Measurement 116 (2018) 273–280

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

Fault detection of broken rotor bar in LS-PMSM using random forests

Juan C. Quiroz[a], Norman Mariun[b,c], Mohammad Rezazadeh Mehrjou[b,c,*], Mahdi Izadi[d,e], Norhisam Misron[c], Mohd Amran Mohd Radzi[b,c]

[a] Sunway University, Bandar Sunway, Selangor, Malaysia
[b] Centre of Advanced Power and Energy Research (CAPER), Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
[c] Department of Electrical and Electronic Engineering, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
[d] Centre for Electromagnetic and Lightning Protection Research (CELP), Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
[e] Department of Electrical Engineering, Firoozkooh Branch, Islamic Azad University, Firoozkooh, Iran

ARTICLE INFO

Keywords:
Line start-permanent magnet motor
Broken rotor bar
Fault detection
Startup current
Statistical features
Random forest

ABSTRACT

This paper proposes a new approach to diagnose broken rotor bar failure in a line start-permanent magnet synchronous motor (LS-PMSM) using random forests. The transient current signal during the motor startup was acquired from a healthy motor and a faulty motor with a broken rotor bar fault. We extracted 13 statistical time domain features from the startup transient current signal, and used these features to train and test a random forest to determine whether the motor was operating under normal or faulty conditions. For feature selection, we used the feature importances from the random forest to reduce the number of features to two features. The results showed that the random forest classifies the motor condition as healthy or faulty with an accuracy of 98.8% using all features and with an accuracy of 98.4% by using only the mean-index and impulsion features. The performance of the random forest was compared with a decision tree, Naïve Bayes classifier, logistic regression, linear ridge, and a support vector machine, with the random forest consistently having a higher accuracy than the other algorithms. The proposed approach can be used in industry for online monitoring and fault diagnostic of LS-PMSM motors and the results can be helpful for the establishment of preventive maintenance plans in factories.

1. Introduction

Electrical motors convert electricity to mechanical energy. They account for two thirds of the total electricity use in industrial sites [1]. As a consequence, electrical machine manufacturers continuously strive to reduce the amount of energy used by motors. The standard IEC/EN 60034-30:2008 proposes IE4 as the highest efficiency for motors [2]. A LS-PMSM consists of a stator and a hybrid rotor. The rotor is comprised of an electricity conducting squirrel-cage and pairs of permanent magnet poles. The efficiency of LS-PMSMs stems from the combination of elements from permanent magnet synchronous motors and induction motors. The LS-PMSM provides (1) high efficiency, similar to permanent magnet synchronous motors, and (2) high starting torque, similar to induction motors [3].

Failures in electrical motors are common and difficult to prevent because motors are generally operated in industrial sites with different types of stress causing failures in various motor parts [4]. This has led to research on methods for early detection of failure in motors, to prevent motor inefficiencies and motor shutdown. In particular, rotor faults are significant because they exacerbate failures in other parts of the motor [5]. Various sensing techniques have been developed for broken rotor bar detection in electrical motors [5]. For instance, motor current signature analysis (MCSA) is a widely used technique due to its low cost and non-invasive nature [6]. In MCSA, the steady state current of a running motor is collected and recorded. From the recorded signal, features are extracted from the time domain, frequency domain, or time-frequency domain. These features are then used to make a diagnosis of the motor.

Fault analysis in induction motors has been widely applied. MCSA has been used to analyze faults in induction motors, such as rotor faults, bearing faults, eccentricity, misalignment, and stator faults [7–11]. Similar techniques have also been used to analyze vibration [12–16] and acoustic [17] signals of induction motors. The limitation of prior work is that most fault analysis has been applied to induction motors, electrical motors, fans, and gear boxes [7–17]. Yet, fault analysis in LS-PMSMs has been limited to a smaller set of faults, such as rotor faults, static eccentricity faults, and demagnetization [18–21]. Fault analysis in LS-PMSMs also suffers from a number of shortcomings: (1) the use of

* Corresponding author at: Department of Electrical and Electronic Engineering, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia.
E-mail address: mehrjou.mo@gmail.com (M.R. Mehrjou).

https://doi.org/10.1016/j.measurement.2017.11.004
Received 29 August 2016; Received in revised form 19 October 2017; Accepted 2 November 2017

mathematical and simulated models to analyze faults, as opposed to using an LS-PMSM machine to collect data for fault detection; (2) the use of steady-state current for fault analysis; and (3) lack of machine learning algorithms for fault detection.

This paper makes three contributions. First, we used an LS-PMSM machine to collect current data while subjecting the motor to different loads. The rotor faults in our LS-PMSM machine were created by physically damaging the rotors of the LS-PMSM. We also analyzed the LS-PMSM starting with an initial load, as opposed to introducing a load after the motor had reached steady state, which is the common practice in prior work [7–11].

Second, we analyzed the transient current from when the motor is started. That is, we started the motor from standstill and waited for the motor to reach steady state, with the current from this transient period used for our analysis. Prior research uses the current from the steady state for fault analysis [7–11]. Finally, our third contribution is that our work is the first to apply machine learning for rotor fault detection in LS-PMSMs. We used random forests for the detection of rotor faults, and assessed the effectiveness of random forests by comparing with a decision tree, a Naïve Bayes classifier, logistic regression, linear ridge, and a support vector machine. To train these machine learning algorithms, we extracted thirteen time domain features from the transient current signal of the LS-PMSM, with the selection of the features based on prior work [22,23]. While machine learning methods have been used for fault detection in induction motors [7–11], to the best of our knowledge this is the first work to present fault analysis in LS-PMSMs by comparing various machine learning algorithms and using features extracted from the transient current signal to train and test these algorithms.

2. Fault detection with machine learning

A random forest is a machine learning algorithm consisting of a number of independent decision trees [24]. A decision tree classifies an instance by testing attributes of the instance at each node of the tree [25]. Each node tests a particular attribute, with the leaves of the tree representing the output labels. Moving down a particular branch of a tree tests particular attributes at each node in order to arrive to an output label. A decision tree is typically built following a greedy approach, with the attribute/feature that results in the best split of the training data being used for the root node, and subsequently the attributes/features that result in the next best splits being used in the children nodes.

In contrast to a decision tree, a random forest uses bagging to build the decision trees in the forest [24]. In bagging, T bootstrap sets are made by sampling with replacement N training examples from the training set, with T indicating the number of trees in the forest. Only 2/3 of each bootstrap set are used to build each tree, with the remaining 1/3, referred to as the out-of-bag data, used to get an estimate of the classification error of each tree. Fig. 1 illustrates the process of building a random forest with T trees.

In a typical decision tree, the greedy approach to building the tree can result in cases where the weaker features are not used at all. A random forest addresses this by choosing the best split in each node from a random subset of all the available features [24]. The random feature subset used for determining best node splits allows the weaker features to be represented in the random forest. Trees are grown to maximum length and without pruning to get low bias. Low correlation between the trees in the random forest is achieved by randomization as a result of the bootstrap samples and the random selection of features at each split [26]. Random forests have performed well in applications of fault diagnosis in rotating machinery [10,24].

Once the random forest has been built, an instance (x) is classified by passing the instance to each decision tree in the random forest (Fig. 2). Each decision tree classifies the instance by following a particular branch of the tree depending on the outcome from each node. The output of the random forest is then decided by taking the majority of the outputs from each tree. That is, the output of each tree is considered a vote, with the majority vote determining the output of the random forest.

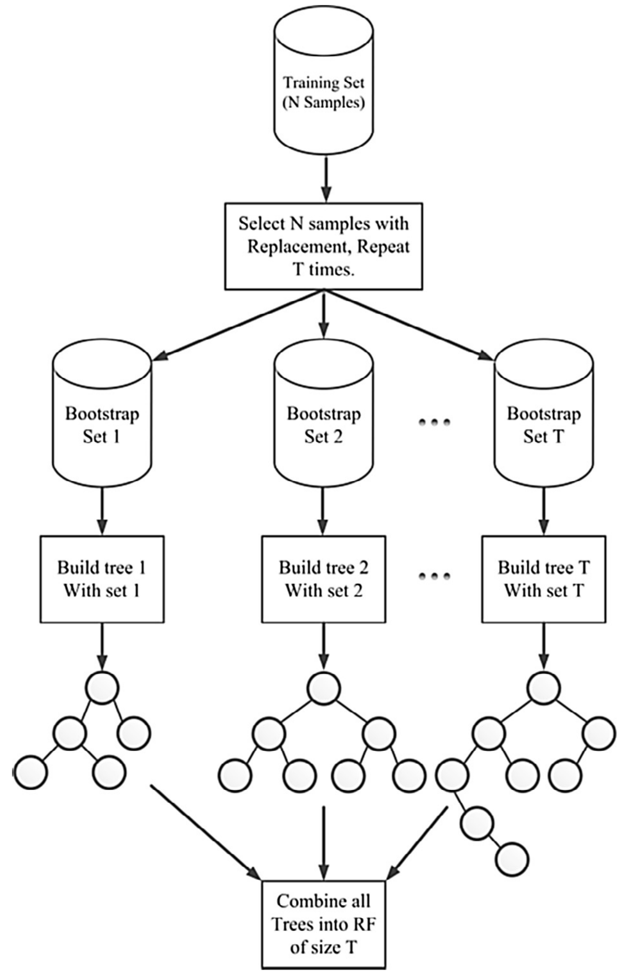

Fig. 1. Building a random forest with bagging.

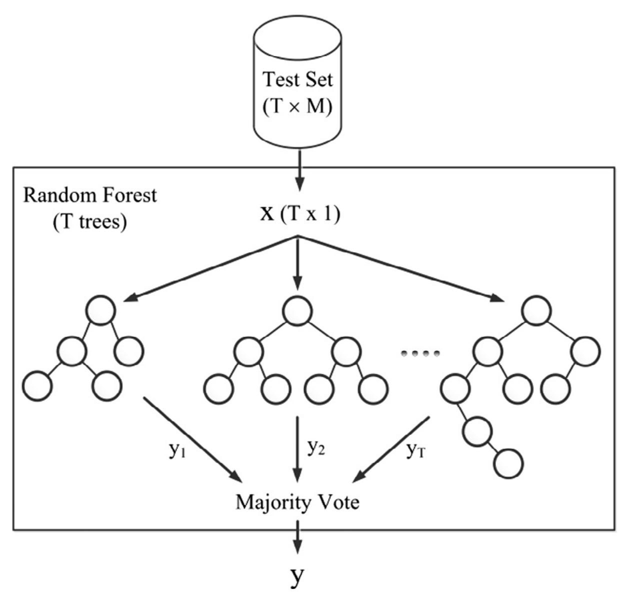

Fig. 2. Classifying an instance from the test set by passing the instance to each tree in the forest, and combining the outputs from all the trees using majority vote.

Random forests provide a way to perform feature selection by using