Elevator Fault Detection Using Profile Extraction and Deep Autoencoder Feature Extraction for Acceleration and Magnetic Signals

Krishna Mohan Mishra * and Kalevi Huhtala

Unit of Automation Technology and Mechanical Engineering, Tampere University, 33720 Tampere, Finland

* Correspondence: krishna.mishra@tuni.fi

Received: 26 June 2019; Accepted: 22 July 2019; Published: 25 July 2019

Abstract: In this paper, we propose a new algorithm for data extraction from time-series data, and furthermore automatic calculation of highly informative deep features to be used in fault detection. In data extraction, elevator start and stop events are extracted from sensor data including both acceleration and magnetic signals. In addition, a generic deep autoencoder model is also developed for automated feature extraction from the extracted profiles. After this, extracted deep features are classified with random forest algorithm for fault detection. Sensor data are labelled as healthy and faulty based on the maintenance actions recorded. The remaining healthy data are used for validation of the model to prove its efficacy in terms of avoiding false positives. We have achieved above 90% accuracy in fault detection along with avoiding false positives based on new extracted deep features, which outperforms results using existing features. Existing features are also classified with random forest to compare results. Our developed algorithm provides better results due to the new deep features extracted from the dataset when compared to existing features. This research will help various predictive maintenance systems to detect false alarms, which will in turn reduce unnecessary visits of service technicians to installation sites.

Keywords: elevator system; deep autoencoder; fault detection; feature extraction; random forest; profile extraction

1. Introduction

In recent years, elevator systems have been used increasingly extensively in apartments, commercial facilities, and office buildings. Presently 54% of the world’s population lives in urban areas [1]. Therefore, elevator systems need proper maintenance and safety. The next step for improving the safety of elevator systems is the development of predictive and pre-emptive maintenance strategies, which will also reduce repair costs and increase the lifetime while maximizing the uptime of the system [2,3]. Elevator production and service companies are now opting for a predictive maintenance policy to provide better service to customers. They are remotely monitoring faults in elevators and estimating the remaining lifetime of the components responsible for faults. Elevator systems require fault detection and diagnosis for healthy operation [4].

Fault diagnosis methods based on deep neural networks [5–7] and convolutional neural networks [8,9] feature extraction methodology are presented as state of the art for rotatory machines similar to elevator systems. Support vector machines [10] and extreme learning machines [11] are also used as fault detection methods for rotatory machines. However, we have developed an intelligent deep autoencoder random forest-based feature extraction methodology for fault detection in elevator systems to improve the performance of traditional fault diagnosis methods.
Profile extraction for health monitoring is a major issue in automated industrial applications such as elevator systems, computer numerical control, machinery, and robotics [12]. Although rotating machine have been running for decades, but profile extraction and processing methods are not widely available [13]. Profile extraction methods have applied in electric vehicles [14], computer numerical control systems [15] and horizontal planes [16]. Kalman filter [17] is one of the methods being used for profile extraction. However, we have developed an off-line profile extraction algorithm based on low-pass filtering and peak detection to extract elevator start and stop events from sensor data including both acceleration and magnetic signals.

In the last decade, neural networks [18] have extracted highly meaningful statistical patterns from large-scale and high-dimensional datasets. Neural networks [19] has also been used to improve elevator ride comfort via speed profile design. Neural networks [20] has been applied successfully to nonlinear time-series modeling. A deep learning network can self-learn the relevant features from multiple signals [21]. Deep learning algorithms are frequently used in areas such as bearing fault diagnosis [22], machine defect detection [23], vibration signal analysis [24], computer vision [25] and image classification [26]. Autoencoding is a process for nonlinear dimension reduction with natural transformation architecture using feedforward neural network [27]. Autoencoders have proven powerful as nonlinear feature extractors [28]. Autoencoders can increase the generalization ability of machine learning models by extracting features of high interest as well as making possible its application to sensor data [29]. Autoencoders were first introduced by LeCun [30], and have been studied for decades. Traditionally, feature learning and dimensionality reduction are the two main features of autoencoders. Recently, autoencoders have been considered one of the most compelling subspace analysis techniques because of the existing theoretical relations between autoencoders and latent variable models [31]. Autoencoders have been used for feature extraction from the data in systems such as induction motors [32] and wind turbines [33] for fault detection, different from elevator systems as in our research.

In our previous research, raw sensor data, mainly acceleration signals, were used to calculate elevator key performance and ride quality features, which we call here existing features. Random forest was used for fault detection based on these existing features. Existing domain specific features are calculated from raw sensor data, but that requires expert knowledge of the domain and results in a loss of information to some extent. To avoid these implications, we have developed an algorithm for profile extraction from the raw sensor data rides including both acceleration and magnetic signals. In addition, a generic algorithm with deep autoencoder random forest approach for automated feature extraction from raw sensor data profiles for fault detection in elevator systems.

Our off-line profile extraction algorithm is signal-based and deep autoencoder random forest method is model-based. First, it extracts profiles from time-series signal and then, calculates highly informative deep features from extracted profiles. It is better than other algorithms because it provides better results, dimensionality reduction and is robust against overfitting characteristics.

We have proposed a reliable fault detection model with above 90% accuracy in fault detection, which will increase the safety of passengers. In addition, we have validated the efficacy of the pre-trained model in terms of false positives with the remaining healthy rides, which is helpful in detecting false alarms for elevator predictive maintenance strategies. It is extremely helpful in reducing unnecessary visits by maintenance personnel to installation sites.

Figure 1 shows the fault detection approach used in this paper, which includes raw sensor data rides extracted based on time periods provided by the maintenance data from all floor patterns. Acceleration and magnetic signal rides collected from an elevator system are fed to the algorithm for profile extraction separately. These extracted profiles from all five traction elevators including both acceleration and magnetic signals are then fed to the deep autoencoder model for feature extraction, and then random forest performs the fault detection task based on extracted deep features. We only extract start and stop profiles from the both acceleration and magnetic signal rides because of the...
different lengths of rides for each floor combination due to the constant speed phase, which is longer when there is longer travel.

This paper provides the following novelties. (1) We propose a new off-line profile extraction algorithm for extracting elevator start and stop events from time-series data. (2) In addition, we propose a new deep autoencoder model to automatically generate highly informative deep features from sensor data for fault detection. The rest of this paper is organized as follows. Section 2 presents the methodology of the paper including profile extraction, deep autoencoder, and random forest algorithms. Then, Section 3 includes the details of experiments performed, results, and discussion. Finally, Section 4 concludes the paper and presents the future work.

Figure 1. Fault detection approach (Acc and Mag represent acceleration and magnetic signals respectively).

2. Methodology

In this study, we have used 12 different existing features derived from raw sensor data describing the motion and vibration of an elevator for fault detection and diagnostics of multiple faults. We have developed an automated feature extraction technique for raw sensor data in this research as an extension to the work of our previous research to compare the results using new extracted deep features. In our previous research [34], we have used only acceleration signal, which represents vibration related features. In this research, we have extended our approach to include magnetic signals, which represents position related features. This will validate our goal of this research to develop generic models for profile extraction and automated feature extraction for fault detection in the health state monitoring of elevator systems. In addition, we have analyzed almost two months of the data from five traction elevators in this research as an extension to one elevator in our previous research. Each elevator usually produces around 200 rides per day. Each ride used in analysis contains around 5000 rows of the data, which proves robustness of the algorithms over large dataset. We have excluded around 20 rides before and after the time period of faulty rides in selecting healthy rides, which will help us to remove suspicious data from the analysis with our algorithm. We have used 70% of the data for training and rest 30% for testing.

2.1. Profile Extraction Algorithm

Raw sensor data collected from elevator systems typically encompass a large collection of data points sampled at high frequency. In order to feed large sensor data to cloud-based applications, it is often desirable to pre-process the data and perform compression before transmission, for example in the form of edge computing performed in the device end. Here we assume that raw data is in the form of a one-dimensional time-series vector with equidistant sampling times. The goal of the proposed
method is to compress the raw time series obtained from machinery while maintaining the information about key events, and secondly, to make the data more applicable for machine learning.

The algorithm works in two stages. In the first stage, the signal is pre-processed and normalized, followed by low-pass filtering. The low-pass filtered signal is used for peak detection, which for each elevator travel detects a local minimum and maximum corresponding to acceleration and deceleration (start and stop) events. The algorithm uses crude low-pass filtering with a low cut-off frequency for peak detection, which ensures that a sustained period of acceleration is required for a peak to be registered. This prevents short bursts of noise from being detected as a movement window. Low-pass filtering is applied to ensure that only a sustained acceleration or deceleration event is registered as a peak as opposed to noise.

Low-pass filter is used to avoid noise spikes being detected as peaks. From low-pass filter signal, the algorithm cannot detect the precise magnitude or timing of peaks, but it will detect the approximate region in which to align the event profiles. Unfiltered data is then used for profile alignment.

In the second stage, alignment and collection of equal length profiles is performed based on windowing of the acceleration signal near the peak events. In this stage, the raw acceleration signal is used instead of the filtered signal. A number of time domain alignment methods have been proposed in the literature. Dynamic time warping (DTW) has been commonly applied, e.g., in speech recognition [35], whereas various alignment techniques for sensor data have been presented in [36]. Here alignment is performed against a reference profile. The reference profile is aligned against the raw data in the window of the detected peaks. The length of the initial profile window \( m \) is selected empirically based on the sample frequency and the maximum estimated length of the elevator acceleration events. The criterion for optimal alignment was defined as the alignment that minimizes the sum of the Euclidean or \( L_2 \) norm. The output from this operation is an \( n \times m \) matrix of aligned profiles describing \( n \) acceleration and deceleration events of length \( m \).

To improve the alignment accuracy, the reference profile is updated iteratively following each batch run. Each sequence in the profile matrix is closely synchronized in time and can hence be considered a repetition of the same signal. Using signal averaging, the new reference profile is calculated as the mean of the \( n \) extracted profiles. This both maintains the main characteristics of the signal and reduces the noise. Assuming white noise and perfect synchronization, signal averaging improves the signal-to-noise ratio (SNR) by a factor of \( \sqrt{n} \).

Information in the obtained reference profile can be used to update the window size \( m \). Assuming an overestimated size of the event window, the averaged reference profile will contain superfluous close to zero values corresponding to no acceleration. The number of elements \( s \) below this threshold in the reference profile can be used to estimate the optimal window length by reducing the window length \( m \) by \( s \) for the following iteration.

The off-line profile extraction algorithm is described as following.

**Off-line profile extraction algorithm**

**Pre-processing**
1. Read a vector of raw acceleration data containing \( k \) elevator travels. Define the zero mean transformed dataset as \( X \).
2. Perform low-pass filtering on \( X \) and obtain denoised dataset \( Y \).

**Initialization**
3. Define parameters for reference profile. Set the approximated maximum window length to \( m \) samples and height \( h \) to the 99th percentile of the low-pass filtered dataset.
4. Define alignment window size \( a \) and set \( k = 1 \).

**Iteration**
5. From \( Y(k) \), detect peak acceleration points \( y_{\text{min}} \) and \( y_{\text{max}} \).
6. Align reference profile \( P \) against raw dataset \( X \) in the vicinity of detected peaks by minimizing the \( L_2 \) norm according to
\[
\begin{align*}
\min & \sum_{i=-a/2}^{a/2} \sum_{j=1}^{m} (-p_{j} - x_{\min+i+j})^2 \\
\min & \sum_{i=-a/2}^{a/2} \sum_{j=1}^{m} (p_{j} - x_{\max+i+j})^2
\end{align*}
\] (1) (2)

7. Add aligned data points from \( X(k) \) as rows into an \( n \times m \) profile matrix, alternatively separate matrices according to direction of travel (min/max).

8. Set travel window \( k = k + 1 \) and repeat steps 5–7 until end of dataset.

9. Update reference profile \( P \) with the signal-averaged profile obtained from the column-wise mean of the new profile matrix.

10. Reduce window length \( m \) by \( s \) samples, where \( s \) is the number of elements in \( P \) that satisfy

\[ p \leq \epsilon, p \in P \] (3)

where \( \epsilon \) is a close to zero number indicating no acceleration.

11. Set \( k = 1 \) and continue with new batch iterations by repeating steps 5–8.

2.2. Deep Autoencoder

The deep autoencoder model is based on deep learning autoencoder feature extraction methodology. A basic autoencoder is a fully connected three-layer feedforward neural network with one hidden layer. Typically, the autoencoder has the same number of neurons in the input and output layer and reproduces its inputs as its output. We are using a five-layer deep autoencoder (see Figure 2) including input, output, encoder, decoder, and representation layers, which is a different approach than in [33,37]. In our approach, we first analyze the data to find all floor patterns and then feed the segmented raw sensor data windows in up and down directions separately to the algorithm for profile extraction. Extracted profiles from both acceleration and magnetic signals are fed to the deep autoencoder model for extracting new deep features. Lastly, we apply random forest as a classifier for fault detection based on new deep features extracted from the profiles. We have combined healthy and faulty profiles as a vector from all five traction elevators including both acceleration and magnetic signals before feature extraction.

![Figure 2. Off-line profile extraction and deep autoencoder feature extraction approach.](image)

The encoder transforms the input \( x \) into corrupted input data \( x' \) using hidden representation \( H \) through nonlinear mapping

\[ H = f(W_1x' + b) \] (4)
where \( f(.) \) is a nonlinear activation function as the sigmoid function, \( W_1 \in \mathbb{R}^{k \times m} \) is the weight matrix and \( b \in \mathbb{R}^k \) the bias vector to be optimized in encoding with \( k \) nodes in the hidden layer [37]. Then, with parameters \( W_2 \in \mathbb{R}^{n \times k} \) and \( c \in \mathbb{R}^m \), the decoder uses nonlinear transformation to map hidden representation \( H \) to a reconstructed vector \( x'' \) at the output layer.

\[
x'' = g(W_2H + c)
\]

where \( g(.) \) is again nonlinear function (sigmoid function). In this study, the weight matrix is \( W_2 = W_1^T \), which is tied weights for better learning performance [38]. Among multiple input variables the use of nonlinear activation functions provides us better opportunity to capture nonlinear relationships. Effective fault detection is a challenge due to nonlinearity of elevator systems and as a result, time-series data will have temporal dependencies. Our proposed approach can capture nonlinear relationships among multiple sensor variables, which has improved the performance in terms of fault detection.

2.3. Random Forest

Random forest includes an additional layer of randomness to bagging. It uses different bootstrap samples of the data for constructing each tree [39]. The best subset of predictors is used to split each node in random forest. This counterintuitive strategy is the best feature of random forest, which makes it different from other classifiers as well as robust against overfitting. It is one of the most user-friendly classifiers because it consists of only two parameters: the number of variables and number of trees. However, it is not usually very sensitive to their values [40]. The final classification accuracy of random forest is calculated by averaging, i.e., arithmetic mean of the probabilities of assigning classes related to all the produced trees \( e \). Testing data \( d \) that is unknown to all the decision trees is used for evaluation by the voting method (see Figure 3).

Specifically, let sensor data value \( v^e_l \) have training sample \( l \)th in the arrived leaf node of the decision tree \( e \in E \), where \( l \in [1, ..., L_e] \) and the number of training samples is \( L_e \) in the current arrived leaf node of decision tree \( e \). The final prediction result is given by [41]:

\[
\mu = \frac{\sum_{e \in E} \sum_{l \in [1, ..., L_e]} I(h_{l}^e = y_j)}{\sum_{e \in E} L_e}
\]

All classification trees providing a final decision by voting method are given by [42]:

\[
H(a) = \arg \max_{y_j} \sum_{i \in \{1,2,\ldots,Z\}} I(h_i(a) = y_j)
\]
where \( j = 1, 2, \ldots, C \) and the combination model is \( H(a) \), the number of training subsets are \( Z \) depending on which decision tree model is \( h_i(a) \), \( i \in [1, 2, \ldots, Z] \) while output or labels of the \( P \) classes are \( y_j \), \( j = 1, 2, \ldots, P \) and combined strategy is \( I(.) \) defined as:

\[
I(x) = \begin{cases} 
1, & h_i(a) = y_j \\
0, & \text{otherwise}
\end{cases} 
\]  

where output of the decision tree is \( h_i(a) \) and \( i \)th class label of the \( P \) classes is \( y_j \), \( j = 1, 2, \ldots, P \).

2.4. Evaluation Parameters

Evaluation parameters used in this research are defined with the confusion matrix in Table 1.

| Actual (P) | Predicted (P) | Predicted (N) |
|------------|---------------|---------------|
| True positive (TP) | False negative (FN) | False positive (FP) | True negative (TN) |

The rate of positive test result is sensitivity,

\[
Sensitivity = \frac{TP}{TP + FN} \times 100\% 
\]  

The ratio of a negative test result is specificity,

\[
Specificity = \frac{TN}{TN + FP} \times 100\% 
\]  

The overall measure is accuracy,

\[
Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% 
\]  

3. Results and Discussion

In this research, we first selected all floor patterns like floor 2–5, 3–8, and so on from the data, some of which are shown in Table 2.

| Start Floor | Stop Floor |
|-------------|------------|
| 0           | 1          |
| 2           | 5          |
| 3           | 8          |
| 4           | 6          |

The next step includes the selection of faulty rides from all floor patterns based on time periods provided by the maintenance data. An equal number of healthy rides are also selected. Only the vertical component of both acceleration and magnetic signal data is selected in this research because it is the most informative aspect, consisting of significant changes in vibration levels as compared to other components. Healthy and faulty rides are fed to the algorithm for profile extraction separately. Start and stop profiles are of equal length, irrespective of floor combination.
First, we have selected all floor patterns from the data and then divided the data into up and down directions. Next, we selected the rides from healthy and faulty parts of the data and extracted profiles from them. These profiles are fed to deep autoencoder model for feature extraction and based on these feature faults are detected.

3.1. Up Movement

We have analyzed up and down movements separately because the traction-based elevator usually produces slightly different levels of vibration in each direction. First, we have selected faulty rides based on time periods provided by the maintenance data, including all floor patterns, which is fed to the algorithm for profile extraction, as shown in Figure 4. Then, we have selected an equal number of rides for healthy data, and the extracted profiles are shown in Figure 5. Visualization of the profiles proved that our proposed algorithm extracted elevator start and stop events have equal length irrespective of the floor combination as shown in Figures 4 and 5.

![Figure 4](image)

**Figure 4.** Profiles from faulty rides (The figure shows first 10 rides from the combined version of elevator start and stop events from magnetic signals in z-direction i.e., Mag vertical. Profiles are extracted from all floor combinations during the time period of fault occurred and fixed by maintenance personal in upward movement of elevator system. Color labelling is based on maximum number of rides available in the data during this time period i.e., variable.).

The next step is to label both the healthy and faulty profiles with class labels 0 and 1, respectively. Healthy and faulty profiles with class labels are fed to the deep autoencoder model and the generated deep features are shown in Figure 6. These are called deep features or latent features in deep autoencoder terminology, which shows hidden representations of the data. In Figure 6, we can see from visualization that both features with class labels are perfectly separated, which results in better fault detection.

Extracted deep features are fed to the random forest algorithm for classification, and the results provide 90% accuracy in fault detection as shown in Table 3. We have compared accuracy in terms
of avoiding false positives from both features and found that new deep features generated in this research outperform the existing features. We have used the remaining healthy rides for extracting profiles to analyze the number of false positives. These healthy profiles are labelled as class 0 and fed to the deep autoencoder to extract new deep features from the profiles, as shown in Figure 7.

Figure 5. Profiles from healthy rides (The figure shows first 10 rides from the combined version of elevator start and stop events from magnetic signals in z-direction i.e., Mag vertical. Equal number of profiles as in Figure 4 are extracted from all floor combinations before the time period of fault occurred in upward movement of elevator system. Color labelling is based on number of rides available in the data during this time period i.e., variable.).

These new deep features are then classified with the pre-trained deep autoencoder random forest model to test the efficacy of the model in terms of false positives. Table 3 presents the results for upward movement of the elevator in terms of accuracy, sensitivity, and specificity. We have also included the accuracy of avoiding false positives as an evaluation parameter for this research. The results show that the new deep features provide better accuracy in terms of fault detection and avoiding false positives from the data, which is helpful in detecting false alarms for elevator predictive maintenance strategies. It is extremely helpful in reducing unnecessary visits by maintenance personnel to installation sites.

Table 3. Fault detection analysis (False positives field related to analyzing remaining healthy profiles after the training and testing phase).

|                             | Deep Features | Existing Features |
|-----------------------------|---------------|-------------------|
| Accuracy                    | 0.90          | 0.54              |
| Sensitivity                 | 0.92          | 0.50              |
| Specificity                 | 0.88          | 0.58              |
| False positives             | 0.86          | 0.31              |
Figure 6. Extracted deep autoencoder features (visualization of the features w.r.t class variable).

Figure 7. Extracted deep features (only healthy profiles).
3.2. Down Movement

For downward motion, we have repeated the same analysis procedure as in the case of upward motion. First we have selected the faulty rides and an equal amount of healthy data rides for profile extraction, as shown in Figures 8 and 9.

![Figure 8. Profiles from faulty rides (similar to Figure 4 but in downward movement of elevator system).](image)

Again, we fed both healthy and faulty profiles with class labels to the deep autoencoder for the extraction of new deep features, as shown in Figure 10.

Finally, the new extracted deep features are classified with random forest model and the results are shown in Table 4. After this, the remaining healthy rides are used to analyze the number of false positives. The extracted deep features are shown in Figure 11.

Table 4 presents the results for fault detection with deep autoencoder random forest model in the downward direction. The results are similar to the upward direction, but we can see significant change in terms of accuracy of fault detection and when analyzing the number of false positives with new deep features.

|                      | Deep Features | Existing Features |
|----------------------|---------------|------------------|
| Accuracy             | 0.95          | 0.59             |
| Sensitivity          | 0.92          | 0.67             |
| Specificity          | 0.97          | 0.50             |
| False positives      | 1             | 0.61             |
Figure 9. Profiles from healthy rides (similar to Figure 5 but in downward movement of elevator system).

Figure 10. Extracted deep features.
4. Conclusions and Future Work

This research focuses on the health monitoring of elevator systems using a novel fault detection technique. The goal of this research was to develop generic models for profile extraction and automated feature extraction for fault detection in the health state monitoring of elevator systems. Our approach in this research provided above 90% accuracy in fault detection and in the case of analyzing false positives for all floor combinations with new extracted deep features from sensor data including both acceleration and magnetic signals. The results support the goal of this research of developing generic models which can be used in other machine systems for fault detection. The results are useful in terms of detecting false alarms in elevator predictive maintenance. The approach will also reduce unnecessary visits of maintenance personnel to installation sites if the analysis results are used to allocate maintenance resources. Our developed models can also be used for different predictive maintenance solutions to automatically generate highly informative deep features for solving diagnostics problems. Our models outperform others because of new deep features extracted from the dataset as compared to existing features calculated from the same raw sensor dataset. The automated feature extraction approach does not require any prior domain knowledge. It also provides dimensionality reduction and is robust against overfitting characteristics. The experimental results show the feasibility of our generic models, which will increase the safety of passengers as well as serve the public interest. Visualization of the extracted profiles and features support our goal of developing generic models for profile and feature extraction for fault detection.

In future work, we will extend our approach on other real-world big data cases to validate its potential for other applications and improve its efficacy.

Author Contributions: Conceptualization, K.M.M. and K.H.; Data curation, K.M.M.; Formal analysis, K.M.M.; Investigation, K.M.M.; Methodology, K.M.M.; Resources, K.M.M.; Software, K.M.M.; Supervision, K.H.; Validation, K.M.M. and K.H.; Visualization, K.M.M.; Writing—original draft, K.M.M.; Writing—review and editing, K.M.M. and K.H.
Funding: This research received no external funding.

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

References

1. Desa, U. World Urbanization Prospects; the 2011 Revision; Population Division, Department of Economic and Social Affairs, United Nations Secretariat: New York, NY, USA, 2014.
2. Ebeling, T. Condition monitoring for elevators—An overview. Lift Rep. 2011, 6, 25–26.
3. Ebeling, T.; Haul, M. Results of a field trial aiming at demonstrating the permanent detection of elevator wear using intelligent sensors. Proc. ELEVCON 2016, 20, 101–109.
4. Wang, P.; He, W.; Yan, W. Fault diagnosis of elevator braking system based on wavelet packet algorithm and fuzzy neural network. In Proceedings of the 2009 9th International Conference on Electronic Measurement & Instruments, Beijing, China, 16–19 August 2009; pp. 4–1028.
5. Zhang, R.; Peng, Z.; Wu, L.; Yao, B.; Guan, Y. Fault diagnosis from raw sensor data using deep neural networks considering temporal coherence. Sensors 2017, 17, 549. [CrossRef] [PubMed]
6. Jia, F.; Lei, Y.; Lin, J.; Zhou, X.; Lu, N. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. Mech. Syst. Signal Process. 2016, 72, 303–315. [CrossRef]
7. Bulla, J.; Orjuela-Cañón, A.D.; Flórez, O.D. Feature extraction analysis using filter banks for faults classification in induction motors. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–6.
8. Xia, M.; Li, T.; Xu, L.; Liu, L.; de Silva, C.W. Fault Diagnosis for Rotating Machinery Using Multiple Sensors and Convolutional Neural Networks. In Proceedings of the 2018 IEEE International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 101–110.
9. Jia, L.; Wang, T.; Zhao, M.; Wang, P. An adaptive multi-sensor data fusion method based on deep convolutional neural networks for fault diagnosis of planetary gearbox. Sensors 2017, 17, 414. [CrossRef] [PubMed]
10. Martinez-Rego, D.; Fontenla-Romero, O.; Alonso-Betanzos, A. Power wind mill fault detection via one-class v-SVM vibration signal analysis. In Proceedings of the 2011 IEEE International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; pp. 511–518.
11. Yang, Z.; Zhang, P. ELM Meets RAE-ELM: A hybrid intelligent model for multiple fault diagnosis and remaining useful life predication of rotating machinery. In Proceedings of the 2016 IEEE International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 2321–2328.
12. Osornio-Rios, R.A.; de Jesús Romero-Troncoso, R.; Herrera-Ruiz, G.; Castañeda-Miranda, R. FPGA implementation of higher degree polynomial acceleration profiles for peak jerk reduction in servomotors. Robot. Comput. Integr. Manuf. 2009, 25, 379–392. [CrossRef]
13. Carr, C.E.; Bryan, N.C.; Saboda, K.N.; Bhattaru, S.A.; Ruvkun, G.; Zuber, M.T. Acceleration profiles and processing methods for parabolic flight. NPJ Microgravity 2018, 4, 14. [CrossRef] [PubMed]
14. Bingham, C.; Walsh, C.; Carroll, S. Impact of driving characteristics on electric vehicle energy consumption and range. IET Intell. Transp. Syst. 2012, 6, 29–35. [CrossRef]
15. Nam, S.; Yang, M. A study on a generalized parametric interpolator with real-time jerk-limited acceleration. Comput. -Aided Des. 2004, 36, 27–36. [CrossRef]
16. Soyka, F.; Giordano, PR.; Beykirch, K.; Bülthoff, H.H. Predicting direction detection thresholds for arbitrary translational acceleration profiles in the horizontal plane. Exp. Brain Res. 2011, 209, 95–107. [CrossRef] [PubMed]
17. Wang, X.; Wu, S.; Wang, K. A run-to-run profile control algorithm for improving the flatness of nano-scale products. IEEE Trans. Autom. Sci. Eng. 2013, 12, 192–203. [CrossRef]
18. Calimeri, F.; Marzullo, A.; Stamilio, C.; Terracina, G. Graph Based Neural Networks for Automatic Classification of Multiple Sclerosis Clinical Courses. In Proceedings of the ESANN, Bruges, Belgium, 25–27 April 2018.
19. Seppala, J.; Koivistö, H.; Koivisto, H. Modeling elevator dynamics using neural networks. In Proceedings of the 1998 IEEE International Joint Conference on Neural Networks Proceedings, IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227), Anchorage, AK, USA, 4–9 May 1998; pp. 2419–2424.
20. Lee, C. Gender classification using M-estimator based radial basis function neural network. In Proceedings of the 2014 International Conference on Signal Processing and Multimedia Applications (SIGMAP), Vienna, Austria, 28–30 August 2014; pp. 302–306.
21. Fernández-Varela, I.; Athanasakis, D.; Parsons, S.; Hernández-Pereira, E.; Moret-Bonillo, V. Sleep Staging with Deep Learning: A Convolutional Model. In Proceedings of the ESANN, Bruges, Belgium, 25–27 April 2018.

22. Zhuang, Z.; Lv, H.; Xu, J.; Huang, Z.; Qin, W. A Deep Learning Method for Bearing Fault Diagnosis through Stacked Residual Dilated Convolutions. Appl. Sci. 2019, 9, 1823. [CrossRef]

23. Lin, W.; Lo, S.; Young, H.; Hung, C. Evaluation of Deep Learning Neural Networks for Surface Roughness Prediction Using Vibration Signal Analysis. Appl. Sci. 2019, 9, 1462. [CrossRef]

24. Wang, J.; Sanchez, J.; Iturrioz, J.; Ayesta, I. Geometrical Defect Detection in the Wire Electrical Discharge Machining of Fir-Tree Slots Using Deep Learning Techniques. Appl. Sci. 2019, 9, 90. [CrossRef]

25. Donati, L.; Iotti, E.; Mordonini, G.; Prati, A. Fashion Product Classification through Deep Learning and Computer Vision. Appl. Sci. 2019, 9, 1385. [CrossRef]

26. Gu, Y.; Wang, Y.; Li, Y. A Survey on Deep Learning-Driven Remote Sensing Image Scene Understanding: Scene Classification, Scene Retrieval and Scene-Guided Object Detection. Appl. Sci. 2019, 9, 2110. [CrossRef]

27. Hänninen, J.; Kärkkäinen, T. Comparison of Four-and Six-Layered Configurations for Deep Network Pretraining. In Proceedings of the ESANN, Bruges, Belgium, 27–29 April 2016.

28. Albuquerque, A.; Amador, T.; Ferreira, R.; Veloso, A.; Ziviani, N. Learning to rank with deep autoencoder features. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.

29. Huet, R.; Courty, N.; Lefèvre, S. A New Penalisation Term for Image Retrieval in Clique Neural Networks. In Proceedings of the ESANN, Bruges, Belgium, 27–29 April 2016.

30. Fogelman-Soulie, F.; Robert, Y.; Tchuente, M. Automata Networks in Computer Science: Theory and Applications; Manchester University Press: Manchester, UK; Princeton University Press: Princeton, NJ, USA, 1987.

31. Madani, P.; Viajic, N. Robustness of deep autoencoder in intrusion detection under adversarial contamination. In Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security, Raleigh, NC, USA, 10–11 April 2018; p. 1.

32. Sun, W.; Shao, S.; Zhao, R.; Yan, R.; Zhang, X.; Chen, X. A sparse auto-encoder-based deep neural network approach for induction motor faults classification. Measurement 2016, 89, 171–178. [CrossRef]

33. Jiang, G.; Xie, P.; He, H.; Yan, J. Wind turbine fault detection using a denoising autoencoder with temporal information. IEEE/ASME Trans. Mech. 2017, 23, 89–100. [CrossRef]

34. Mishra, K.M.; Krogerus, T.; Huhtala, K. Profile extraction and deep autoencoder feature extraction for elevator fault detection. In Proceedings of the 16th International Conference on Signal Processing and Multimedia Applications (SIGMAP), Prague, Czech Republic, 26–28 July 2019; pp. 319–326.

35. Di Martino, J. Dynamic time warping algorithms for isolated and connected word recognition. In New Systems and Architectures for Automatic Speech Recognition and Synthesis; Springer: Berlin/Heidelberg, Germany, 1985; pp. 405–418.

36. Rhudy, M. Time alignment techniques for experimental sensor data. Int. J. Comput. Sci. Eng. Surv. 2014, 5, 1–14. [CrossRef]

37. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 1096–1103.

38. Liu, Z.; Tang, B.; He, X.; Qiu, Q.; Liu, F. Class-specific random forest with cross-correlation constraints for spectral–spatial hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 2017, 14, 257–261. [CrossRef]