

Article

RETRACTED: Supervisory Control and Data Acquisition for Fault Diagnosis of Wind Turbines via Deep Transfer Learning

Silvio Simani ^{1,*} , Saverio Farsoni ¹  and Paolo Castaldi ² ¹ Department of Engineering, University of Ferrara, 44122 Ferrara, Italy² Department of Electrical, Electronic, and Information Engineering, University of Bologna, 40136 Bologna, Italy

* Correspondence: silvio.simani@unife.it; Tel.: +39-0532-974844

Abstract: The installed wind power capacity is growing worldwide. Remote condition monitoring of wind turbines is employed to achieve higher up-times and lower maintenance costs. Machine learning approaches can be used for detecting developing faults in wind turbines in their earlier occurrence. However, training fault detection models may require large amounts of past and present data. These data are often not available or not representative of the current operation behaviour. These data can be acquired with supervisory control and data acquisition systems. Note also that newly commissioned wind farms lack data from previous operation, whilst older installations may also lack representative working condition data as a result of control software updates or component replacements. After such events, a turbine's operation behaviour can change significantly so its data are no longer representative of its current behaviour. Therefore, this paper shows that cross-turbine transfer learning can improve the accuracy of fault detection models in turbines with scarce data from supervisory control and data acquisition systems. In particular, it highlights that combining the knowledge from turbines with scarce data and turbines with plentiful data enables earlier detection of faults than prior art methods. In this way, the reuse and the knowledge transfer across wind turbines allows us to overcome this lack of data, thus enabling accurate fault detection in wind turbines.

Keywords: condition monitoring; diagnostics; transfer learning; fault detection; convolutional neural networks; supervisory control and data acquisition; wind turbines



Citation: Simani, S.; Farsoni, S.; Castaldi, P. RETRACTED: Supervisory Control and Data Acquisition for Fault Diagnosis of Wind Turbines via Deep Transfer Learning. *Energies* **2023**, *16*, 3644. <https://doi.org/10.3390/en16093644>

Academic Editors: Tariq Kamal and Syed Zulqadar Hassan

Received: 11 February 2023

Revised: 31 March 2023

Accepted: 20 April 2023

Published: 24 April 2023

Retracted: 2 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The globally installed wind power capacity is growing steadily, in line with efforts to increase the share of renewable energy production around the world [1,2]. The operation and maintenance costs of wind farms make up about one third of the electricity production costs over the lifetime of a wind farm [3].

Condition monitoring and intelligent fault detection approaches can enable a transition towards condition-based maintenance of Wind Turbines (WTs), thereby increasing the up-time of the monitored WTs and reducing their maintenance costs. Fault detection algorithms that rely on Machine Learning (ML) are designed to trigger a notification to the operator when unusual operation behaviour is observed. This requires models of the WT's normal operation behaviour which can be trained on condition data from healthy WTs.

The use of condition data from the Supervisory Control And Data Acquisition (SCADA) system was successfully demonstrated to train such Normal Behaviour Models (NBM)s especially for fault detection tasks, as shown in [4–9].

However, this approach requires condition monitoring data to be available over a sufficiently long operation period covering a representative range of operating conditions. Indeed, when condition monitoring data are scarce or when they are no longer representative of the WT's current behaviour, fault detection based on NBMs is usually not an option because NBMs cannot then be trained. This is the case for newly commissioned WTs and

in the initial stage of the operation life of a WT. Moreover, major changes in the operation conditions, regardless of the WT age, can result in a lack of SCADA data representative of the new operation behaviour. This scenario can arise, e.g., after control software updates or hardware retrofits.

Note that usually SCADA data can be acquired for years, and are almost able to represent the entire life of the process. However, these data are not always exploited for fault diagnosis tasks, but only for condition monitoring purposes and sometimes for performing statistical analysis of the fault occurrence. This is especially valid for wind turbine installations, where SCADA systems are normally available and exploited for this task, see, e.g., [10]. Thus, due to the availability of large amount of data for wind turbine plants, this paper proposed the approach developed in the work, which aims at employing the advantages of the availability of massive amount of data sequences for a wind turbine plant. Moreover, the information acquired from this plant is transferred to a process whose working condition data is very limited.

Therefore, the use of Transfer Learning (TL) in connection with Deep Learning (DL) is considered in this work. In fact, DL methods are very common but require huge amounts of data [11]. Therefore, to solve this limitation, details acquired from processes characterised by many data are transferred to plants with scarce data. These data are often available from SCADA systems.

In particular, this study demonstrates the potential of TL for enabling SCADA-based fault detection with NBMs even when NBM training data are scarce. We investigate how NBMs can be transferred from a WT with sufficient training data (source WT) to a WT with scarce operation data (target WT) to achieve accurate NBMs and, thus, reduce fault detection delays. The work investigates multiple TL strategies for training a NBM in the target WT, and studies how they affect the accuracy of the trained NBM. It also analyses how the most preferable strategy depends on the amount of training data available. The proposed approach drastically reduces the amount of SCADA data required to train a NBM for fault detection tasks in the target WT. It is shown that TL results in a higher NBM accuracy and, thus, earlier fault detection when less than a year of training data are available.

It is worth noting that the use of DL is available in many cases [12], for example in computer vision [13], speech recognition [14]. Thus, researchers are involved in fault detection [15,16], as in this paper. With respect to other situations, it is normal to have data from healthy conditions and only a few sequences of the faulty ones [17]. In this paper, a scheme was exploited to extend the knowledge acquired from different WTs to similar WTs. WTs can be similar, but with different Working Conditions (WC)s, as shown in [18,19]. The development of the Internet of Things (IoT) can allow the collection of data from remote and different tools, such as SCADA devices.

Whereas typical ML models are predicated on the idea that both the training and testing data belong to the same data distribution, TL aims to improve learners' performance by transferring information from a related domain. The concept of TL is influenced by how people learn, which involves applying prior knowledge to solve difficulties. For instance, if a person can ride a bike, learning to drive a car will be easier than starting from scratch with no prior driving experience. As will be seen in the paper, TL allows ML models to transfer learned information from source domains to a target domain in order to enhance the effectiveness of the target learning function, even though both the source and target domains have different data distributions [11].

Moreover, it is possible to transmit data samples from the source domain to the target model to enhance learning [20]. There are still significant research gaps that need to be filled, despite the fact that TL is becoming more and more popular and has been used in various sectors, including defect detection.

TL for defect detection will be the sole issue covered in this paper because these are the subjects of this research. The learnt information about defects is sent from a source machine to a target machine in fault detection TL. The methods that are currently accessible only use

one computer as a source [21]. In order to learn features from the source and target domains together, domain adaptation is proposed in the study as a fault detection technique. A support vector machine classifier is then used to predict problems [22]. In order to forecast bearing inner race, ball, and outer race problems under fluctuating working conditions, a TL approach was applied. In order to classify gear pitting flaws, the publication [23] suggested an enhanced deep neural network optimized by a particle swarm optimization methodology and a regularization method.

Although labeling data is a challenging operation when fault data is unbalanced, the information can still be sent to a destination without labeled data. When the labeled data was unavailable for the target, the greatest mean discrepancy was employed to reduce the difference between the source and target domains. Several DL models, such as the sparse autoencoder [24] or the Convolutional Neural Network (CNN) [25], are employed for condition recognition in addition to the domain adaptation utilizing the maximum mean discrepancy. In order to extract transferable features from the unprocessed vibration data, the article suggested a feature-based CNN [26].

Then, to lessen the distribution mismatch of learnt transferable features, multi-layer domain adaptation is implemented, and pseudo label learning is utilized to train from unlabeled target domain samples [27]. Pre-trained networks, on the other hand, can be utilized to train a deep learning network for fault classification, as demonstrated in [28]. The sensor data is converted to picture data by plotting [29] or applying wavelet transformation [30] to generate a time-frequency distribution, which is then used to fine-tune the high-level network layers, and the low-level features are extracted from the pre-trained network. The TL technique to defect detection was confirmed using lab-generated data from a bearing data-set [31].

The existing state-of-the-art approaches do not take into account the fact that there are certain differences between the source and target machines. Disagreements known as negative transfer may impair the target model's performance. Positive transfer, as opposed to negative transfer, is the helpful information from the source, as discussed in [11]. Combining DL and TL allows you to train nonlinear high-dimensional DTL models with a minimal data set. There are four types of TL: instance-based DTL, network-based DTL, mapping-based DTL, and adversarial-based DTL. To overcome the negative transfer problem, the work [11] offered for example an instance-based DTL approach.

On the other hand, with reference to DL applied to fault diagnosis, DL-based fault diagnosis models are relatively widespread, such as, e.g., in the deep variational autoencoder [32,33], multiscale deep belief network [34], hybrid deep learning [35] and stacked denoising autoencoder [36].

The fault detection methods can also be based on the one-class model, such as the One-Class Support Tensor Machine (OCSTM) [37,38], which has been applied extensively in many fields. However, in the case of small-sample training, the OCSTM and other one-class methods estimate the geometric boundary of the training sample very compactly, and the class of the training sample is under-estimated, leading to the problems of low recall rate of normal samples, namely a high false alarm rate and low overall detection accuracy. To address the aforementioned problems, a fault detection scheme based on One-Class Tensor HyperDisk (OCTHD) was also proposed, e.g., in [39].

A few application examples can be also reported here, as the objective of this paper is to propose a methodology in wind power plant fault diagnosis. Several remarks are made regarding the use of SCADA systems in wind turbine power plants. As highlighted in Figure 1, SCADA systems are responsible for controlling and monitoring many of the processes that make life in the industrial world possible, such as power distribution, oil flow, communications, and many more. In this work, an overview of SCADA at the wind power plant is presented, and operational concerns are addressed and examined. Notes on future trends are also provided. Therefore, Figure 1 summarises some important points regarding SCADA systems and their application in the wind power plant environment.

One of the most significant aspects of SCADA is its ability to evolve with the ever-changing face of Information Technology systems.

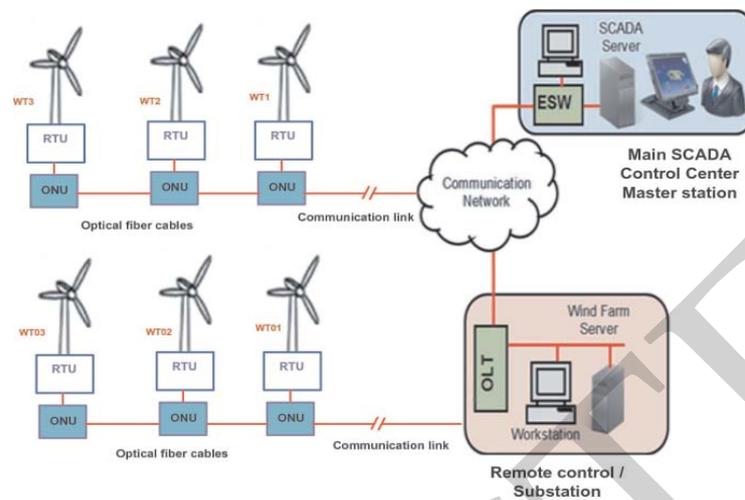


Figure 1. Example of SCADA system application to wind power plant environment.

On the other hand, as depicted in Figure 2, a local control unit is placed at the bottom of each wind turbine, which independently collects data from transducers placed across various components of the turbine. This includes everything from the wind blades to the gearbox and generator. From there, the local control unit performs a periodic health check of all systems, set by the turbine operator. The supervisor tool reports back to the SCADA system, a centralized industrial approach for monitoring and controlling machinery distributed across large factories, mills, farms, etc. Once implemented, operators can view the collected diagnostics from the entire wind farm due to mesh network topology, which enables signals to be transmitted from different devices simultaneously. With the local control unit, operators can view turbine information such as vibration, temperature, load characteristics, force, oil pressure and more. This approach allows operators to take preventative measures prior to become a major problem. On top of detailed insight into each turbine, operators can set alerts to signal when a diagnostic crosses a specified set point. For example, if the operator is alerted of a temperature increase in the generator, someone can quickly fix the immediate issue. Previously, the entire generator may be destroyed before realizing there was an escalating problem.

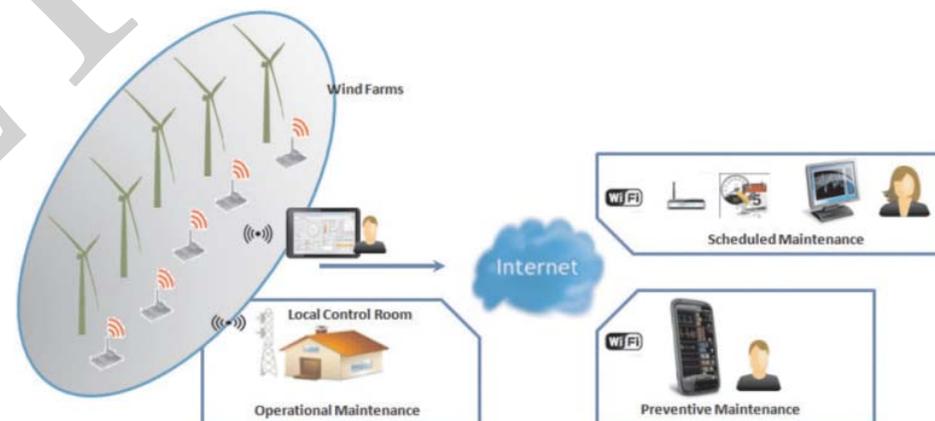


Figure 2. Example of centralised industrial SCADA approach for WT monitoring.

In particular for this research, which combines TL with DL for defect diagnosis, it focuses on TL that allow data to be combined across numerous devices. To cope with big data sets and capture nonlinear trends from diverse measures, an approach such as DL is required rather than shallow machine learning models. TL with DL is used to

transfer gained knowledge from vast fault history WT to scarce fault history WT, which is insufficient to train typical ML models. In fact, in contrast to prior research by the authors [40], this paper employs DL and TL methods for fault diagnostic application to WT SCADA data. Other well-established traditional approaches showing the application of artificial intelligence tools for fault diagnosis can be found in [41–48].

This paper is organized as follows. Section 2 reviews existing TL approaches and how they can be beneficially applied in wind power applications. The first recently-proposed applications of TL to condition monitoring tasks in wind farms are also considered. SCADA-based modelling of the normal operation behaviour and the detection of deviations from that behaviour are some of the most relevant fault detection approaches in WTs in practice [6]. Despite its relevance, the potential of TL for this fault detection task has not been investigated so far. This work demonstrates how TL can be beneficial for extracting operation knowledge from WTs with sufficient training data and for re-using this knowledge to improve the fault detection accuracy in WTs with scarce training data. Section 3 introduces new strategies for training a NBM for fault detection despite scarce target WT training data. Section 4 discusses the results of a case study of the presented strategies and their performances. Finally, Section 6 presents some conclusions and possible directions for future research.

2. Problem Formulation

Fault detection methods for WTs usually assume that the training and the test data originate from the same multivariate distribution in the same feature space. For example, it is usually assumed that a model of the WT's normal operation behaviour, once trained on past operation data, will continue to perform well when applied to data from the future operation of the WT. However, this assumption is often not correct in practice as the distribution of the operation data can evolve over time and even change abruptly. This can be reflected in distribution shifts, such as shifts in the power curve or in other correlated SCADA variables. Various processes can cause these distribution changes, including control software updates and the replacement of hardware. In such situations, SCADA data that are representative of the current normal operation of the WT are scarce. Representative SCADA data are also scarce after the WT commissioning and in the beginning of its operation life. If condition data are barely available, the feature space will be populated only sparsely, so a probability density distribution that is representative of the WT's normal operation behaviour cannot be estimated reliably. For instance, sufficient SCADA data might not be available at high wind speeds, so the power generation cannot be estimated accurately.

On the other hand, after software and hardware updates, SCADA data from before the update are often plentifully available but no longer representative of the current operation behaviour. In this case, the feature space is comprehensively populated but the distribution of the data does not fully reflect the WT's current normal operation behaviour. Nevertheless, if a NBM is trained on the past operation data, it tends to be less capable of detecting anomalous operation behaviour and, thus, causes delays in the detection of incipient faults [9].

TL relies on the mathematical concepts of domain and task. Formally, a domain $\mathcal{D} = \{\mathcal{X}, P(\mathcal{X})\}$ consists of a feature space \mathcal{X} and a marginal probability distribution $P(\mathcal{X})$. A task \mathcal{T} is defined by a label set \mathcal{Y} and a conditional probability distribution $P(\mathcal{Y}|\mathcal{X})$, which can be estimated from a training set $\{(x_i, y_i) | x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$. TL considers a source domain \mathcal{D}_s , a target domain \mathcal{D}_t , a source task \mathcal{T}_s and a target task \mathcal{T}_t , and aims to estimate the conditional probability distribution in the target domain, $P(\mathcal{Y}_t|\mathcal{X}_t)$ from information extracted from the source domain $\mathcal{D}_s = \{\mathcal{X}_s, P(\mathcal{X}_s)\}$ with tasks \mathcal{T}_s and \mathcal{T}_t [49,50].

Thus, it enables the transfer of information learnt on a source feature space and task to a different but related target feature space and task. In this way, it can help to address mismatches in the training and the test data. TL has shown success in situations when the source and target data follow similar but different distributions or populate similar but

different feature spaces; it can also be successful for applications when the source task and the target task are similar but different [50–53].

Such situations arise frequently in machine health applications in industrial fleets. Wind farms comprise different but similar WT types, operating conditions, and SCADA systems. TL has a strong potential for improving WT condition monitoring tasks including performance monitoring, fault diagnostics and machine health state prognostics. This study demonstrates the application of TL for training accurate models of the normal operation behaviour of WTs for fault detection tasks. The representativeness and accuracy of NBMs is of major importance for enabling the early detection of incipient faults [9]. TL strategies can be particularly useful when many training data in the source domain are available but little training data in the target domain, which is the case in the situations of scarce WT condition data considered in this study.

Few studies have investigated the re-use and transfer of knowledge across different WTs. The potential of TL methods in wind energy applications, and notably fault diagnostics tasks, has barely been studied yet. The work [54] proposed to apply a fault detection model across WTs without retraining the model on the target WT. They made use of an offshore source WT and multiple onshore target WTs. Fault labels were available for the source WT and unavailable for the target WT. The model was trained on data from the SCADA system of the source WT. The fault detection model was applied to the target domain without retraining, the entire target domain data-set being used as a test set. The paper [55] addressed methods for image-based blade damage detection in WTs. They applied a TL approach to demonstrate improved feature extraction and damage detection accuracy in the images. The work [56] applied TL to fault diagnosis tasks in WTs. They investigated the application to gear cog belt fractures and to blade icing detection based on SCADA data. The authors compared classical ML methods and TL with an algorithm [49] for the two fault classification tasks, finding superior performance of the TL-based method. The paper [57] presented a fault diagnosis method for WT gearboxes that makes use of generative TL. They demonstrated their approach in the laboratory with accelerometer measurements from the gearbox of a 3 kW WT. The work [58] investigated the transfer of fault diagnosis tasks on SCADA and failure status data-sets. The authors considered different fault types, focused on the transfer of fault diagnostics models, and demonstrated the application of autoencoders to this end. Finally, the work [59] proposed a TL approach based on multiple autoencoders with step-wise customization. They chose the application field of short-term forecasting of wind power and demonstrated their proposed approach in 50 commercial WTs.

Surprisingly, none of the existing TL methods for SCADA-based fault detection involve normal behaviour modelling, even though it is one of the most relevant data-driven methods for early fault detection in WTs in practice. This study investigates TL strategies to fault detection based on NBMs. The research issues that this work aims at solving are as follows:

- How to re-use and transfer operational knowledge across WTs;
- The effect of the transfer on the performance of fault detection models for the target WT;
- The results obtained with a one-fits-all NBM for fault detection at the target WT when applied to a customized model;
- The improvement in the accuracy of SCADA-based fault detection with TL of a NBM with respect to training a NBM from scratch;
- The relation between the accuracy improvement and the amount of training data from the target WT.

3. Transfer Learning for Fault Detection

The task in this study is the SCADA-based detection of incipient faults in WTs whose SCADA data are scarce or no longer representative of the current normal operation behaviour. To this end, the work demonstrates how to train a NBM and transfer it to a target WT, as shown in Figure 3. It is sketched how TL strategies can facilitate the re-use and

transfer of knowledge about the normal operation behaviour of a source WT to a target WT for fault detection tasks.

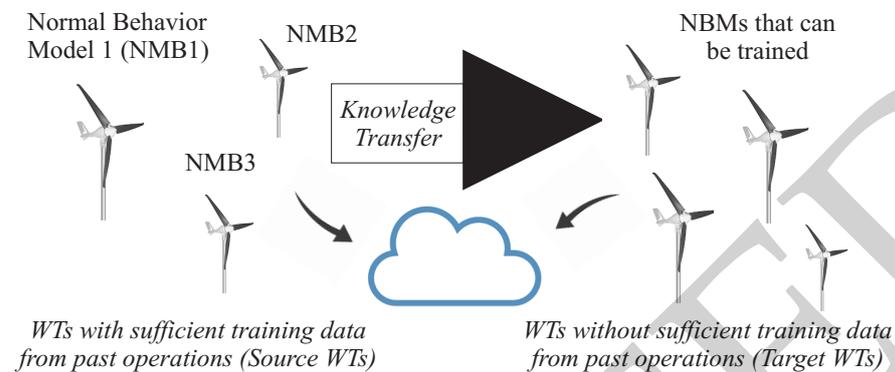


Figure 3. Transfer learning strategy enhancement.

A NBM f is a statistical or ML model that was trained on SCADA features $x_1(t), x_2(t), \dots, x_m(t)$ to learn the distribution of one or several target SCADA variables $y_1(t), y_2(t), \dots, y_n(t)$ to be monitored, as shown in Figure 4. In particular, the flow chart of Figure 4 illustrates the SCADA-based fault detection methods based on training and application of NBMs.

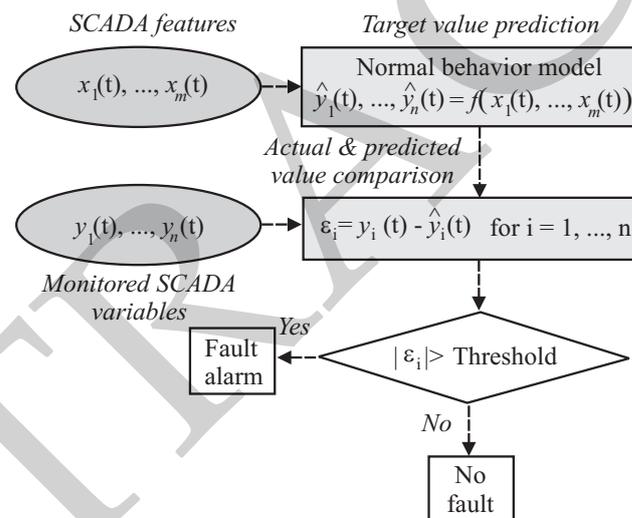


Figure 4. Flow chart of the fault detection strategy.

The model f maps the SCADA features $x_1(t), x_2(t), \dots, x_m(t)$ to the SCADA variable estimates $\hat{y}_1(t), \hat{y}_2(t), \dots, \hat{y}_n(t)$ expected under normal operation behaviour, with $m, n \in \mathbb{N}$. The features $x_1(t), x_2(t), \dots, x_m(t)$ on which f is trained are chosen to reflect the full range of operating conditions that can arise in a fault-free normally-functioning WT and that are relevant for predicting the estimated values of the monitored variables $y_1(t), y_2(t), \dots, y_n(t)$. The SCADA input features typically include environmental conditions, in particular the wind speed. The SCADA target variables to be monitored should be suitable indicators of fault conditions. For example, operation temperatures of critical components and fluids are often selected as the SCADA target variables to be monitored for early fault detection tasks. This is because unusual heat generation, such as from excessive friction, can indicate operating problems and incipient faults.

Note that, as recalled in Figure 4, this paper will not consider the problem of the evaluation of the diagnostic signals $\epsilon_i = y_i - \hat{y}_i$, with $i = 1, \dots, n$ since it focuses on the improvement of the accuracy of the estimates \hat{y}_i . However, it is worth noting that as already remarked, the focus of the paper is on transfer learning approaches. The fault diagnosis

represents a possible application, even if it is not the main point of the work. As proposed in [60], the selection of the fault detection threshold ϵ_i can be based on a simple geometrical test or more complicated statistical schemes. In particular for this study, the fault detection thresholds are settled by means of the empirical solution that considers a margin of $\pm 10\%$ of the minimal and maximal values of the fault-free values of $\epsilon_i = y_i - \hat{y}_i$. However, for improving the fault detection accuracy, more complex approaches may be exploited, as addressed in [60].

The aim of the work is the early detection of faults based on an accurate estimation of the normal operation behaviour of the monitored WT. In a TL sense, the task to be learned is the same on the source and the target data-sets, namely the accurate estimation of the normal operating behaviour to enable an early detection of developing faults based on the deviation from normal behaviour. Observations of actual WT faults are relatively scarce and of diverse types. Therefore, many automated fault detection algorithms rely on evaluating signal reconstruction errors in normal behaviour modelling approaches instead of learning fault signatures [6,7]. Normal behaviour models were shown to successfully detect developing faults in WT drive trains, as in [5]. They constitute one of the most relevant SCADA-based fault detection approaches in practice [6]. Single- and multi-target NBMs have been proposed, with multi-target NBMs being capable of exploiting the covariance among the target variables $y_1(t), y_2(t), \dots, y_n(t)$ which can enable more accurate NBMs and earlier fault detection [8,9]. In practice, a separate WT-specific NBM is usually trained and deployed to operation for every single WT. This is performed to account for site-specific aspects, such as wake effects, on the WT operation, to achieve a high NBM monitoring accuracy and, thus, enable an earlier detection of developing faults. Training a NBM requires the availability of training data $\{x_1(t), x_2(t), \dots, x_m(t), y_1(t), y_2(t), \dots, y_n(t)\}$ that are representative of the current normal operation behaviour of the WT under all operation conditions. In practice, ideally at least one year of SCADA data are used for training a NBM. However, so much representative training data are not always available. Newly commissioned WTs and WTs in their first year of operation naturally lack these training data. Even older WTs may be scarce in representative SCADA data that can be used for training a NBM. For example, this may be the case after control software or hardware updates.

To overcome the scarcity of training data, this work proposes to exploit the information contained in the NBM of a WT abundant in training data (source WT) and to transfer that information to the WT lacking representative training data (target WT), as remarked in Figure 3.

On the other hand, Figure 5 summarises the different stages exploited for the deployment and the development of the proposed ML and DL strategies for WT SCADA data.

Different computational strategies will be analysed for this knowledge transfer in the following. The NBMs trained in this study are convolutional neural networks as detailed below. As shown in Figures 6–10, five different transfer strategies are considered.

In particular, three different transfer strategies (strategies 1–3) are compared to two strategies in which a NBM is trained from scratch at least in part on SCADA data from the target WT (strategies 4 and 5).

Therefore, according to transfer strategy 1 of Figure 6, the model is trained on the training set of the source WT. The scarce training data of the target WT are used to retrain only the last layers of the model. The network weights of the other layers are transferred to the target WT without any changes. In more detail, the NBM has been trained and optimised on the source WT training and validation sets. The NBM for the target WT was initialised with the weights of the CNN trained on the source WT. Thus, the initial state of the target NBM already comprised knowledge about the normal operation behaviour of a WT and the covariance among the input and target variables even before the training on the target WT's training data started. Next, the weights of the two last layers of the NBM were retrained on the 24 h sequences of the training set of the target WT. The last two layers were the two fully connected layers with eight neurons in the hidden layer and seven neurons in the output layer, as described above.

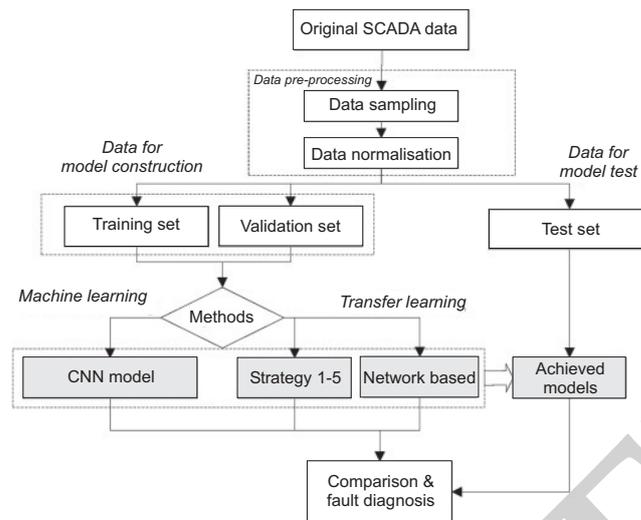


Figure 5. Block diagram with the summary of the proposed strategy and its stages.

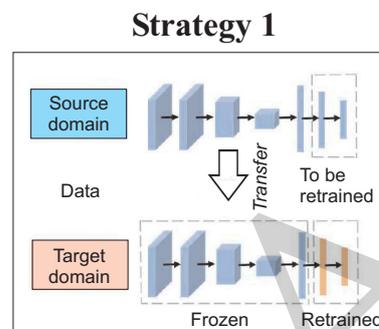


Figure 6. Strategy 1 for training a NMB for fault detection tasks on a target WT.

Strategy 2 in Figure 7 is the same as strategy 1, except that all the layers of the network are retrained on the target WT data. In particular for this strategy, it requires a NBM to be trained and optimised on the source WT training and validation sets followed by initializing the target WT's NBM with the thus trained CNN and its weights. Unlike strategy 1, however, all weights of all layers have been retrained on the target WT's training set.

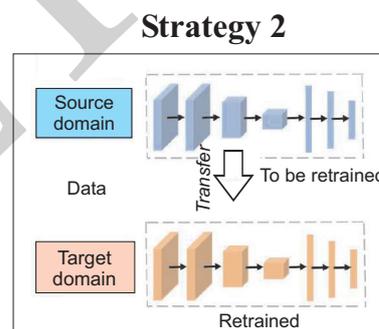


Figure 7. Strategy 2 for training a NMB for fault detection tasks on a target WT.

Strategy 3 of Figure 8 involves training the model on the source WT and transferring and applying the trained model to the target WT data without any modifications. In more detail, a NBM has been trained on the source WT's data as discussed above. The trained model has been then applied to the test data of the target WT without any retraining or any other customization to the target WT. So no condition monitoring data from the target WT were used as part of the training or validation according to strategy 3.

Strategy 3

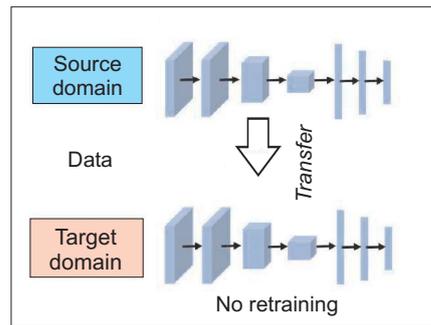


Figure 8. Strategy 3 for training a NMB for fault detection tasks on a target WT.

Strategy 4 of Figure 9 shows that the NBM is trained from scratch on the merged SCADA training data-sets from the source and the target WTs. In particular, this strategy has provided that the SCADA data of the source and the target WTs have been combined prior to the NBM training. A CNN has been trained on the merged data-set of source and target WT SCADA data. The trained NBM has then been tested on the target WT.

Strategy 4

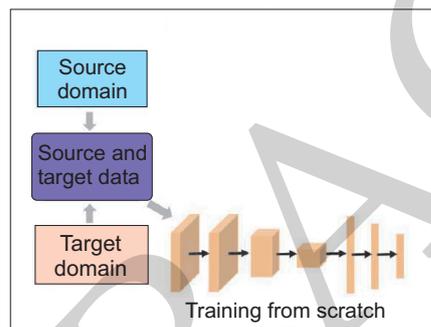


Figure 9. Strategy 4 for training a NMB for fault detection tasks on a target WT.

Finally, strategy 5 of Figure 10 involves training a NBM from scratch on the scarce training data-set of the target WT and not to make use of the NBM or of the training data from the source WT. In more detail for this strategy, no SCADA data or other information from the source WT have been used in the training of the target WT's NBM. Thus, training, validation and testing have been performed from scratch on the SCADA data of the target WT. The performance of each strategy has been measured in terms of the Root Mean Square Error (RMSE) of the trained NBM on the target WT test-set.

Strategy 5

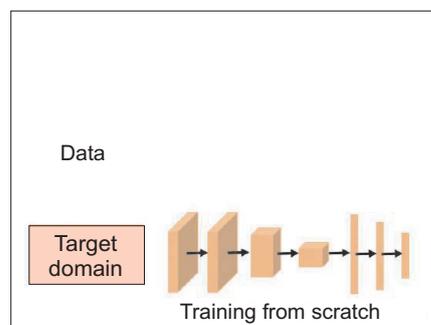


Figure 10. Strategy 5 for training a NMB for fault detection tasks on a target WT.

4. Simulation Results

SCADA data from six 3 MW onshore WT's were used for a case study in which performances of the proposed TL strategies are also compared. Eight combinations of source and target WT's were randomly selected from the six available WT's. The WT's are commercially operated horizontal-axis variable-speed machines with pitch regulation. Fourteen months of SCADA data were available from each WT. The SCADA data were provided at 10 min resolution and computed as mean values over periods of ten minutes. The location and observation time of the WT's are not disclosed to ensure the privacy of the operator.

The SCADA target variables $y_1(t), y_2(t), \dots, y_7(t)$ to be monitored in the case study considered in this work are provided in Tables 1 and 2 along with the SCADA features $x_1(t), x_2(t), x_3(t)$ to estimate them.

Table 1. SCADA features.

SCADA Features	$x_i(t)$
Wind speed	$x_1(t)$
Wind direction	$x_2(t)$
Air temperature	$x_3(t)$

Table 2. SCADA monitored target values.

Monitored Target Variables	$y_i(t)$
Gear bearing temperature	$y_1(t)$
Hydraulic oil temperature	$y_2(t)$
Transformer temperature	$y_3(t)$
Active power generation	$y_4(t)$
Rotor speed	$y_5(t)$
Generator speed	$y_6(t)$
Current	$y_7(t)$

In particular, Table 2 reports the target variables $y_i(t)$ to be monitored and the features $x_i(t)$ used for estimating the NBM based on which fault detection is performed. The features comprise the 10 min mean wind speeds and wind directions measured at the nacelle and the ambient air temperatures. The wind speed is the main predictor of the target variables. Wind speed and air temperature are also relevant features to account for the environmental conditions, including wake effects and air pressure changes. The ambient temperature serves as an imperfect proxy of the air pressure in this study because air pressure measurements were not available from the WT's.

It should be noted that the experimental data represent a large SCADA database of the six WT's investigated in this work. As previously stated, these data are generated by the WT's SCADA and gathered using an Open Platform Communications (OPC) protocol in accordance with IEC 61400-25 standards [61]. As a result, the data are formatted to include WT status, which is represented by logical variables, as well as the obtained measurements of physical systems or subsystems. Events as well as statistical indicators are documented. Tables 1 and 2 summarize the sensor store values. In addition, the minimum, maximum, mean, and standard deviation values are obtained. As a result, the database includes equivalent variables from the various sensors. The events in the database were initially labeled with '0' to indicate normal operation and '1' to indicate warning or alert states (the turbine must be diagnosed or stopped in this instance). Alarms are quite rare in any wind turbine databases because the turbines are usually operational. An alarm may be generated if a warning is not appropriately checked and addressed. As a result, warning and alarm circumstances are merged in the database, reducing the problem to a two-class scenario (normal operation and fault condition). The purpose is to predict when a WT will begin working with possible issues, resulting in a warning or alarm condition. The variables

in Tables 1 and 2 are used in the experiments. Smartive (<http://smartive.eu>, accessed on 10 February 2023) supplied the database, which was used in other publications [62,63]. Because of the true nature of the data, the collection of SCADA variables has significantly distinct dynamic ranges between them. As a result, a pre-processing phase is required. To accomplish this, the data is normalised by subtracting the mean value from each variable and dividing the result by the standard deviation of that variable. The values of means and standard deviations are saved since they are required to perform further denormalisations of the data and proceed with the defect identification task, which will not be covered in this work.

Once the SCADA data are normalised, this ensures that all features have a similar order of magnitude and to enable a fast model training. Then, the data of the source and target WTs were each split into training, validation and test sets, with 20% of both data-sets used as the respective test sets and 10% as validation sets. The same split was applied for training and testing all five strategies and models compared in this study.

It is worth observing also that the SCADA features $x_i(t)$, such as wind speed and air temperature reported in Tables 1 and 2 are inherently included in the target data, since the wind turbine control system regulates the wind turbine outputs by monitoring these variables, as described in [64].

The SCADA features served as input to train a NBM to provide estimates $\hat{y}_1(t), \hat{y}_2(t), \dots, \hat{y}_7(t)$ of the target variables. A multi-target Convolutional Neural Network (CNN) model f was trained to this end. The WTs' normal operation behaviour was modelled with a CNN to capture the time dependence of the WT operation patterns. Sequences with a length of 144 time steps corresponding to a 24 h period were created from the normalised time series as input to the CNN. Thus, the CNN predicts the target values based on a sequence of past operation states rather than on same-time observations only. The input data corresponded to 144 time steps by three input variables and a moving window over the past 144 time steps. The CNN was trained on the resulting sequences of the source WT's training set to compare strategies 1–5, as shown in Figures 6–10.

CNNs are artificial neural networks which comprise layers from which feature maps are compiled by filters moving over the SCADA sequences with a convolution operation [12,65]. Note that the architecture of the CNN was defined as part of a hyperparameter optimisation process in which a grid search was performed with regard to the number of layers and layer sizes that resulted in the lowest validation set error. Architectures with up to two convolutional and two dense hidden layers with different numbers of neurons were compared to this end. A number of 4, 8 and 16 neurons were assessed in each hidden layer, and convolutional filter sizes of 2 and 3 in the optimization process. The mean squared error was chosen as a loss function for training the CNN and to compare the models' performances. The resulting CNN architecture is composed of a convolutional layer with 16 convolutional filters followed by a max pooling layer and a second convolutional layer with 16 filters. The features thus extracted were passed to a dense hidden layer with 8 neurons connected to a dense output layer with seven neurons corresponding to the seven target variables. The CNN was trained and optimized on the training and validation sets of the source WT to establish a NBM of the source WT. The Adaptive Moment Estimation (Adam) optimiser was employed for stochastic gradient descent [66]. The training was conducted over 100 epochs with a batch size of 256 data.

The following TL strategies in this case study in accordance with Figures 6–10 were compared.

For strategy 1, the CNN NBM was trained and optimised on the source WT training and validation sets as discussed above. The CNN NBM for the target WT was initialised with the weights of the CNN trained on the source WT. Thus, the initial state of the target NBM already comprised knowledge about the normal operation behaviour of a WT and the covariance among the input and target variables even before the training on the target WT's training data started. Next, the weights of the two last layers of the CNN were retrained on the 24 h sequences of the training set of the target WT. The last two layers were the

two fully connected layers of the CNN with eight neurons in the hidden layer and seven neurons in the output layer, as described above. The training was again performed with 100 training epochs and a batch size of 256.

Strategy 2 involved a CNN NBM to be trained and optimised on the source WT training and validation sets followed by initializing the target WT's NBM with the thus trained CNN and its weights. Unlike strategy 1, however, all weights of all layers were retrained on the target WT's training set.

According to strategy 3, a CNN was trained as the NBM on the source WT's data as discussed above. The trained model was then applied to the test data of the target WT without any retraining or any other customization to the target WT. So no condition monitoring data from the target WT were used as part of the training or validation according to strategy 3.

Strategy 4 provided that the SCADA data of the source and the target WTs were combined prior to the NBM training. A CNN was trained on the merged data-set of source and target WT SCADA data. The trained NBM was then tested on the target WT.

According to strategy 5, no SCADA data or other information from the source WT were used in the training of the target WT's NBM. Thus, training, validation and testing were performed from scratch on the SCADA data of the target WT. The performance of each strategy was measured in terms of RMSE of the trained NBM on the target WT test set.

An issue of particular interest was how the five strategies compare for various amounts of training data from the target WT. The performances of the five strategies were compared for a range of scarce training set sizes of the target WT in the case study. To this end, a randomly selected source WT was selected from among the six WTs and the transfer to four of the other WTs was investigated. Each of these target WTs was randomly selected in turn to serve as the respective target WT.

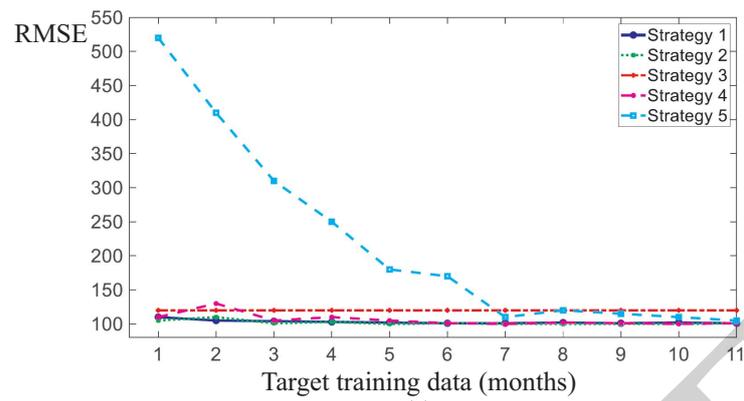
Figures 11a,b and 12a,b show how the performance of the NBMs trained in accordance with each of the five strategies vary with the size of the training data-set of the target WT. Training set sizes of the target WT between 0.3 and 11 months were investigated.

In particular, Figures 11a,b and 12a,b summarise the accuracy of the NBM trained for a randomly selected source WT and four randomly selected target WTs for varying amounts of target WT training data. Each graph of Figures 11a,b and 12a,b corresponds to a different target WT.

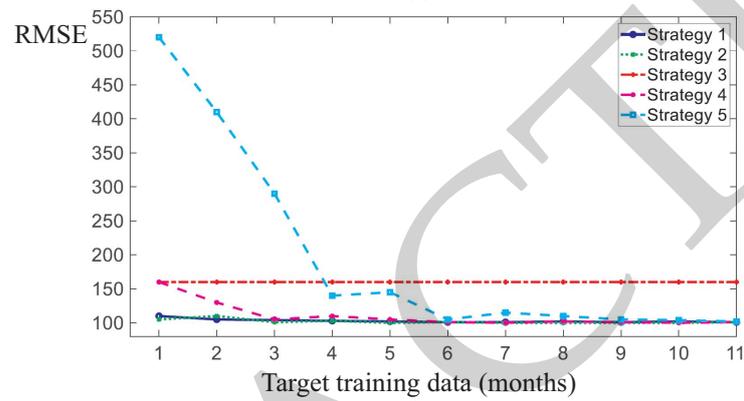
Subsequently, this process was repeated by randomly selecting another source WT and four target WTs from among the six WTs, and then training the NBMs for varying amounts of target training data. The accuracy of the NBMs resulting from the five learning strategies are shown in Figure 13a,b, as well as Figure 14a,b for target WT training set sizes of between 0.3 and 11 months.

In particular, Figure 13a,b, as well as Figure 14a,b depict the accuracy of the NBMs trained for another randomly selected source WT, which are different from the ones in Figures 11 and 12. In this case, four randomly selected target WTs for varying amounts of target WT training data. Each subfigure corresponds to a different target WT.

A separate CNN NBM was trained for each strategy and for each of the considered training set sizes. Random sampling of the source and target WTs were applied to assess eight combinations of source and target WTs instead of evaluating all 30 possible combinations. The sampling of eight out of 30 pairs of source and target WTs was performed due to the significant computational cost of the analysis. Each subfigure of Figures 11–14 amounted to training, validation and testing 55 separate CNNs, which corresponded to several hours of run time on a standard medium performance PC.

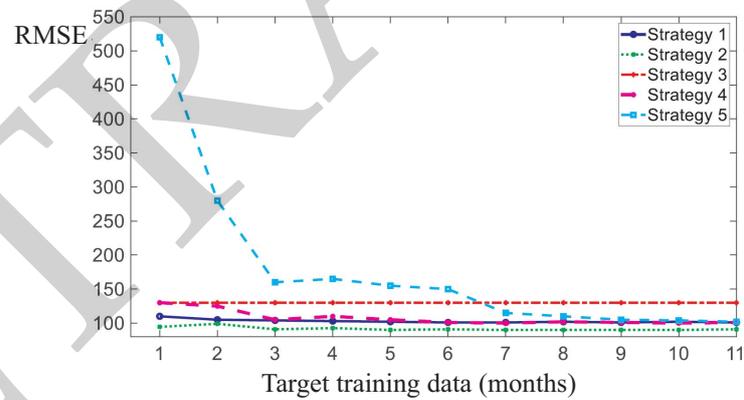


(a)

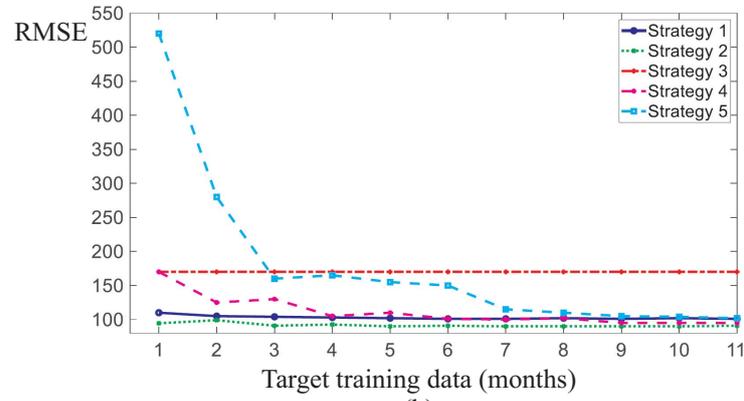


(b)

Figure 11. NBM accuracy for two of four randomly selected source WTs.

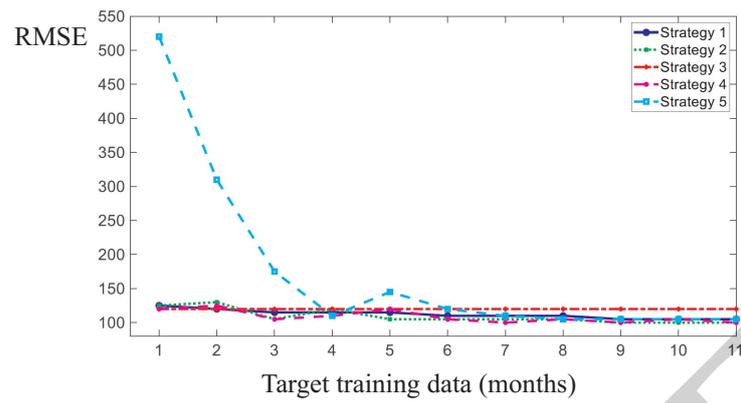


(a)

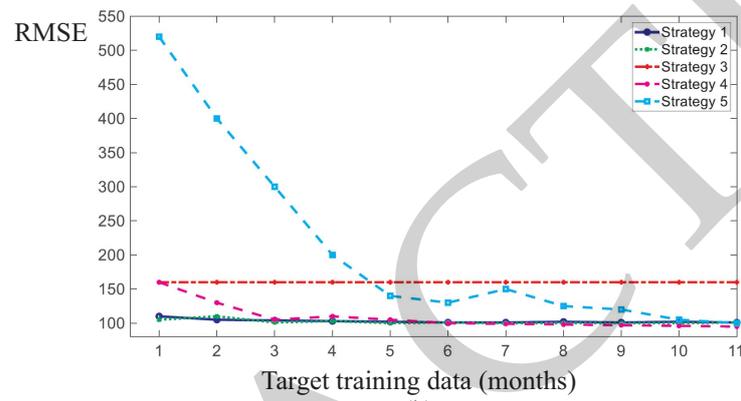


(b)

Figure 12. NBM accuracy for two of four randomly selected source WTs.

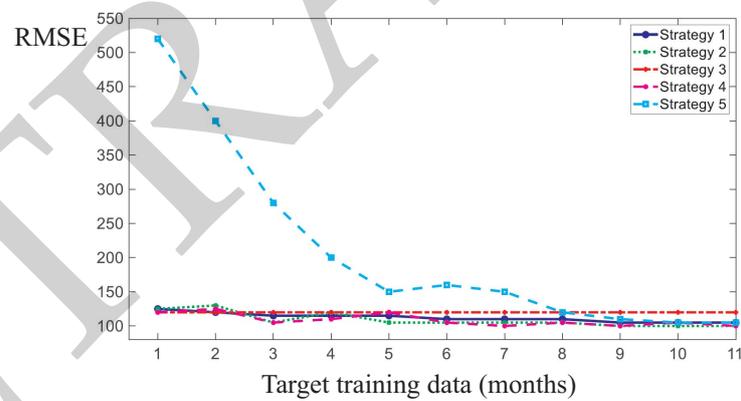


(a)

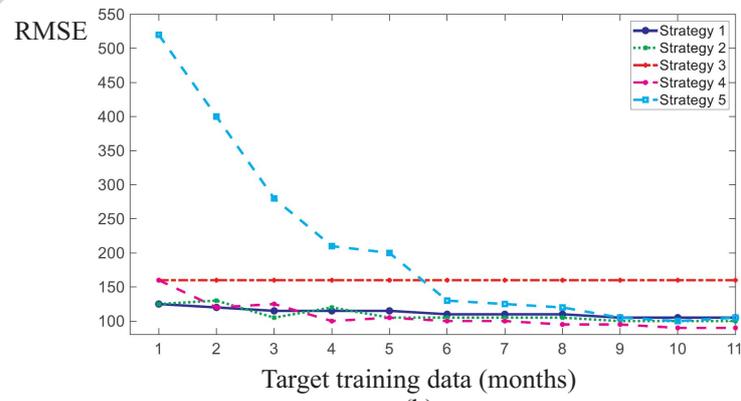


(b)

Figure 13. NBM accuracy for two of four different randomly selected source WTs.



(a)



(b)

Figure 14. NBM accuracy for two of four different randomly selected source WTs.

In order to summarise the overall performance of the proposed strategies, Table 3 reports the accuracy of the considered algorithms, when their modelling capabilities are considered. The modelling accuracy is evaluated as mean relative percentage error between the measured and reconstructed outputs. The standard deviation is also indicated.

Table 3. Accuracy and standard deviation values of the algorithms.

Modelling Method	Modelling Accuracy	Standard Deviation
Strategy 1	99.89%	$\pm 0.27\%$
Strategy 2	98.78%	$\pm 0.34\%$
Strategy 3	98.05%	$\pm 0.66\%$
Strategy 4	97.68%	$\pm 0.62\%$
Strategy 5	96.33%	$\pm 1.39\%$

The proposed solutions were also verified and validated by considering more traditional (without TL) or different approaches (with TL) available in the related literature, see, e.g., [11]. Therefore, to compare the performance of the strategies proposed in this work and identify the effect of advantages and drawbacks with respect to different schemes, a Classic DL (CDL) method and Network-Based TL approach (NBTL) have been implemented. The NBTL method is inspired by the procedure proposed in [22], where trained weights of the source Deep Neural Network (DNN) model are passed to the target DNN model. The used DL method is the baseline traditional ML method without TL and DNN trained on target machine data without any prior knowledge. Furthermore, the state-of-the-art TL methods were validated on data-sets with a limited number of source domain training samples because they were lacking the ability to avoid the negative transfer. Moreover, to validate the proposed approaches on the state-of-the-art methods (CDL and NBTL) and demonstrate the true potential of the proposed strategies, experiments with scarce training samples have been considered.

Additionally, in this case, Table 4 summarises the performance of the considered schemes when the modelling capabilities are evaluated. For each approach, the modelling accuracy provided by the different DL and TL methodologies is computed in terms of mean relative percentage error between measured and reconstructed outputs. The standard deviation is also evaluated.

Table 4. Modelling accuracy and standard deviation values of the considered solutions.

Modelling Method	Modelling Accuracy	Standard Deviation
CDL	82.35%	$\pm 1.04\%$
NBTL	91.56%	$\pm 0.77\%$
Strategy 1	99.24%	$\pm 0.19\%$
Strategy 2	98.34%	$\pm 0.20\%$
Strategy 3	97.65%	$\pm 0.57\%$
Strategy 4	96.60%	$\pm 0.51\%$
Strategy 5	94.48%	$\pm 1.20\%$

The results of all seven methods are summarised in Table 4, when no TL occurs (CDL) or if the knowledge is transferred from scarce source samples (NBTL and strategies 1–5). The detailed results for all methods are thus provided in Table 4. In particular, the mean and the standard deviation values are calculated over 100 repeated experiments, and the improvement is observed with the strategies 1–5, with respect to CTL and DBTL. The CTL method observes notable deterioration, whilst NBTL shows quite consistent results with strategy 5. Therefore, the improvement from the base case CDL method is quite interesting when compared with the NBTL method, but significant if strategies 1–5 are also considered. It can be concluded that the results of this experimental analysis indicate an excellent performance of the proposed strategies 1–5. They outperform conventional DL and more complex DL approaches, such as NBTL.

Finally, with reference to the algorithmic complexity, it depends on the complexity of the CNN models. Given the high variability within each class and high similarity between certain data classes, an appropriately large number of convolutional layers and fully connected dense layers are needed to be able to accurately distinguish between the four classes. The model consists of a series of alternating 2D convolutional layers and max-pooling layers. The final two layers consist of fully connected dense layers. The first dense layer consists of 64 dimensions and the second layer consists of four dimensions representing the four classes. The highest value of the final dense layer, which consists of a 1D array of length 4, indicates the prediction of that model for given pattern and target data.

The models were trained for about 200 epochs with a learning rate of 0.0005. Slow learning was preferred in these models due to their complexity. Slow learning also allowed for size constriction within the models, as previous attempts in the training process with a faster learning rate required significantly larger models to achieve comparable results.

Using smaller models was thus found to be more resource, memory, and time-intensive. The resulting model consisted of 60,772 trainable parameters, which is 99.78% smaller than the average number of parameters of the models with TL.

5. Final Discussion and Remarks

The results show that the TL schemes (strategies 1 and 2) drastically improved the accuracy of the NBMs for fault detection tasks with scarce target training data when compared to strategies 3 and 5. The latter ones are referred as naive strategies in the sense that they only use either the operation knowledge of the source WT or the scarce operation knowledge of the target WT. Strategy 3 involves training a NBM on the source WT and applying that model to the target WT without any modifications. Strategy 3 can be considered a simple TL strategy because the NBM is directly transferred and applied to the target WT without any customisation. This strategy provides a constant NBM accuracy on the target WT as the model training does not involve any SCADA data from the target WT. Strategy 5 provides that a NBM of the target WT is to be trained from scratch on the scarce training data of the target WT.

As highlighted in Figures 11–13, this strategy performs by far the worst among all investigated strategies in case of scarce target WT data. It results in the lowest NBM accuracy on the target WT test set. The scarcer the target training data, the worse the performance of strategy 5 in absolute and relative terms compared to the other strategies. With increasing amounts of target WT training data, the performance of strategy 5 starts to approach the performances of the other four strategies and to converge with them at about 8–10 months of target WT training data. At the same time, applying the source WT model without modifications (strategy 3) tends to be outperformed by the other four strategies at these target WT training data sizes. This is because strategy 3 is the only strategy that does not consider any operational data from the target WT in the target NBM training. This reduces its performance relative to the other strategies when more and more target WT training data become available.

The study also highlighted that strategies which make use of the combined operational knowledge from both the source and target WTs outperform the naive strategies 3 and 5. The performance of TL strategies 1 and 2 and of strategy 4 tends to increase with larger amounts of target WT training data, as expected. If training data are scarce, the naive strategies 3 and 5 result in lower NBM accuracy and, therefore, in larger fault detection delays in the target WT than the other strategies. The latter also include strategy 4 which requires the NBM training to be performed on the combined training sets of the source and target WTs. Strategy 4 achieves similar NBM accuracy and, thus, similar fault detection delays as strategies 1 and 2, and thereby clearly outperforms the naive strategies. Transfer learning strategies 1 and 2 perform very similarly, with strategy 1 (retraining the last layers only) being moderately outperformed by strategy 2 (retraining all layers) in the case of more dissimilar source and target WTs as measured by the accuracy of the NBM trained

following strategy 3. Strategies 3 and 4 converge towards identical training data-sets and, thus, the same degree of accuracy as the size of the target WT training set is decreased.

The main research issues investigated in this work concern the following issues. The re-use and the transfer operational knowledge across WTs. The effect of this procedure on the performance of fault detection models with respect to the target WT. The improvement of the accuracy provided by TL for the case of SCADA-based fault detection when scarce target WT training data compared to training from scratch, and if so, the varying of the accuracy improvement with respect to the size of the target WT training set. The effect of one-fits-all NBM used for fault detection relatively to a NBM customized to the target WT. These topics were analysed by considering five knowledge transfer strategies and investigated in six commercial multi-MW WTs.

The major advantages of TL that were demonstrated in other fields include its ability to enable the training of models for the target system even if few or no training data are available for that system [52]. Hence, less training data are required and more accurate model predictions can be given with the scarce training data. The work confirmed these advantages in the performed case study. Comparing the performances of our five proposed strategies, it was found that the TL strategies 1 and 2 and strategy 4 which involves training on the combined source and target data outperform the naive strategies 3 and 5. Thus, strategies that make use of the combined operational knowledge available from both the source and the target WTs outperform the naive strategies in case of scarce training data of the target WT. Strategies 3 and 5 make use of training data from only the source or the target WT, which results in less accurate normal behaviour models and larger fault detection delays when target WT training data are scarce. It was demonstrated that operators and asset managers can overcome a lack in representative training data by re-using and transferring operational knowledge across WTs by adopting the proposed TL strategies 1 and 2 or strategy 4. Therefore, knowledge transfer across WTs can drastically improve the accuracy of SCADA-based normal behaviour models and fault detection, thus, reducing fault detection delays in the case of scarce target WT training data. This improvement is particularly pronounced when compared to training a NBM from scratch on the scarce target WT training data. In fact, the scarcer the target WT training data, the more strongly strategies 1, 2 and 4 outperform the training from scratch on the target WT (strategy 5). NBMs trained with strategies 1, 2 or 4 also tend to outperform one-fits-all NBMs trained on the source WT and applied to the target WT without modifications (strategy 3).

Even though SCADA data from the source WT tend to provide only an imperfect description of the NBM of the target WT, they are plentifully available and can be used to train a more accurate NBM for the target WT in case of scarce training data. The results demonstrated that even small amounts of target WT training data can be applied beneficially to fine-tune and improve the performance of the NBM of the target WT. It was also found that a NBM should be customized for each individual WT. A one-fits-all NBM for all WTs in the wind farm (strategy 3) and training from scratch despite lack of training data (strategy 5) are the worst performing ones of the five compared strategies. The developed knowledge transfer strategies facilitate the early fault detection in WTs with scarce SCADA data. Thereby, they help to improve the up-time of WT fleets by facilitating earlier and more informed responses to unforeseen maintenance actions, in particular for target WTs with limited availability of past condition monitoring data.

Finally, it is worth highlighting the overall key achievements and results gained from this study. Implementing machine learning and deep learning algorithms for WT FD based on SCADA data can represent a relevant opportunity to reduce the operation and maintenance costs of wind farms. The development of practically implementable algorithms requires addressing the issue of their scalability to large wind farms. Two of the main challenges here are reducing the training times and enabling training with scarce or limited data. Both of these challenges can be addressed with the help of TL methods, in which a base model is trained on a source WT and the learned knowledge is transferred to a target WT. This paper has suggest different TL frameworks also designed

to transfer a FD task between turbines. As a reference model, DL structures have been proven to perform well on the single turbine FD task. The TL frameworks have been tested for transfer between WTs from the same farm, and possibly from different farms. It is concluded that for the purpose of scaling up training for large farms, a simple TL approach basically representing a transformation of the target predictions is an attractive high performance solution. For the challenging task of TL based on scarce target data, it is shown that the considered TL frameworks can outperform other methods. The scheme can be also extended in order to enable the evaluation of different TL frameworks for FD without the need for labeled faults, even if this point requires further investigation.

6. Conclusions

Data-driven fault detection in wind turbines requires sensor data that are representative of the past and present normal operation behaviour of the monitored turbines. Such data are often unavailable, for example, after the commissioning and in the first months of operation of the turbines, but also after major turbine updates. Fault detection models can barely be trained under these conditions. To overcome these limitations, the paper proposed and demonstrated the re-use and transfer of knowledge about normal operation behaviour from turbines with sufficient training data to turbines with scarce samples. Therefore, the paper showed that transfer learning between wind turbines from the same wind farm was possible via a simple and computationally efficient approach using deep learning. This framework could be therefore used to scale up the training of large wind farms by training on one source wind turbine only instead of individually on each turbine. Moreover, the developed strategy was able to outperform other frameworks including different deep learning schemes. On the other hand, these algorithms outperformed training from scratch with limited data of the target turbine. In this way, the suggested methodology could also be exploited for evaluating and comparing if transfer learning approaches is possible even in the absence of labelled faults, if the reference or normal behaviour model for fault detection was defined. It was thus concluded that transfer learning algorithms enabled scalable and reliable fault detection for wind turbines based exclusively on supervisory control and data acquisition devices. Future studies will investigate transfer learning strategies for varying degrees of operational similarity between machines of the same fleet and across different fleets. Moreover, transfer learning strategies might be employed to reduce the amount of data storage needed for condition monitoring tasks. Future studies should also investigate this use case in more detail. Another important open problem that will be pursued in the future is the challenge of optimizing the goodness of transfer by an appropriate selection of the source turbine. The results seem very encouraging, but determining how to select the optimal source wind turbines for the knowledge transfer of machine health monitoring tasks will be also investigated.

Author Contributions: Conceptualization, S.S. and S.F.; methodology, S.S.; software, S.S.; validation, S.S.; formal analysis, S.S.; investigation, S.S.; data curation, P.C.; writing—original draft preparation, S.S.; writing—review and editing, S.S.; supervision, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data for experiments are available from the Smartive website: <http://smartive.eu>, accessed on 10 February 2023.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Frank, S.; Gusti, M.; Havlík, P.; Lauri, P.; DiFulvio, F.; Forsell, N.; Hasegawa, T.; Krisztin, T.; Palazzo, A.; Valin, H. Land-based climate change mitigation potentials within the agenda for sustainable development. *Environ. Res. Lett.* **2021**, *16*, 024006. [[CrossRef](#)]
2. Liu, H.; Khan, I.; Zakari, A.; Alharthi, M. Roles of trilemma in the world energy sector and transition towards sustainable energy: A study of economic growth and the environment. *Energy Policy* **2022**, *170*, 113238. [[CrossRef](#)]

3. Peinado Gonzalo, A.; Benmessaoud, T.; Entezami, M.; Garcia Marquez, F.P. Optimal maintenance management of offshore wind turbines by minimizing the costs. *Sustain. Energy Technol. Assess.* **2022**, *52*, 102230. [[CrossRef](#)]
4. Zaher, A.; McArthur, S.; Infield, D.; Patel, Y. Online wind turbine fault detection through automated SCADA data analysis. *Wind Energy* **2009**, *12*, 574–593. [[CrossRef](#)]
5. Schlechtingen, M.; Santos, I.F.; Achiche, S. Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 1: System description. *Appl. Soft Comput.* **2013**, *13*, 259–270. [[CrossRef](#)]
6. Tautz-Weinert, J.; Watson, S.J. Using SCADA data for wind turbine condition monitoring—A review. *Iet Renew. Power Gener.* **2017**, *11*, 382–394. [[CrossRef](#)]
7. Stetco, A.; Dinmohammadi, F.; Zhao, X.; Robu, V.; Flynn, D.; Barnes, M.; Keane, J.; Nenadic, G. Machine learning methods for wind turbine condition monitoring: A review. *Renew. Energy* **2019**, *133*, 620–635. [[CrossRef](#)]
8. Meyer, A. Multi-target normal behaviour models for wind farm condition monitoring. *Appl. Energy* **2021**, *300*, 117342. [[CrossRef](#)]
9. Meyer, A. Early Fault Detection with Multi-Target Neural Networks. In Proceedings of the Computational Science and Its Applications—CCSA 2021: 21st International Conference, Cagliari, Italy, 13–16 September 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 429–437. [[CrossRef](#)]
10. Maldonado-Correa, J.; Martin-Martinez, S.; Artigao, E.; Gomez-Lazaro, E. Using SCADA Data for Wind Turbine Condition Monitoring: A Systematic Literature Review. *Energies* **2020**, *13*, 3132. [[CrossRef](#)]
11. Jamil, F.; Verstraeten, T.; Nowe, A.; Peeters, C.; Helsen, J. A deep boosted transfer learning method for wind turbine gearbox fault detection. *Renew. Energy* **2022**, *197*, 331–341. [[CrossRef](#)]
12. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
13. Gavali, P.; Banu, J.S. Deep Convolutional Neural Network for Image Classification on CUDA Platform. In *Deep Learning and Parallel Computing Environment for Bioengineering Systems*; Sangaiah, A.K., Ed.; Academic Press: Cambridge, MA, USA, 2019; pp. 99–122. [[CrossRef](#)]
14. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [[CrossRef](#)]
15. Lei, Y.; Jia, F.; Lin, J.; Xing, S.; Ding, S.X. An Intelligent Fault Diagnosis Method Using Unsupervised Feature Learning Towards Mechanical Big Data. *IEEE Trans. Ind. Electron.* **2016**, *63*, 3137–3147. [[CrossRef](#)]
16. Zhang, W.; Li, C.; Peng, G.; Chen, Y.; Zhang, Z. A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mech. Syst. Signal Process.* **2018**, *100*, 439–453. [[CrossRef](#)]
17. Chen, L.; Xu, G.; Zhang, Q.; Zhang, X. Learning deep representation of imbalanced SCADA data for fault detection of wind turbines. *Measurement* **2019**, *139*, 370–379. [[CrossRef](#)]
18. Verstraeten, T.; Nowe, A.; Keller, J.; Guo, Y.; Sheng, S.; Helsen, J. Fleetwide data-enabled reliability improvement of wind turbines. *Renew. Sustain. Energy Rev.* **2019**, *109*, 428–437. [[CrossRef](#)]
19. Li, C.; Zhang, S.; Qin, Y.; Estupinan, E. A systematic review of deep transfer learning for machinery fault diagnosis. *Neurocomputing* **2020**, *407*, 121–135. [[CrossRef](#)]
20. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A Survey on Deep Transfer Learning. In *Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2018*; Kurkova, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 270–279.
21. Lu, W.; Liang, B.; Cheng, Y.; Meng, D.; Yang, J.; Zhang, T. Deep Model Based Domain Adaptation for Fault Diagnosis. *IEEE Trans. Ind. Electron.* **2017**, *64*, 2296–2305. [[CrossRef](#)]
22. Zhang, R.; Tao, H.; Wu, L.; Guan, Y. Transfer Learning With Neural Networks for Bearing Fault Diagnosis in Changing Working Conditions. *IEEE Access* **2017**, *5*, 14347–14357. [[CrossRef](#)]
23. Bhuiyan, M.R.; Uddin, J. Deep Transfer Learning Models for Industrial Fault Diagnosis Using Vibration and Acoustic Sensors Data: A Review. *Vibration* **2023**, *6*, 218–238. [[CrossRef](#)]
24. Wen, L.; Gao, L.; Li, X. A New Deep Transfer Learning Based on Sparse Auto-Encoder for Fault Diagnosis. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 136–144. [[CrossRef](#)]
25. Guo, L.; Lei, Y.; Xing, S.; Yan, T.; Li, N. Deep Convolutional Transfer Learning Network: A New Method for Intelligent Fault Diagnosis of Machines With Unlabeled Data. *IEEE Trans. Ind. Electron.* **2019**, *66*, 7316–7325. [[CrossRef](#)]
26. Yang, B.; Lei, Y.; Jia, F.; Xing, S. An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings. *Mech. Syst. Signal Process.* **2019**, *122*, 692–706. [[CrossRef](#)]
27. Yang, X.; Song, Z.; King, I.; Xu, Z. A Survey on Deep Semi-Supervised Learning. *IEEE Trans. Knowl. Data Eng.* **2022**, *1*, 1–20. [[CrossRef](#)]
28. Shao, S.; McAleer, S.; Yan, R.; Baldi, P. Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2446–2455. [[CrossRef](#)]
29. Cao, P.; Zhang, S.; Tang, J. Preprocessing-Free Gear Fault Diagnosis Using Small Datasets With Deep Convolutional Neural Network-Based Transfer Learning. *IEEE Access* **2018**, *6*, 26241–26253. [[CrossRef](#)]
30. Antonini, M.; Barlaud, M.; Mathieu, P.; Daubechies, I. Image coding using wavelet transform. *IEEE Trans. Image Process.* **1992**, *1*, 205–220. [[CrossRef](#)]

31. Smith, W.A.; Randall, R.B. Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study. *Mech. Syst. Signal Process.* **2015**, *64–65*, 100–131. [[CrossRef](#)]
32. Yan, X.; She, D.; Xu, Y.; Jia, M. Deep regularized variational autoencoder for intelligent fault diagnosis of rotor–bearing system within entire life-cycle process. *Knowl.-Based Syst.* **2021**, *226*, 107142. [[CrossRef](#)]
33. Yan, X.; She, D.; Xu, Y. Deep order–wavelet convolutional variational autoencoder for fault identification of rolling bearing under fluctuating speed conditions. *Expert Syst. Appl.* **2023**, *216*, 119479. [[CrossRef](#)]
34. Yan, X.; Liu, Y.; Jia, M. Multiscale cascading deep belief network for fault identification of rotating machinery under various working conditions. *Knowl.-Based Syst.* **2020**, *193*, 105484. [[CrossRef](#)]
35. Yan, X.; Liu, Y.; Xu, Y.; Jia, M. Multistep forecasting for diurnal wind speed based on hybrid deep learning model with improved singular spectrum decomposition. *Energy Convers. Manag.* **2020**, *225*, 113456. [[CrossRef](#)]
36. Yan, X.; Liu, Y.; Jia, M. Health condition identification for rolling bearing using a multi-domain indicator-based optimized stacked denoising autoencoder. *Struct. Health Monit.* **2020**, *19*, 1602–1626. [[CrossRef](#)]
37. Rahimi, A.; Recht, B. Random Features for Large-Scale Kernel Machines. In *Proceedings of the Advances in Neural Information Processing Systems*; Platt, J., Koller, D., Singer, Y., Roweis, S., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2008; Volume 20, pp. 1177–1184.
38. Le, Q.; Sarlos, T.; Smola, A. Fastfood—Approximating Kernel Expansions in Loglinear Time. In *Proceedings of the 30th International Conference on Machine Learning*; Dasgupta, S., McAllester, D., Eds.; Microtome Publishing: Brookline, MA, USA, 2013; Volume 28, pp. 244–252.
39. He, Z.; Zeng, Y.; Shao, H.; Hu, H.; Xu, X. Novel motor fault detection scheme based on one-class tensor hyperdisk. *Knowl.-Based Syst.* **2023**, *262*, 110259. [[CrossRef](#)]
40. Farsoni, S.; Simani, S.; Castaldi, P. Fuzzy and Neural Network Approaches to Wind Turbine Fault Diagnosis. *Appl. Sci.* **2021**, *11*, 5035. [[CrossRef](#)]
41. Hoskins, J.C.; Kaliyur, K.M.; Himmelblau, D.M. Fault diagnosis in complex chemical plants using artificial neural networks. *AIChE J.* **1991**, *37*, 137–141. [[CrossRef](#)]
42. Watanabe, K.; Matsuura, I.; Abe, M.; Kubota, M.; Himmelblau, D.M. Incipient fault diagnosis of chemical processes via artificial neural networks. *AIChE J.* **1989**, *35*, 1803–1812. [[CrossRef](#)]
43. Himmelblau, D.M.; Barker, R.W.; Siewatanakul, W. Fault classification with the aid of artificial neural networks. In *Proceedings of the IFAC/IMACS Symposium SAFEPROCESS '91*, Baden-Baden, Germany, 10–13 September 1991; Volume 2, pp. 369–373.
44. Shaker, M.S.; Patton, R.J. Active sensor fault tolerant output feedback tracking control for wind turbine systems via T–S model. *Eng. Appl. Artif. Intell.* **2014**, *34*, 1–12. [[CrossRef](#)]
45. Xu, D.; Jiang, B.; Shi, P. Nonlinear actuator fault estimation observer: An inverse system approach via a T–S fuzzy model. *Int. J. Appl. Math. Comput. Sci.* **2012**, *22*, 183–196. [[CrossRef](#)]
46. Isermann, R. On fuzzy logic applications for automatic control, supervision and fault diagnosis. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **1998**, *28*, 221–235. [[CrossRef](#)]
47. Palade, V.; Patton, R.J.; Uppal, F.J.; Quevedo, J.; Daley, S. Fault Diagnosis of an Industrial Gas Turbine Using Neuro–Fuzzy Methods. In *Proceedings of the IFAC'02—15th IFAC World Congress on Automatic Control*, Barcelona, Spain, 21–26 July 2002; pp. 471–476.
48. Uppal, F.J.; Patton, R.J.; Palade, V. Neuro-Fuzzy Based Fault Diagnosis Applied to an Electro–Pneumatic Valve. In *Proceedings of the IFAC'02—15th IFAC World Congress on Automatic Control*, Barcelona, Spain, 21–26 July 2002; pp. 477–482.
49. Dai, W.; Yang, Q.; Xue, G.R.; Yu, Y. Boosting for Transfer Learning. In *ICML '07: Proceedings of the 24th International Conference on Machine Learning*; Association for Computing Machinery: New York, NY, USA, 2007; pp. 193–200. [[CrossRef](#)]
50. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [[CrossRef](#)]
51. Olivas, E.S.; Guerrero, J.D.M.; Sober, M.M.; Benedito, J.R.M.; Lopez, A.J.S. *Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques—2 Volumes*; Information Science Reference; IGI Publishing: Hershey, PA, USA, 2009. [[CrossRef](#)]
52. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *J. Big Data* **2016**, *3*, 1–40. [[CrossRef](#)]
53. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proc. IEEE* **2020**, *109*, 43–76. [[CrossRef](#)]
54. Chatterjee, J.; Dethlefs, N. Deep learning with knowledge transfer for explainable anomaly prediction in wind turbines. *Wind Energy* **2020**, *23*, 1693–1710. [[CrossRef](#)]
55. Yang, X.; Zhang, Y.; Lv, W.; Wang, D. Image recognition of wind turbine blade damage based on a deep learning model with transfer learning and an ensemble learning classifier. *Renew. Energy* **2021**, *163*, 386–397. [[CrossRef](#)]
56. Chen, W.; Qiu, Y.; Feng, Y.; Li, Y.; Kusiak, A. Diagnosis of wind turbine faults with transfer learning algorithms. *Renew. Energy* **2021**, *163*, 2053–2067. [[CrossRef](#)]
57. Guo, J.; Wu, J.; Zhang, S.; Long, J.; Chen, W.; Cabrera, D.; Li, C. Generative Transfer Learning for Intelligent Fault Diagnosis of the Wind Turbine Gearbox. *Sensors* **2020**, *20*, 1361. [[CrossRef](#)]
58. Li, Y.; Jiang, W.; Zhang, G.; Shu, L. Wind turbine fault diagnosis based on transfer learning and convolutional autoencoder with small-scale data. *Renew. Energy* **2021**, *171*, 103–115. [[CrossRef](#)]

59. Liu, X.; Cao, Z.; Zhang, Z. Short-term predictions of multiple wind turbine power outputs based on deep neural networks with transfer learning. *Energy* **2021**, *217*, 119356. [[CrossRef](#)]
60. Chen, J.; Patton, R.J. *Robust Model-Based Fault Diagnosis for Dynamic Systems*; Kluwer Academic Publishers: Boston, MA, USA, 1999.
61. IEC 61400-25; Communications for Monitoring and Control of Wind Power Plants. IEC: Geneva, Switzerland, 2023.
62. Marti-Puig, P.; Blanco-M, A.; Cardenas, J.J.; Cusido, J.; Sole-Casals, J. Feature Selection Algorithms for Wind Turbine Failure Prediction. *Energies* **2019**, *12*, 453. [[CrossRef](#)]
63. Blanco-M, A.; Marti-Puig, P.; Gibert, K.; Cusido, J.; Sole-Casals, J. A Text-Mining Approach to Assess the Failure Condition of Wind Turbines Using Maintenance Service History. *Energies* **2019**, *12*, 1982. [[CrossRef](#)]
64. Odgaard, P.F.; Stoustrup, J.; Kinnaert, M. Fault-Tolerant Control of Wind Turbines: A Benchmark Model. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 1168–1182. [[CrossRef](#)]
65. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. *Handb. Brain Theory Neural Netw.* **1995**, *3361*, 1995. [[CrossRef](#)]
66. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [[CrossRef](#)]

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