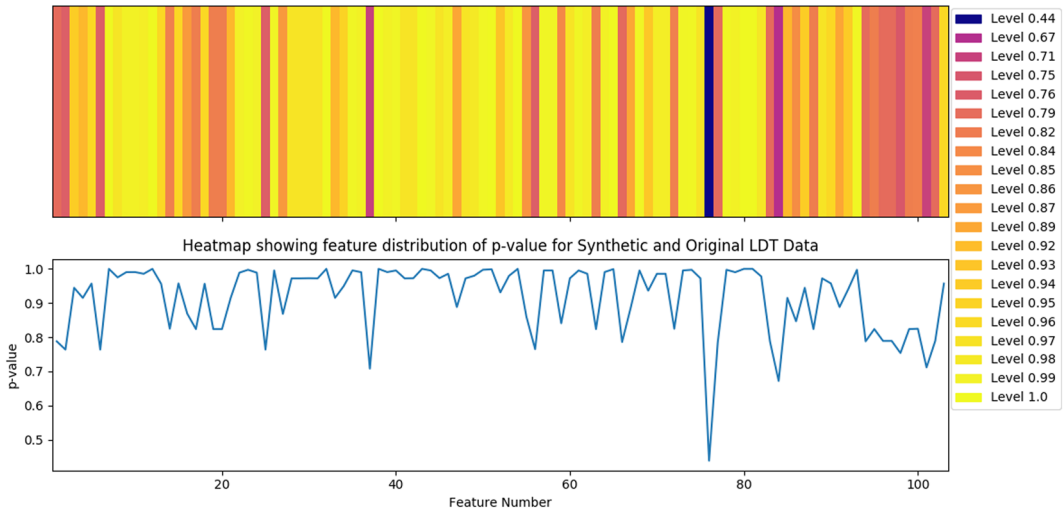

FIGURE 7 Heatmap visualisation of *p* values for original and synthetic Levenmouth demonstration turbine (LDT) data [Colour figure can be viewed at wileyonlinelibrary.com]

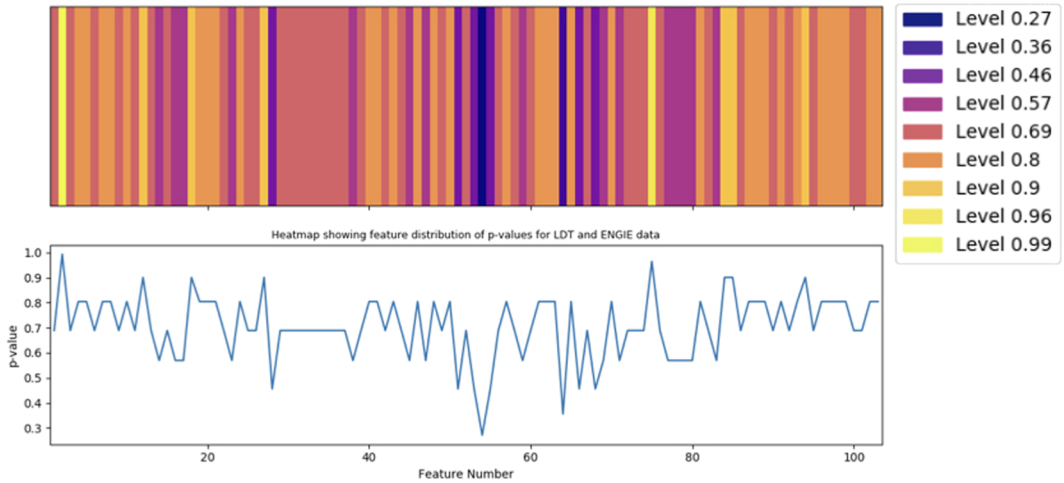

FIGURE 8 Heatmap describing *p* values for the feature distributions of the source and target domains. LDT, Levenmouth demonstration turbine [Colour figure can be viewed at wileyonlinelibrary.com]

good probabilistic approximation of the original data. The average *p* value between the original and synthetically generated data was found to be 0.898 using the KS test.

4.2 | Target domain: ENGIE La Haute Borne

As a target domain, the SCADA data from the ENGIE La Haute Borne open wind farm are used,[†] an onshore wind farm in Meuse, France. This dataset contains of a total of four turbines, each with a rated power of 2 MW, thus making the wind farm capable of producing up to 8 MW of power. The four turbines are denoted by their respective legends as R80711, R80790, R80721 and R80736. The La Haute Borne SCADA data are utilised with the same 102 features that were held in common with the LDT source domain data, and a dataset of 840 380 samples was obtained for the entire wind farm (i.e., 210 095 samples for each turbine). For transfer learning, the exact same processing was applied in the target domain as in the source data. Synthetic data were not created for the target domain. This is because our model was not actually retrained from data in the target domain as we were primarily interested in the effect of transferring the learnt model from the source to the target domain for prediction making. Thereby, the entire data in the target domain were used as the test data. Figure 8 shows the (actual, not ideal) power curves for the four wind turbines.

Seen that our source and target domains are different in terms of the datasets used, the feature distributions in both datasets are notably dissimilar. This is illustrated with a heatmap similar to Figure 7 in Figure 8, showing that most of our 102 features follow a substantially different

[†]ENGIE Open Data Wind Farm: https://opendata-renewables.engie.com/pages/home/.

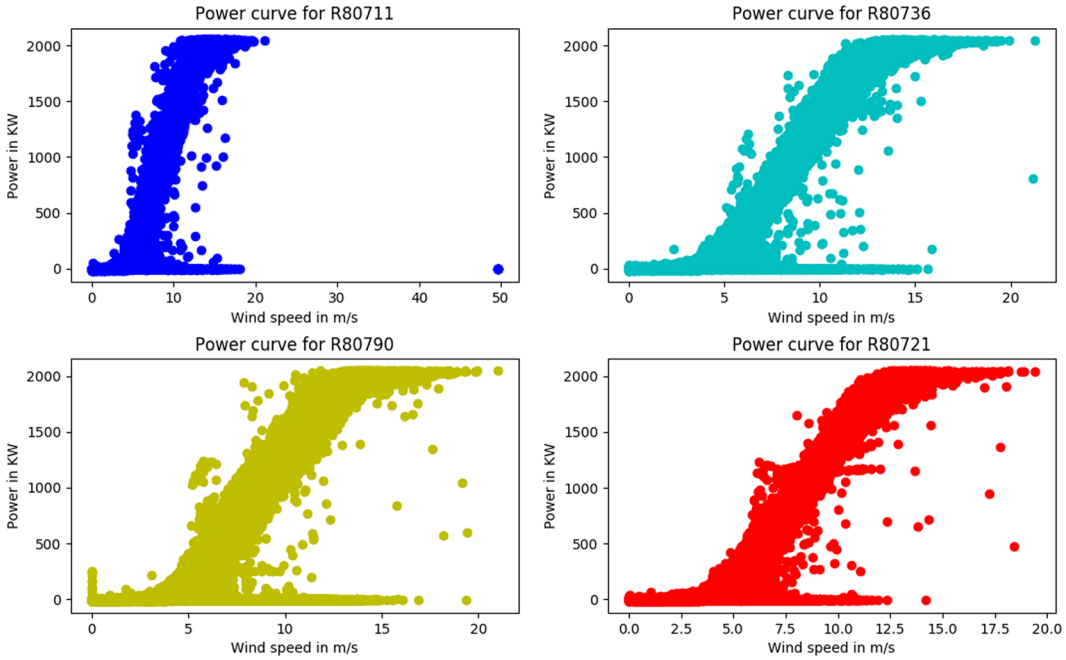

FIGURE 9 Power curves for all turbines in the La Haute Borne wind farm based on original data [Colour figure can be viewed at wileyonlinelibrary.com]

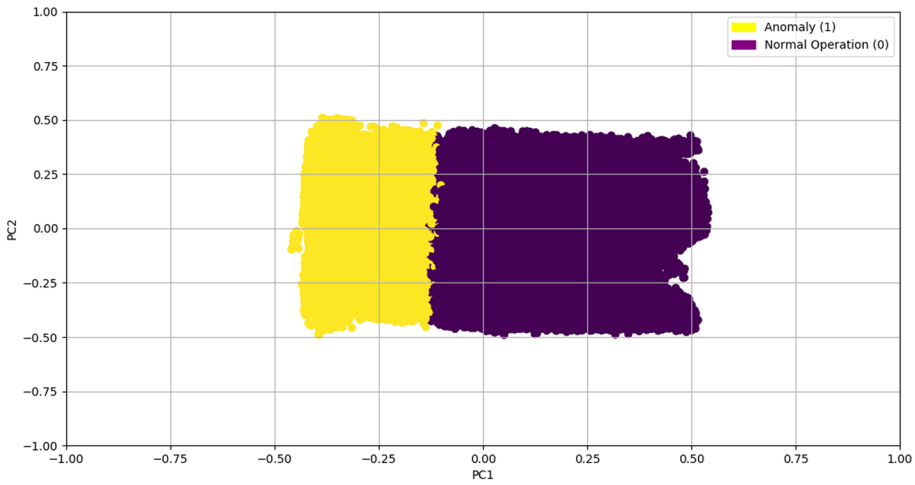

FIGURE 10 Scatter plot for the clustered labels pertaining to the top two principal components [Colour figure can be viewed at wileyonlinelibrary.com]

distribution in both datasets. For the target domain, the average p value was obtained to be 0.705 using the KS test. This can be seen as an evidence for the challenge in transfer learning between these two domains.

4.2.1 | *K*-means++ clustering for label validation

As the data obtained from ENGIE is not explicitly labelled with instances of anomalies and normal operation, the k-means++ clustering algorithm is used[44] to group the ENGIE data into two clusters for anomaly and normal operation. The algorithm assigns individual data points to one of k classes depending on their distance from the cluster centroid, see Krishnan[45] for details. We use $k = 2$ over 500 maximum iterations and the Euclidean distance metric for cluster estimation. Although it was not possible to verify these clusters against their true labels, an evaluation in the source domain is offered instead to offer a comparative result on the likely accuracy of the clusters. For the purpose of validating our clustering results before using it on the ENGIE data, the same algorithm was applied to the original LDT data, for which the ground truth labels were available. After running the algorithm 100 times with varying centroid seeds, an accuracy of 87.8% was obtained in predicting normal operation versus anomaly, validating the suitability of the proposed technique for the ENGIE target domain.

Applying k-means++ clustering to our ENGIE data with a total of 840 380 samples, a total of 633 365 samples were classified as normal operation, and the remaining 207 015 samples were classified as anomalies. Figure 10 shows the clusters for the two classes. It can be seen that both classes are close together in some instances with overlaps but also that the overall classes are still clearly separable, a pattern which is not uncharacteristic of real-life data distributions.[46]

SVM type	Kernel	Accuracy (%)
Linear SVM	Linear	92.9
Quadratic SVM	Quadratic	79.9
Cubic SVM	Cubic	49.2
Fine Gaussian SVM	Gaussian	95.2
Medium Gaussian SVM	Gaussian	92.9
Coarse Gaussian SVM	Gaussian	92.8

TABLE 1 Summary of various combinations of support vector machine (SVM) utilised

5 | EXPERIMENTS AND RESULTS

5.1 | Training of LSTM-XGBoost model for LDT source data

The LSTM learning model is trained with *Tensorflow*,[47] an open source Python library widely used in modelling scalable neural network models. A 80–20% split of training and test data was utilised, where our training data consist of the 27 188 SMOTE-sampled data and the test data are the remainder of 4279 data points from the original dataset. Thereby, the test data that were used for evaluating our model performance is a hold-out portion of the original, untouched SCADA data, and it was only the oversampled training data that were used for training the model for learning purposes.

To the best of our knowledge, there is presently no specific set of criteria for selecting an LSTM network architecture, and different learning tasks require varying architectures to be successful. Wherever possible, the known rules of thumb were followed for this purpose, as outlined, for example, by Bengio.[48] In this study, experimentation was performed with different numbers of parameters, for example, number of layers, optimisers, hidden units, etc., and finally, an LSTM network with two layers was utilised, with 100 hidden neurons each, a rectified linear unit (ReLu) activation function, Adam optimisation, an initial learning rate of 0.001 and batch size of 128. The model was trained over a maximum of 2000 epochs. The final model parameters were decided based on randomised trial and error through hyper-parameter optimisation. Similar comparisons were carried out with the learning and dropout rates, choice of optimiser, epochs, hyper-parameters for the optimisers, etc. As the end goal was to ensure a reasonably good accuracy, which is close to state of art as well as facilitate transfer learning, the two hidden layer model was opted for as the final choice. Though, the model can be made deeper by adding more layers, the training time and complexity increases alongside in comparison to a negligible change in performance, so the network architecture opted for seemed to work reasonably well for the given application domain and context. The model loss function was estimated using softmax-based cross-entropy. The hidden representations induced with the LSTM model were then used to fit our XGBoost classifier, which finally produced the outputs (0 for normal operation and 1 for anomalies) as a binary classifier. Because XGBoost returns a probability distribution over the two output classes rather than a discrete decision, a confidence value of 70% was utilised to class an observation as an anomaly, otherwise it was treated as normal operation.

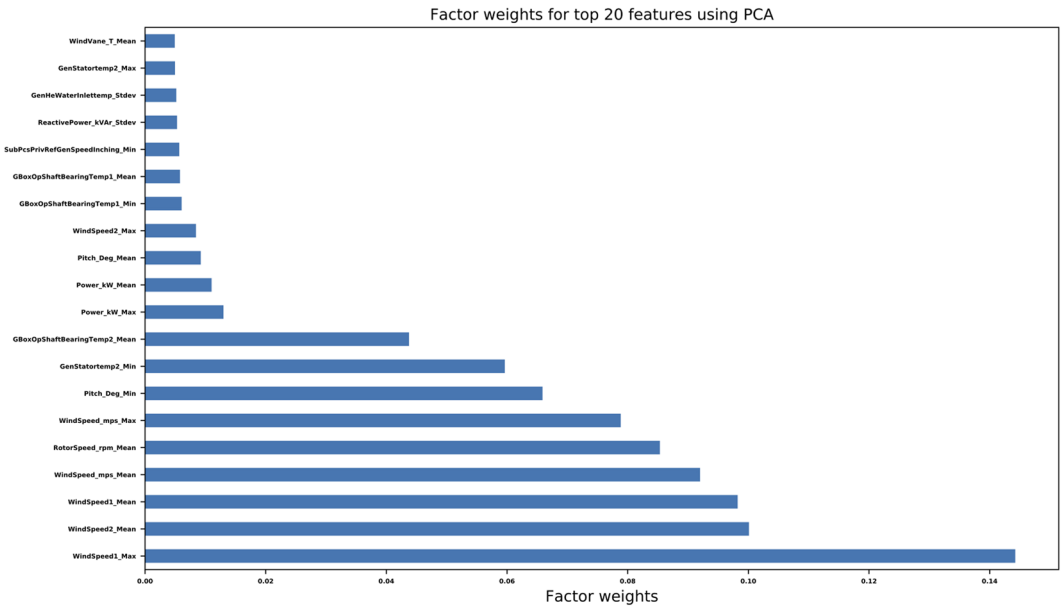

FIGURE 11 Factor weights for top 20 features in the Levenmouth demonstration turbine supervisory control and data acquisition data. PCA, principal component analysis [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Summary of top 3 optimisation characteristics for artificial neural network

Optimiser	Learning rate	Optimisation parameters	Accuracy (%)
Adam[50]	0.001	$\beta_1 = 0.90$, $\beta_2 = 0.999$	93.81
RMSProp[51]	0.001	$\rho = 0.90$	92.64
Stochastic gradient descent[52]	0.01	Momentum = 0 (Undamped)	78.36

TABLE 3 Summary of the performance of various ensemble classifiers

Ensemble classifier	Accuracy (%)
Ensemble boosted trees	95
Ensemble bagged trees	98.25
Ensemble subspace discriminant	88.4
Ensemble subspace K-nearest neighbours	95.8
Ensemble RUSBoosted trees	25.9

TABLE 4 Comparison of results with related work

Related study	Technique used	Accuracy on our data (%)	Reported accuracy in related work (%)
Zhao et al.[9]	Support vector machine	95.2	94
Ibrahim et al.[24]	Current signature analysis and artificial neural network	93.81 (without current signature analysis)	98
I. Abdallah et al.[11]	Ensemble bagged trees	98.25	Not reported
Proposed work	LSTM-XGBoost hybrid model	96.634	N/A

5.1.1 | Accuracy results

To assess our proposed approach in comparison with existing competitive techniques for anomaly prediction in turbines, the accuracy of our model was evaluated against various baselines. As in traditional machine learning, feature selection can play an integral role in improving the model performance, the MATLAB classification learner toolbox was utilised for evaluating various permutations of features, and finally, principal component analysis (PCA) was used to explain 95% of variance of the original features. Fivefold cross-validation was also applied to prevent overfitting on the training dataset. Figure 11 depicts the factor weights for the top 20 features in our data, obtained after principal component analysis.

- A support vector machine (SVM) with a fine Gaussian kernel yielded an accuracy of 95.2% on our LDT data. For the SVM, optimisation was used based on multiple kernel functions, box constraints within [0.001,1000], a kernel scale varying from positive log-scaled values ranging in [0.001,1000] and data standardisation. Table 1 summarises the various combinations of SVM that were evaluated for the study. SVMs have previously been used by Zhao et al.,[9] where the authors obtained an accuracy of 94% using SCADA data from a different source. Interestingly, in Santos et al.,[49] they show that linear kernel-based SVM outperforms artificial neural networks in terms of accuracy. Our results show a similar trend, wherein the SVM outperforms the neural network. However, the fine Gaussian SVM gave the best model performance in our study, possibly due to the variation in the nature of SCADA data used in real world.
- An artificial neural network (ANN) with a single hidden layer, utilising the Adam optimiser, gave us an accuracy of 93.81% on our data. It was seen that adding multiple hidden layers to the ANN did not improve the model performance any further. A variety of optimisations and hyper-parameter tuning was performed, and the results for the top 3 cases are summarised below in Table 2. All the ANN models were trained over maximum of 500 epochs.
  Previous work by Ibrahim et al.[24] combined an ANN with current signature analysis and reported an accuracy of 98% using synthetic rather than real data as in our case.
- Ensemble bagged trees gave us an accuracy of 98.25% on our own data achieving the best performance in our comparison. The learning rate was varied automatically between [0.001,1]. Table 3 summarises the performance of various ensemble classifiers for this study. Abdallah et al.[11] have previously used ensemble bagged trees for classifying the root cause of failures, but no accuracy is reported in the paper.

Table 4 summarises the above comparison. On an interesting note, by performing both training as well as testing on the original data before SMOTE, it was seen that the accuracy dropped to 83.20%, primarily due to the extreme bias of the model towards the majority no fault condition. As the primary aim of this study was to evaluate the feasibility of the technique on unchanged test data from the original SCADA logs before SMOTE, it can be seen that the proposed technique achieves state-of-the-art performance and, in contrast to existing deep learning techniques, balances the trade-off between accuracy and transparency as is discussed in the next section. Notably, in the source domain, 89.41% of faults is correctly identified by the model, with a false discovery rate of 5.26%. Figure 12 illustrates the confusion matrix for the source domain evaluation on the test set.

5.1.2 | Transparency results

Because our XGBoost model provides feature importance for its resulting target predictions based on the F scores, the proposed model is able to provide the list of features that are most likely to contribute to a detected anomaly. This is illustrated in two separate examples in Figures 13

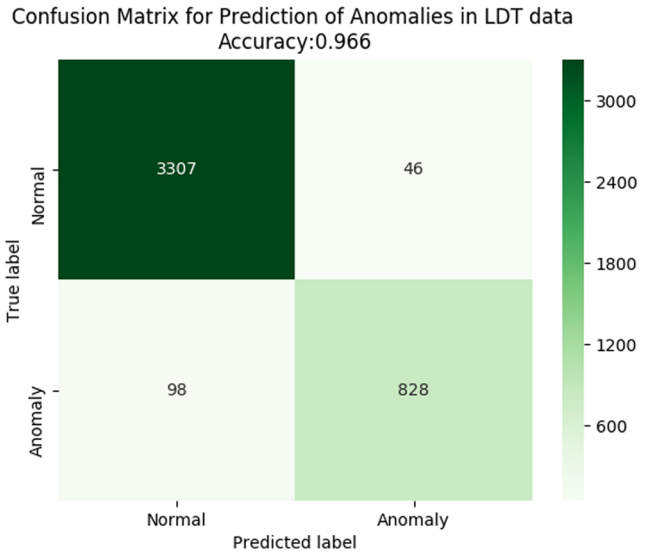

FIGURE 12 Confusion matrix for evaluation of source domain performance with proposed model. LDT, Levenmouth demonstration turbine [Colour figure can be viewed at wileyonlinelibrary.com]

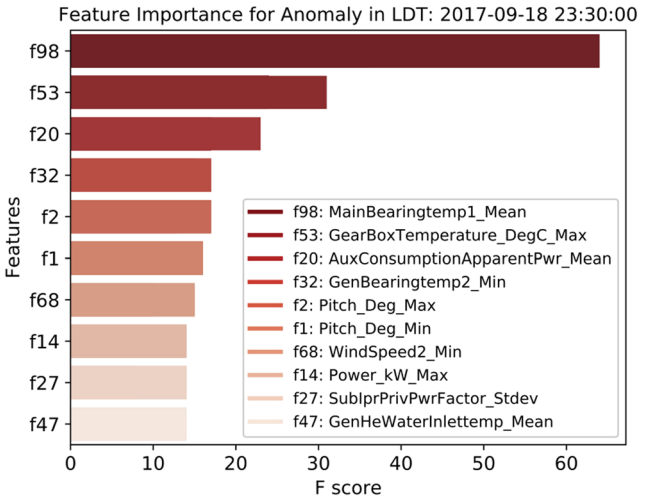

FIGURE 13 Example 1: Feature importance plot when there was an actual anomaly in gearbox (18 September 2017 23:30:00). LDT, Levenmouth demonstration turbine [Colour figure can be viewed at wileyonlinelibrary.com]

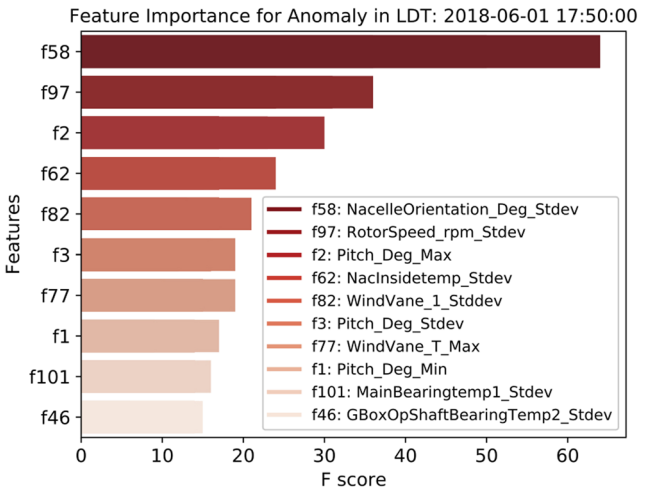

FIGURE 14 Example 2: Feature importance plot when there was an actual anomaly in yaw (1 June 2018 17:50:00). LDT, Levenmouth demonstration turbine [Colour figure can be viewed at wileyonlinelibrary.com]

and 14. Note that the feature importance in each case is arranged in descending order, that is, features with a higher F score are more likely contributing to the anomaly than those with lower F scores.

Figure 13 shows an anomaly in the gearbox. It can be seen that *MainBearingtemp1_Mean* and *GearBoxTemperature_DegC_Max* are the most highly ranked features, which can likely be attributed to overheating of the high speed gearbox shaft bearings and the gearbox housing, which directly contributes to an anomaly in the gearbox. Similarly, Figure 14 shows the scenario for an anomaly in the yaw of the turbine. The highest feature importance is attributed to *NacelleOrientation_Deg_Stdev* and *RotorSpeed_rpm_Stdev*, among other features. As the yaw system of a

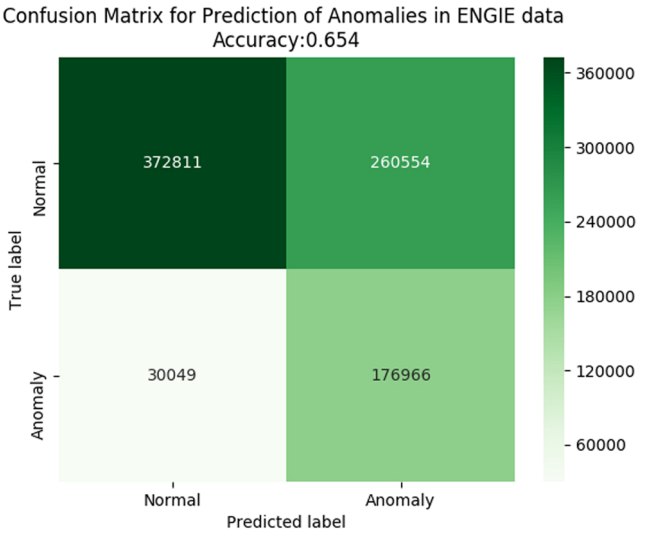

FIGURE 15 Confusion matrix for evaluation of target domain performance with proposed model after knowledge transfer [Colour figure can be viewed at wileyonlinelibrary.com]

turbine is located in between the nacelle and the tower, and the yaw brakes are designed to hold the nacelle in its azimuthal orientation,[53] this shows a possible disorientation of the nacelle is contributing to the anomaly in the yaw system. Further, a wind turbine is said to have an error in the yaw if the rotor is not directly perpendicular to the wind,[54] and the predicted feature importance clearly enunciates some problem in the rotational behaviour of the rotor, which is leading to the anomaly. The feature importance analysis was performed on a randomised portion of the test set, with faults in various functional groups such as pitch system, gearbox, yaw brake, hydraulic system, wind condition alarms, etc., based on the LDT alarm log. ‡ It was observed that the model provided reasonably well in identifying feature importance for faults occurring in the pitch system, followed by the gearbox, wherein, the feature importance was reasonable in around 70% of cases. However, for the hydraulic system, and faults resulting from the moisture vapour transmission rate (MVTR), the observations were not in line with the results expected by expert systems. It is imperative to mention here that although the feature importance was reasonable in most cases, the ultimate judgement on what exactly caused a fault is case specific, and due to lack of historical data on the exact features responsible for the faults due to the commercial sensitivity, instead of relying completely on autonomous feature importance, it is important to combine expert decisions with feature importance for the most reliable results.

Our evaluations in terms of transparency and accuracy show that the LSTM model supports the wind farm operators to learn when the faults would occur based on time-series SCADA measurements. Also, the interpretable and transparent nature of the decision tree facilitates the turbine operators to understand the context of the predicted fault, including which specific measurements in the SCADA data lead to the fault. As the engineers and technicians have expert domain knowledge on averting failures based on irregularities in SCADA measurements, they can take the appropriate steps towards maintenance and repair of specific subcomponents of the turbine. This helps to make machine learning a reliable source of intelligent decision support for wind farm operators, as until recently, given the black box nature of conventional models, such techniques have not seen much uptake in practical applications at the wind farms.

5.1.3 | Model transfer from source domain to target domain

In order to make predictions on the unlabelled ENGIE data, the LSTM-XGBoost model trained for the LDT data was used to transfer any knowledge acquired for the source domain to the ENGIE target domain. The features learned from the source domain are transferred to the target domain, thereby not requiring any additional training with the ENGIE data. Comparing predicted accuracy against our estimated *k*-means++ labels, an accuracy of 65.42% was achieved in the target domain, and it was seen that 85.48% of anomalies are identified correctly. The lower accuracy is caused by a high rate of false alarms identified by our model – 41.14% of cases. Although this number is quite high as false alarms can at times be expensive to have, the figures are encouraging nonetheless seen that predictions are made entirely based on training in a different domain without consulting any ENGIE training data during the process at all. Figure 15 depicts the confusion matrix for model evaluation in the target domain, after knowledge transfer from source.

Similar to the LDT source domain, XGBoost was applied to the target data to extract the subset of relevant features likely contributing to an anomaly. The resulting feature importance plots are illustrated below for two different situations, wherein the anomaly occurred in two different turbines (R80711 and R80736) at different timestamps. Figure 16 shows *DCs_avg* and *DCs_min* to be the leading features, and it can be inferred that the anomaly is caused due to the generator converter speed not being in the suitable range, thereby attributing an error directly to the generator. Similarly, from Figure 17, it can be enunciated that *Wa_avg* and *Ya_std* are the primary features leading to the anomaly. They show an unusual value of the absolute wind direction and the nacelle angle, respectively. This can likely be attributed to a fault in the turbine's rotor,

‡Platform for operational data (POD) disseminated by ORE Catapult: https://pod.ore.catapult.org.uk.

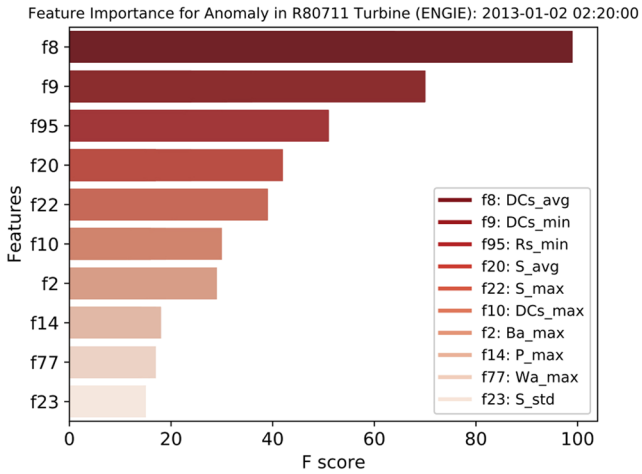

FIGURE 16 Example 1: Feature importance plot for a predicted anomaly in R80711 turbine (2 January 2013 02:20:00) [Colour figure can be viewed at wileyonlinelibrary.com]

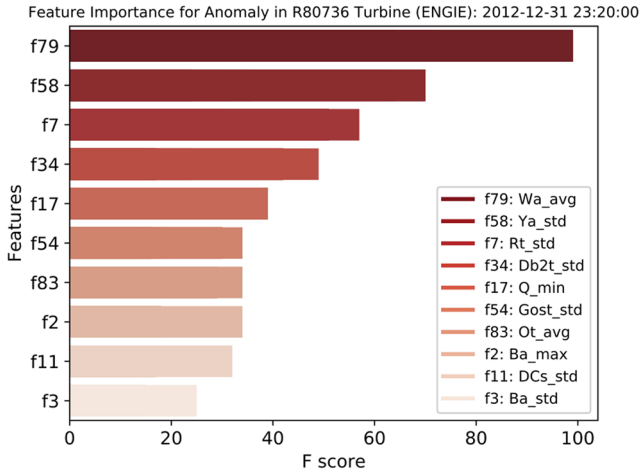

FIGURE 17 Example 2: Feature importance plot for a predicted anomaly in R80736 turbine (31 December 2012 23:20:00) [Colour figure can be viewed at wileyonlinelibrary.com]

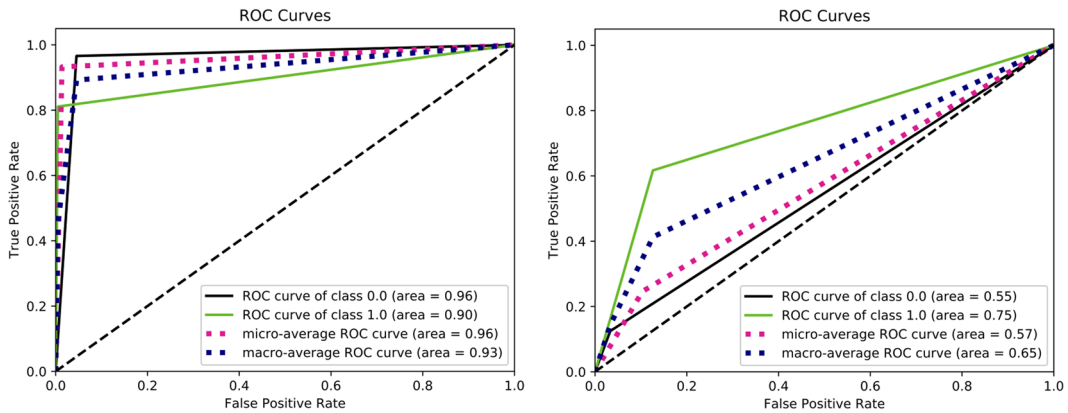

FIGURE 18 Left: receiver operating characteristics (ROC) curve for Levenmouth demonstration turbine data showing normal operation (0) and anomaly (1); Right: ROC curve for ENGIE data [Colour figure can be viewed at wileyonlinelibrary.com]

as the third significant feature *Rt_std* shows an uneven rotor speed, which may be occurring due to a rotor imbalance—a direct consequence of wind shear and torque oscillation in the nacelle.[55]

5.1.4 | Receiver operating characteristics curves

Figure 18 on the left depicts the receiver operating characteristics (ROC) curve for normal operation (0) and anomaly (1) classes in the LDT source domain. It is clear that the area under curve (AUC), which is a vital measure of the performance of the classification model (the closer to 1 the better), is 0.96 for class 0 and 0.90 for class 1, thus showing that the model learnt a high degree of separability between the two classes. This is compared with our model's performance in the target ENGIE domain in Figure 18 on the right. Although there is clearly an expected reduction in the AUC, it is still 0.75 for the anomaly class. This is promising as the model predicts anomalies in the new domain with good reliability. The AUC

for normal operation is only 0.55, owing to the high false positive rate, and this is something that can be an interesting problem to tackle in the near future.

6 | CONCLUSIONS AND FUTURE WORK

To the best of our knowledge, this article is the first in the wind energy domain to propose a tailored deep learning model for anomaly prediction and transparent decision-making, and extending it to transfer learning from one domain to another, eliminating the requirement of training in the new domain. Our study shows that transfer learning is feasible and promising for the wind industry and can be extremely helpful in situations when there is a lack of labelled training data. The wind farm operators can utilise the vast amounts of SCADA data that are available, but not effectively and fully utilised for making intelligent judgements on preventing impending faults in the turbines. Further, the transparency of the model aids explainable predictions, which would encourage uptake of artificial intelligence-based decision-making by the turbine operators. In cases wherein sufficient SCADA data are not available for training (e.g., in new wind farms), the wind farm operators can utilise data available from different wind farms to facilitate decision support. This can also help the operators to evaluate feasibility of operations of new wind farms for which SCADA data have been generated through simulations, but the labelled history on failures is not available. We make the following contributions:

- An LSTM model is applied towards anomaly detection in wind turbines achieving an accuracy of 97%. We confirm that deep learning is promising for wind energy and that models that capture time-series information can outperform models that consider only the local context.
- A novel hybrid model is presented that combines an LSTM for accurate classification with an XGBoost classifier to extract feature importance. This helps us overcome a critical shortcoming of deep learning systems that operate as black boxes and offer no rationales for their decisions. Our model is able to give detailed fault diagnoses for any detected anomaly, which we find to be in line with human analyses for anomalies in various subcomponents in the turbine.
- The feasibility of transfer learning is demonstrated, by porting our learnt model from an offshore wind turbine to an onshore wind farm wherein anomalies are identified in 85% of cases. This is promising considering that the target data are completely unseen and the method can be utilised in any scenario with no or little data to make predictions on reliability of wind farms.

Future work can extend our model with more sophisticated deep learning methods and explore alternative algorithms for transparency, such as attention mechanisms. Because the target domain evaluations for our model show high rates of false alarms, minimisation of false alarm rates would form an integral aspect of future study. An interesting path of future work would be to explore the use of natural language processing for generating detailed error logs for turbines tailored to target readers with different levels of expert knowledge.

ACKNOWLEDGEMENTS

The authors are grateful to the Offshore Renewable Energy Catapult (OREC) for giving us access to the Levenmouth demonstration turbine operational data through platform for operational data. Further, we acknowledge the ENGIE open data wind farm for providing data from the La Haute Borne onshore wind farm. The authors would also acknowledge VIPER, the high-performance computing facility at the University of Hull and its support team, and the Aura Innovation Centre.

ORCID

Joyjit Chatterjee https://orcid.org/0000-0003-2672-3832

REFERENCES

1. Crabtree CJ, Zappala D, Hogg SI. Wind energy: UK experiences and offshore operational challenges. In: Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy, Vol. 229; 2015:727-746.
2. Hara R. Chapter 9—offshore wind. *Wind Power Gen*. 2016:75-84. https://doi.org/10.1214/aoms/1177729586
3. Carroll J, McDonald A, Dinwoodie I, McMillan D, Revie M, Lazakis I. Availability, operation and maintenance costs of offshore wind turbines with different drive train configurations. *Wind Energy*. 2016;20:361-378.
4. Odgaard PF, Johnson KE. Wind turbine fault detection and fault tolerant control—an enhanced benchmark challenge. In: 2013 American Control Conference; 2013.
5. Si Y, Qian L, Mao B, Zhang D. A data-driven approach for fault detection of offshore wind turbines using random forests. In: Iecon 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society; 2017; Beijing, China:3149-3154.
6. Qiao W, Lu D. A survey on wind turbine condition monitoring and fault diagnosis—Part II: signals and signal processing methods. *IEEE Trans Ind Electron*. 2015;62(10):6546-6557.
7. Pandit R. K., Infield D. SCADA-based wind turbine anomaly detection using Gaussian process models for wind turbine condition monitoring purposes. *IET Renew Power Gen*. 2018;12(11):1249-1255.
8. Sun Z, Sun H. Health status assessment for wind turbine with recurrent neural networks. *Math Prob Eng*. 2018;2018:1-16.
9. 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(8):1210.

10. Yang C, Liu J, Zeng Y, Xie G. Real-time condition monitoring and fault detection of components based on machine-learning reconstruction model. *Renew Energy*. 2019;133:433-441.
11. Abdallah I, Dertimanis V, Mylonas H, et al. Fault diagnosis of wind turbine structures using decision tree learning algorithms with big data. In: Safety and Reliability. Safe Societies in a Changing World, Proceedings of the European Safety and Reliability Conference; 2018June; Trondheim, Norway:3053-3061.
12. Du M, Ma S, He Q. A SCADA data based anomaly detection method for wind turbines. In: China International Conference on Electricity Distribution. IEEE; 2016August; Xi'an, China.
13. Schmidhuber J. Deep learning in neural networks: an overview. *Neural Netw*. 2014;61.
14. Zaher A, McArthur SDJ, Infield DG. Online wind turbine fault detection through automated SCADA data analysis. *Wind Energy*. 2009;12:574-593.
15. Lu Y, Sun L, Zhang X, Fenga F, Kang J, Fu G. Condition based maintenance optimization for offshore wind turbine considering opportunities based on neural network approach. *Appl Ocean Res*. 2018;74:69-79.
16. Yu Y, Cao H, Liu S, Yang S, Bai R. Image-based damage recognition of wind turbine blades. In: 2nd International Conference on Advanced Robotics and Mechatronics (ICARM); 2017August; Hefei and Tai'an, China:161-166.
17. Li H, Zhou W, Xu J. Structural health monitoring of wind turbine blades. *Wind Turbine Control Monit Part Adv Ind Control book ser (AIC)*. 2014:231-265.
18. Papatheou E, Dervilis N, Maguire AE, Antoniadou I, Worden K. A performance monitoring approach for the novel Lillgrund offshore wind farm. *IEEE Trans Ind Electron*. 2015;62:6636-6644.
19. Marvuglia A, Messineo A. Monitoring of wind farms and power curves using machine learning techniques. *Appl Energy*. 2012;98:574-583.
20. Gill S, Stephen B, Galloway S. Wind turbine condition assessment through power curve copula modeling. *IEEE Tran Sustainable Energy*. 2012;3:94-101.
21. Stephen B, Galloway SJ, McMillan D, Hill DC, Infield DG. A copula model of wind turbine performance. *IEEE Trans Power Syst*. 2011;26:965-966.
22. Bach-Andersen M, Romer-Odgaard B, Winther O. Scalable systems for early fault detection in wind turbines : a data driven approach. In: European Wind Energy Association Annual Conference and Exhibition; 2015November; Paris, France:382-390.
23. Wang L, Zhang Z, Long H, Xu J, Liu R. Wind turbine gearbox failure identification with deep neural networks. *IEEE Trans Ind Inf*. 2017;13:1360-1368.
24. Ibrahim RK, Tautz-Weinert J, Watson SJ. Neural networks for wind turbine fault detection via current signature analysis. In: WindEurope Summit; 2016September; Hamburg, Germany.
25. Chen H, Lundberg S, Lee S. Hybrid gradient boosting trees and neural networks for forecasting operating room data. *arXiv e-prints*. 2018Jan:arXiv:1801.07384.
26. Lei J, Liu C, Jiang D. Fault diagnosis of wind turbine based on long short-term memory networks. *Renew Energy*. 2019;133(C):422-432.
27. Liu Y, Guan L, Hou C, Han H, Liu Z, Sun Y, Zheng M. Wind power short-term prediction based on LSTM and discrete wavelet transform. *Appl Sci*. 2019;9(6):1108.
28. Lei J, Liu C, Jiang D. Fault diagnosis of wind turbine based on long short-term memory networks. *Renew Energy*. 201810;133.
29. Bengio Y, Frasconi P, Simard P. The problem of learning long-term dependencies in recurrent networks. *IEEE Int Conf Neural Netw*. 1993.
30. Hochreiter S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int J Uncertain Fuzziness Knowl Based Syst*. 1998;06(02):107-116.
31. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9(8):1735-780.
32. Mikolov T, Karafiát M, Burget L, Cernocký J, Khudanpur S. Recurrent neural network based language model. In: INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010; 2010:1045-1048.
33. Sutskever I, Martens J, Hinton G. Generating text with recurrent neural networks. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11) Getoor L., Scheffer T., eds., ICML '11. ACM; 2011June:1017-1024.
34. Graves A. Generating sequences with recurrent neural networks. *CoRR*. 2013;abs/1308.0850. http://arxiv.org/abs/1308.0850
35. Chen T, Guestrin C. Xgboost. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16; 2016.
36. Leventis D, Leventis D. XGBoost Mathematics Explained: Towards Data Science; 2018.
37. 1404033991. A Comprehensive Hands-On Guide to Transfer Learning with Real-World Applications in Deep Learning: Towards Data Science; 2018.
38. Kaboli M. A review of transfer learning algorithms. *Research Report*, Technische Universität München; 2017. Transfer Learning Algorithms.
39. Pan SJ, Yang Q. A survey on transfer learning. *IEEE Trans Knowledge Data Eng*. 2010;22(10):1345-1359.
40. Trivellato F, Battisti L, Miori G. The ideal power curve of small wind turbines from field data. *J Wind Eng Ind Aerodyn*. 2012;107-108:263-273.
41. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. *J Artif Int Res*. 2002;16:321-357.
42. Lemaître G, Nogueira F, Aridas CK. Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning. *J Mach Learn Res*. 2017;18(17):1-5.
43. Hassani H, Silva E. A Kolmogorov-Smirnov based test for comparing the predictive accuracy of two sets of forecasts. *Econometrics*. 2015;3(3):590-609.
44. Wu J. The uniform effect of k-means clustering. *Advances in K-means Clustering Springer Theses*. 2012:17-35.
45. Krishnan M, Krishnan M. Mathematics behind k-mean clustering algorithm; 2018.
46. Whang JJ, Dhillon IS, Gleich DF. Non-exhaustive, overlapping k-means. In: Proceedings of the 2015 SIAM International Conference on Data Mining; 2015.
47. Abadi M, Agarwal A, Barham P, et al. TensorFlow: large-scale machine learning on heterogeneous systems. Software available from tensorflow.org; 2015.
48. Bengio Y. Practical recommendations for gradient-based training of deep architectures. *Lect Notes Comput Sci Neural Netw Tricks Trade*. 2012:437-ĂŞ4.
49. Santos P, Villa Montoya L, Renones A, Bustillo A, Maudes J. An SVM-based solution for fault detection in wind turbines. *Sensors (Basel, Switzerland)*. 2015;15:5627-48.
50. Kingma D, Ba J. Adam: a method for stochastic optimization. *Int Conf Learn Represent*. 201412.
51. Hinton G, Srivastava N, Swersky K. *Overview of mini-batch gradient descent*: University of Toronto.
52. Robbins H, Monro S. A stochastic approximation method. *Ann Math Statist*. 195109;22(3):400-407. https://doi.org/10.1214/aoms/1177729586

53. Kim M-G, Dalhoff PH, Gust P. Yawing characteristics during slippage of the nacelle of a multi MW wind turbine. *J Phys Conf Ser*. 2016;753:062010.
54. Wan S, Cheng L, Sheng X. Effects of yaw error on wind turbine running characteristics based on the equivalent wind speed model. *Energies*. 2015;8:6286-6301.
55. Rotor imbalance cancellation whitepaper (KK Wind Solutions A/S). https://www.kkwindsolutions.com/Files/Files/WhitePapers/20160311_Rotor-Imbalance-Cancellation.pdf Accessed: 2019-05-08.

How to cite this article: Chatterjee J, Dethlefs N. Deep learning with knowledge transfer for explainable anomaly prediction in wind turbines. *Wind Energy*. 2020;23:1693–1710. https://doi.org/10.1002/we.2510