Development and Application of Deep Belief Networks for Predicting Railway Operation Disruptions

Olga Fink, Enrico Zio, Ulrich Weidmann

International Journal of Performability Engineering Vol. 11, No. 2, March, 2015, pp. 121-134.

Printed in India

Development and Application of Deep Belief Networks for Predicting Railway Operation Disruptions ‡

OLGA FINK[1,2*], ENRICO ZIO[3,4] and ULRICH WEIDMANN[5]

[1]*Institute of Data Analysis and Process Design, Zurich University of Applied Sciences* (ZHAW), *SWITZERLAND*
[2]*Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, USA*
[3]*Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Electricité de France (EDF) at École Centrale Paris and SUPELEC, FRANCE*
[4]*Department of Energy, Politecnico di Milano, ITALY*
[5]*Institute for Transport Planning and Systems, ETH Zurich, SWITZERLAND*

(Received on May 11, 2014, revised on September 25, 2014)

Abstract: In this paper, we propose to apply deep belief networks (DBN) to predict potential operational disruptions caused by rail vehicle door systems. DBN are a powerful algorithm that is able to detect and extract complex patterns and features in data and has demonstrated superior performance on several benchmark studies. A case study is shown whereby the DBN are trained and applied on real case study from a railway vehicle fleet. The DBN were shown to outperform a feedforward neural network trained by a genetic algorithm.

Keywords: *Deep belief networks, railway operations disruptions, discrete-event diagnostic data, door system, multilayer perceptron.*

1. Introduction

The application of data-based approaches in the field of detection, diagnostics and prognostics has been constantly growing with the increased availability of complex high-dimensional monitoring data on system states and condition. Machine learning is thereby a major family of approaches applied to data-based tasks requiring self-adaptive learning abilities.

Many different machine learning techniques have been proposed for tasks in the field of detection, diagnostics and prognostics, including support vector machines, artificial neural networks and different clustering techniques [1]. Several machine learning approaches have also been proposed for the application to railway systems [2].

Deep learning has become a promising research direction in machine learning. The algorithms with deep learning have been widely applied to many different applications, including speech recognition [3], dimensionality reduction [4] and image classification [5]. The main reason for their successful application in many different fields is their ability to extract high-order correlations and deep features in data through several layers within the network structure. This is possible due to their hierarchical structure.

One of the main advantages of DBN is that they can be applied not only to labelled data but also to sparsely or not labelled data. In the latter cases, the DBN are applied as

‡This research was performed at ETH Zurich, where O. Fink was previously employed.

*Corresponding author's email: olga.fink@zhaw.ch

auto-encoders and are trained to reproduce their own input. In this case, the output of an intermediate layer is able to provide a lower level representation of the input data. This lower-dimensional representation can be used to perform clustering tasks on a lower dimensional level [4]. The main benefit of the auto-encoder approach is that the data do not have to be labelled to perform the classification task and that the higher-dimensional features can be extracted by the layered structure of the DBN. In deep learning, higher-order correlations can be detected by extracting the patterns in the features that were extracted in the preceding layer. When applied for pure feature extraction, DBN have been shown to perform better than some more commonly applied feature extraction algorithms, such as for example principal component analysis (PCA) [4].

Several extensions have been introduced in the field of deep learning, such as convolutional deep belief networks [6].

Because data derived from monitoring and diagnostics devices usually contain high-dimensional structures and high-order correlations, deep learning has also been applied in the field of diagnostics [7], [8]. However, there is a lot of potential in applying deep learning to tasks in diagnostics and prognostics and their potential has not yet been investigated in the field of predicting disruption events of railway systems.

In this paper, we propose to apply deep belief networks (DBN) with Gaussian hidden units within the restricted Boltzman machines to extract dynamic patterns from diagnostic data and to predict the occurrence of operational disruptions of railway systems. The DBN were applied to a case study with real data from a rolling stock door system.

The remainder of the paper is organized as follows. Section 2 presents an introduction to DBN. Section 3 describes the applied case study. Section 4 describes the applied algorithm and the pre-processing techniques. In Section 5, the performance of the DBN algorithm is evaluated and compared to that of MLP trained with GA. Finally, Section 6 discusses the results and presents conclusions.

Notation

CE	Cross Entropy
“D”	Pattern belonging to the class with an impending operational disruption
DBN	Deep Belief Networks
GA	Genetic Algorithm
MLP	Multilayer Perceptron
“N”	Pattern belonging to the class without predicted operational disruptions
RBM	Restricted Boltzmann Machines
TN	True Negatives
TNR	True Negative Rate
TP	True Positives
TPR	True Positive Rate

2. Theoretical Background

2.1 Introduction to Deep Belief Networks

Deep belief networks are a special type of artificial neural networks and comprise several layers of restricted Boltzmann machines (RBM) (Figure 1). The top two layers usually serve as associative memory of the input so that the input can be retrieved from the memory of the network [9].

Restricted Boltzmann machines comprise symmetrically connected neuron-like units, composed to a visible and a hidden layer. RBM, generally, learn in an unsupervised way

and are applied to extract features in data. They are often used in combination with other algorithms. However, they have also been applied in stand-alone applications [10].

The learning process of DBN differs from the unsupervised learning process used for RBM. The learning process comprises two steps. In the first step, the network extracts

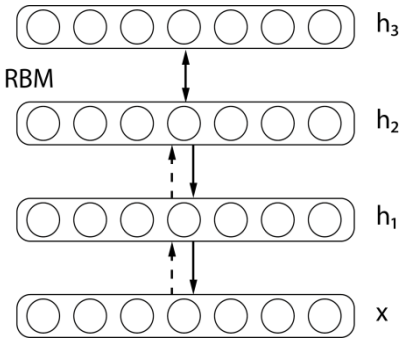

Figure 1: Deep Belief Network Composed of Several Layers of RBM

the features layer by layer in an unsupervised way and the output of one layer serves as the input to the next layer [11]. Thereby the network learns to extract features from those features extracted in the previous layer. Thereby it is able to detect high-order correlations within the data by extracting layer by layer deeper features.

In the second step, after all the layers have been pre-trained, the weights between the single layers are fine-tuned in a supervised way. For the supervised learning a back-propagation learning algorithm is applied [11]. Back-propagation is known for its shortcomings to find globally optimal solutions and for its computational inefficiency. However, applying it to a pre-trained network accelerates the learning process because the weights are only fine-tuned in the back-propagation learning. Therefore, the typical disadvantages do not apply for this application [11].

2.2 Basic Concepts of Restricted Boltzmann Machines

Restricted Boltzmann machines are also referred to as stochastic neural networks. They consist of two layers: a visible and a hidden layer (Figure 2). Each unit in the visible layer is connected to all units in the hidden layer and vice versa. However, the units within one layer are not interconnected, only between the layers. Therefore, the networks are called “restricted”. This restriction simplifies the learning process. The visible layer contains the input parameters; the hidden layer contains the latent parameters that the networks learn [11]. The hidden layer learns to model the distribution of the visible layer of variables.

Hidden units in RBM learn the structure and features that are contained in the data. The extracted features can be, for example, parameters that influence the input data and cause a higher-order correlation between the input dimensions but cannot be observed or measured.

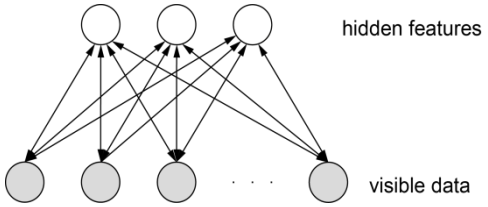

Figure 2: Network Structure of a Restricted Boltzmann Machine

The theoretical concepts and the training algorithms for the RBM were derived from [11], [12], [13], [14], [15] and are presented in the following.

A RBM network consists of visible and hidden units. Hidden units can be either modeled as binary or as Gaussian units [12]. In this research, Gaussian units were used.

A joint configuration $(\mathbf{v}, \mathbf{h})$ of the visible and hidden units has an energy given by

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i \in visible} a_i v_i - \sum_{j \in hidden} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \,, \tag{1}$$

where v_i , h_j are the binary states of visible unit i and hidden unit j, a_i , b_j are their biases and w_{ij} is the weight between them. The network assigns a probability to every pair of a visible and a hidden vector via the following function:

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})} \,, \tag{2}$$

where the *partition function Z* is a normalization term obtained by summing over all possible pairs of visible and hidden vectors:

$$Z = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \tag{3}$$

The probability that the network assigns to a visible vector, $\mathbf{v}$, is given by summing over all possible hidden vectors:

$$p(v) = \frac{1}{Z} \sum_{h} e^{-E(\mathbf{v},\mathbf{h})} \tag{4}$$

Given a training set of state vectors (the data), learning consists of finding the pertinent parameters (weights and biases) that define a Boltzmann distribution in which the training vectors have high probability.

$$\frac{\partial \log p(\mathbf{v})}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \,, \tag{5}$$

where $\langle \cdot \rangle_{\text{data}}$ is an expected value in the data distribution and $\langle \cdot \rangle_{model}$ is an expected value when the Boltzmann machine is sampling state vectors from its equilibrium distribution. This leads to a very simple learning rule for performing stochastic steepest ascent in the log probability of the training data:

$$\Delta w_{ij} = \epsilon \frac{\partial \log p(\mathbf{v})}{\partial w_{ij}} = \epsilon \left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \right) \,, \tag{6}$$

where ε is a learning rate.

Because there are no direct connections between hidden units in an RBM, it is very easy to get an unbiased sample of $\langle v_i h_j \rangle_{data}$. Given a randomly selected training pattern, **v**, the binary state, h_j, of each hidden unit, j, is set to 1 with probability

$$p(h_j = 1|\mathbf{v}) = \sigma\left(b_j + \sum_i v_i w_{ij}\right), \tag{7}$$

where $\sigma(x)$ is the logistic sigmoid function $1/(1 + exp(-x))$. $v_i h_j$ is, then, an unbiased sample.

Because there are no direct connections between visible units in an RBM, it is also very easy to get an unbiased sample of the state of a visible unit, *given a hidden vector*

$$p(h_j = 1|\mathbf{h}) = \sigma\left(a_j + \sum_j h_j w_{ij}\right) \tag{8}$$

To get an unbiased sample of $\langle v_i h_j \rangle_{model}$ an approach to reconstruct $\langle v_i h_j \rangle_{model}$ has been proposed. In this case, $\langle v_i h_j \rangle_{model}$ is replaced by $\langle v_i h_j \rangle_{recon}$. In this case, firstly, the states of the visible units to a training vector are set; secondly, the binary states of the hidden units are computed by applying Equation 7. Once binary states have been chosen for the hidden units, a reconstruction is produced by setting each v_i to 1 with a probability given by Equation 8. Therefore, an incremental learning process is required.

For Gaussian hidden units, Equation 1 is adjusted to:

$$E(\mathbf{v}, \mathbf{h}) = -\sum_i a_i v_i - \sum_j \frac{(h_j - b_j)}{2\sigma_j^2} - \sum_{i,j} v_i \frac{h_j}{\sigma_j} w_{ij}, \tag{9}$$

where σ_j^2 is the variance of the hidden unit j. The variances of all hidden units are usually fixed to $\sigma_j^2 = 1$.

3. Case Study

The DBN was developed and applied to real case study based on diagnostic discrete event data from a European railway fleet consisting of 52 train sets, whose composition was mixed between 9-coach- and 11-coach-train-sets. The available observation period was 313 days (approximately ten months).

The information recorded by the diagnostic system of the considered trains is discrete events. In contrast to monitoring and diagnostic systems that observe predefined state parameters continuously, discrete event diagnostic systems only trigger events if predefined conditions occur. In some cases, the state parameter is observed continuously. However, an event is only recorded by the diagnostic system if the observed parameter exceeds a defined threshold. Thereby, the evolution of the state condition in time is represented by a time series of discrete events.

The information content is significantly reduced, compared to continuously monitored parameters. Therefore, usually, the sequence of state transitions or event occurrences of a single state or event will give an insufficient representation of the overall system condition. Yet, if several state transitions and event occurrences are monitored, the system state can be described in a more complete way [16]. Consequently, we assume that the combination of a large number of observed state transitions and event occurrences provides a sufficiently complete representation of the system state. Furthermore, we

assume that performed maintenance is reflected in the frequency of the occurring diagnostic events.

For each system, a set of diagnostic events is defined during the design and development process. There are 261 distinct event codes for the door system, which was used as a case study in this research. Each time an event is recorded, a set of complementary parameters and states is additionally logged by the system. These parameters either give information on the environmental condition, on the condition of other systems or subsystems that may affect the operation. Additionally, the complementary parameters can give information on the condition or the mode of operation of the entire system. The complementary information is required to embed the specific isolated events in the context of the overall system and its operating and environmental conditions. The information supports the maintenance crew to resolve the failure or malfunction. The number and character of the complementary parameters and system states are predefined for each specific event in the design process of the diagnostic system.

There are two main types of diagnostic events: those potentially affecting operation and consequently communicated to the driver; and those on the evolution of the system condition, used by the maintenance crew for fault finding or resolving of failures.

For door systems, the events that could potentially induce operation disruption include inability to retract the steps, to open the door or fully close it prior to departure. If a door fails to open, it reduces passenger flow, causing extended dispatching time and thereby also potential delays to operation. Furthermore, for example, if a sliding step, designed to bridge the gap between train door and platform, cannot be retracted, the train will not be permitted to depart. In this case, the sliding step must be retracted manually, causing a delay. A door that fails to close prior to departure has the same effect on schedule adherence. Similar to the malfunction of the sliding step, the door would have to be closed manually by the train conductor or by the train driver [16].

The diagnostic events of the condition of the system include the occurrence of such states in which the system deviates from normal operation. This is, for example, the case when the speed of the door closing process is slower than a specified threshold. A diagnostic event is also recorded when a sliding step needs longer than the specified time to retract. Further diagnostic events include push button failing to be activated or a door being constantly activated. Diagnostic events are also recorded if a subcomponent fails. This information is intended to facilitate the maintenance personnel fault and failure finding and to isolate failure causes. Additionally, the usage profiles of each door can be deduced from the diagnostic data (the points in time when the doors are enabled on train arrival are logged).

Not all of the events that were defined as requiring a warning to be sent to the driver and a specific type of intervention were relevant to the prediction task. Two criteria were used to select the events that could potentially cause operational disruptions. Firstly, because predicting an impending disruptive event is only useful if the failure's root cause can be determined and preventive maintenance actions can be taken to address the problem before failure, only those events that could be connected to specific door subsystem technical malfunctions were analysed. Operational disruption events caused by several technical components or by external factors were not analysed. Secondly, for some of the distinct operational disruption events, the datasets were too small to generate statistically significant prediction results. Using these two criteria, one relevant operational disruption event type for further analysis was selected [16].

The event codes for the door system indicate the specific door affected by the event. For instance, there can be four different codes for one type of event (one for each door in the coach). In this research, the allocation of an event to a specific door system is performed in the structure of the data and not in the coding of the events. Therefore, it was possible to reduce the 261 codes to 72 distinct events [17].

4. Defining the Prediction Problem and the Applied Algorithm

4.1 Defining the Prediction as a Binary Classification Task

The prediction problem considered in this research was defined as a binary classification task. The prediction horizon in which the event is predicted to occur was set to seven days. While the time period of seven days might appear imprecise, it is sufficiently precise for practical purposes. The time period of one week enables the operators to schedule the required maintenance task and to anticipate the occurring failure and the operational disruption. This simplification increases the flexibility of the algorithm and enables the use of discrete event diagnostic data for prediction purposes.

The following two classes were defined to predict the occurrence of disruption events:

- Class “D”: impending operational disruption due to a failure of the door system within the prediction horizon
- Class “N”: no occurring operational disruption events caused by the door system within the prediction horizon.

For each of the 72 distinct diagnostic events, the input data patterns represent the time elapsed from the specific observation time point to the previously occurred event. Although this approach neglects the density of the occurring events, it enables us to integrate information on the time series of the occurring events in the input patterns.

The observation time window was set at four weeks because this was considered sufficiently long to observe the evolution of the diagnostic events and their influence on the state of the system. By taking the time to last occurred event into consideration, also the events occurred before the observation time window were included.

The input patterns were derived by moving the observation time window over the 313- day study period one day at a time. This approach enabled the observation of small changes in the system condition and also provided a sufficient number of input patterns for the defined algorithm to extract the evolving features. One of the consequences of this approach is that the data patterns can show high similarities. However, these similarities do not only occur within one class, but also between the classes. Similar patterns will be located close to the decision boundary which is difficult to derive. Therefore, to discriminate between these patterns and to generalize the decision boundary, the classification algorithm must possess very good classification abilities.

4.2 Pre-Processing of Input Data

The value ranges are different for each of the distinct diagnostic events. To balance the influence of each of the input signals on the classification task, the input signals were normalized to have a zero mean and a unit standard deviation. This approach preserves the variance contained in the data and ensures a comparable value range for all the input signals.

Compared to the number of patterns not resulting in a disrupted operation, the number of diagnostic events that can cause operational disruptions is small. Therefore, the applied dataset was highly unbalanced. When algorithms learn from imbalanced datasets,

the penalty for misclassifying the under-represented class may be too low for the algorithm to learn to extract these patterns. This may result in weak generalization ability. Several approaches have been proposed to deal with unbalanced datasets [18]. In this research, the dataset was balanced by omitting parts of the data patterns from the "healthy" state. The same number of patterns from the "healthy" state was included in the dataset as there were from the "faulty" state. The selection of the patterns from the "healthy" state was performed randomly to avoid bias in the selection process. On the contrary, all available patterns from the "faulty" state were included in the considered dataset.

The application of this approach is based on the assumption that the data patterns from the "healthy" state included in the dataset are representative for the distribution of the whole class of "healthy" state patterns and contain sufficient information for the algorithm to extract the relevant features and to generalize them [17]. Since data patterns have a high degree of similarity, the assumption that the selected data patterns from the "healthy" state are sufficiently representative for the whole class is valid.

4.3 Applied Algorithm

In this research, RBM with Gaussian units are used to construct the DBN network structure. This is contrary to typically used binary units within the network structure of the RBM [9]. The application of Gaussian units provides more flexibility and preserves more information in the input data compared to purely binary units.

The applied DBN was composed of two restricted Boltzmann machines. Additional layers did not improve the performance of the algorithm. In the first step, the single layers of the DBN extract the features in an unsupervised way. The extracted features of one layer become the input to the next layer. Thereby, the algorithm learns to extract the deep features of the input data by the layer-wise learning approach. The layers were pre-trained for 100 epochs. After the pre-training of the single RBM, a back-propagation learning algorithm is applied to fine-tune the weights between the single layers in supervised learning. The error function applied within the back-propagation learning algorithm was the cross entropy (CE) error function [19]. The CE error function performs better than the mean squared error function both in terms of computational speed and prediction performance [20].

First, the input patterns, having a dimension of 72 distinct event types, were presented to the first RBM within the DBN network structure. Within the RBM, which itself consists of a visible and a hidden layer, the visible input is expanded to a dimension of 300 in the hidden layer. The output of the hidden layer of the first RBM becomes the input to the visible layer of the second RBM. The input of the visible layer is expanded to the dimension of 600 in the hidden layer of the second RBM (Figure 3). The structure of the algorithm was selected in a trial and error approach. The selected combination of the dimensions of the visible and hidden layer provided the best results.

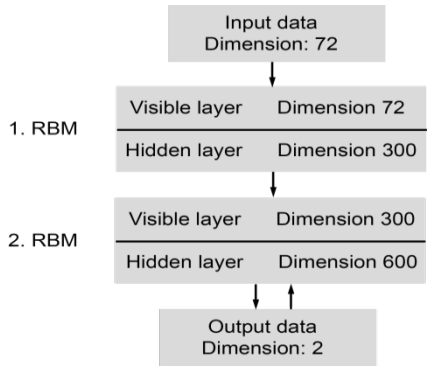

Figure 3: Structure of the Applied Algorithm

4.4 Applied Alternative Algorithm

To evaluate the performance of the proposed DBN algorithm, an alternative state of the art neural network algorithm, a multilayer perceptron [21], was applied to the same classification problem using the same input data and target output. Because back-propagation learning, which is typically applied within the MLP, suffers from several drawbacks, including local minima, genetic algorithm (GA) [22] was used as the learning algorithm to train the weights of the MLP. For the applied MLP a sufficiently complex network structure was provided with two hidden layers (40 neurons in the first hidden layer and 20 neurons in the second hidden layer). The GA was run for 20 individuals in the population over 500 generations with a crossover probability of 85%. The two-layered structure of the MLP algorithm is considered sufficiently complex to be applied to the selected problem. Also the number of neurons in each layer is considered sufficiently large to learn the classification task.

5. Performance Evaluation

5.1 Evaluating the Average Generalization Performance

The decisions based on the predictions of “healthy” or “faulty” states of the considered system are very critical and can have substantial consequences on system availability and economic impacts. Therefore, the performance of the algorithms applied within the predictions has to be tightly evaluated and monitored. Particularly, before applying a new algorithm in real world applications, its performance needs to be assessed with respect to different criteria.

Prediction precision is one of the most important performance assessment criteria. Besides selecting the performance evaluation measure, also the type of resampling needs to be selected. In this research, the holdout approach was selected, which does not require any additional re-sampling [23]. Holdout technique is usually applied if the dataset is sufficiently large and the algorithm shows a stable performance. The dataset is subdivided into a training and a testing dataset. The size of the subsets depends on the size of the entire available dataset. The size of the testing dataset was set to 10% of the entire dataset. The training and the testing subsets were assumed to be representative for the underlying data distribution of the entire dataset. Furthermore, it was assumed that both

subsets are independent. The training dataset consisted of 1220 data patterns and the testing dataset of 136 patterns.

In the first step, as an initial criterion to estimate the generalization ability of the applied algorithms, the misclassification rate was used. The misclassification rate is defined as the ratio of the sum of all the incorrectly classified patterns to all patterns in the considered dataset (either training or testing dataset). This metric computes the rate of patterns that were misclassified by the algorithm without specifying from which class they were misclassified [24]. The value achieved by the DBN algorithm was 3.7%: this means that 96.3% of the testing data patterns were classified correctly.

5.2 Evaluating the Generalization Performance of MLP

The same generalization measure was also applied to evaluate the performance of the alternatively applied algorithm, the MLP trained with GA. Contrary to the good generalization ability achieved by the DBN, the misclassification rate achieved by the MLP on the testing dataset was 43%. This performance is in the range of the performance of a random classifier (where data patterns are assigned randomly to the classes).

Generally, an insufficient performance on the testing dataset can be due to overfitting. In overfitting, a significant performance drop between the training and the testing dataset can be observed. This occurs if the algorithm is not able to generalize the extracted patterns from the training to the testing data and only memorizes the patterns presented during the training procedure. However, in this research the misclassification rate was similar between the training and the testing dataset. This means that the weak performance has not been caused by overfitting, but rather by the similarity of the data patterns and the inability of the feedforward algorithm to discriminate between the patterns from both classes.

A possible explanation for this unsatisfactory performance of the MLP is that the algorithm is not able to extract the features that are contained in the data. The MLP would have assumably required an additional pre-processing step in which a feature extraction algorithm would have been applied. However, to compare the performance of both algorithms, the same input data were applied for the DBN as for the MLP algorithm. The DBN algorithm was obviously able to extract the features in the data and to further process these features in a supervised learning process.

5.3 Sensitivity and Specificity Evaluation

While the misclassification rate solely measures the average accuracy of the predictions, for binary classifications, the metrics of sensitivity and specificity can be used to assess the classification performance of the algorithm within the classes [24]. The patterns with the property of interest are categorized as “positives” and patterns from the other class as “negatives”. In this study, the patterns of interest are those corresponding to disruption events (patterns from class “D”).

Sensitivity measures the ability to identify the positives [24] and corresponds to the true positive rate (TPR), which is computed as the ratio of correctly classified positive patterns to all the positive patterns in the dataset:

$$TPR = \frac{TP}{TP + FN} \quad , \tag{10}$$

where TP are the true positives and FN are the false negatives.

Specificity measures the ability to identify the negatives [24] and corresponds to the true negative rate (TNR), which is computed as the ratio of the correctly classified negative patterns to all the negative patterns in the dataset:

$$TNR = \frac{TN}{TN + FP} \quad , \tag{11}$$

where TN are the true negatives and FP are the false positives.

There is usually a trade-off between the sensitivity and specificity. Therefore, it is possible to adjust the weights between the two measures. The specific weights depend on the criticality and the costs of FP (*e.g.*, replacing components that may not fail) and FN *(e.g.*, operational disruptions that were not anticipated).

The DBN algorithm achieved a sensitivity of 95.7% and a specificity of 97.0%. Although there is a marginal difference between the sensitivity and the specificity, the algorithm is not biased towards either of the two classes and learned to discriminate between the two classes.

6. Conclusions and Discussion

This paper proposes the application of deep belief networks for predicting the occurrence of potential railway operation disruption events. The applicability of the approach was demonstrated by a case study based on real discrete-event diagnostic data from door systems of a railway rolling stock fleet. The prediction results obtained from the case study confirm the suitability of the proposed approach and show a good prediction precision.

The performance of the DBN was compared to that of a more commonly applied machine learning algorithm, a MLP trained with a GA. The applied MLP was not able to discriminate between the patterns from both classes, and only achieved a prediction performance in the range of a random classifier. A possible explanation for this performance is that the DBN was able to extract the features within the input data while the MLP would have required an additional feature extraction step before performing the classification task.

The prediction task was defined as a binary classification. The patterns were classified as indicating a disruption event in the next seven days or as being in a healthy condition. This prediction approach was found to be sufficiently precise for the operators to be able to anticipate the prediction and schedule a pertinent maintenance task prior to the occurrence of the event. However, it is possible, to extend the definition of the problem to a multiclass classification. This extension would provide additional information on intermediate degraded states and would provide more flexibility to the operators to monitor the evolution of the system condition and to anticipate the failures.

In this research, the input patterns measured the time elapsed from the last occurred event. In a previously conducted study, we have defined the input patterns by the density of occurring events. Combining the complementary information on the input patterns is subject for further research.

In this case study, DBN were used for a classification task with labeled data. However, it has been shown that DBN are particularly powerful when labeled data are either sparse or not available at all [4]. In these cases, DBN are applied as auto-encoders. The lower dimensional representations of the input data can then be used for clustering tasks, which do not require prior information of the actual condition of the system. Applying the DBN as auto-encoders and extracting the lower-dimensional representations

of the input data that can be used for unsupervised clustering approaches is therefore subject for further research.

The network structures applied in deep belief networks are usually more complex and have more layers compared to the structures applied in the current study. In this case study, the deep belief networks are not used to their maximum potential since the input data are not as complex and, thus, do not require as many lower level features as other applications of DBN such as extracting features in images [4] and in speech recognition [3]. Therefore, there is potential to extend the application of DBN in the field of railway disruption prediction based on more complex input data that contain more hidden structure.

With respect of the relevance of the research for practical applications, the proposed approach can be applied to forestall railway operational disruption events. This would enable a proactive maintenance regime, which does not only increase the operational reliability but also decreases the maintenance costs by enabling planning and scheduling of the required maintenance activities.

Acknowledgements: The participation of Olga Fink to this research is partially supported by the Swiss National Science Foundation (SNF) under Grant No. 205121_147175.

The participation of Enrico Zio to this research is partially supported by the China NSFC under Grant No. 71231001.

The authors would like to thank Alstom Transportation for providing the data for this research project.

References

[1]. Jardine, A.K.S., D. Lin, and D. Banjevic. *A Review on Machinery Diagnostics and Prognostics Implementing Condition-based Maintenance*. Mechanical Systems and Signal Processing. 2006; 20(7): 1483–1510.

[2]. Smith, A. E., D.W. Coit, and L. Yun-Chia. *Neural Network Models to Anticipate Failures of Airport Ground Transportation Vehicle Doors.* Automation Science and Engineering, IEEE Transactions on. 2010; 7(1):183–188.

[3]. Dahl, G. E., D. Yu, L. Deng, and A. Acero. *Context-dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition*. IEEE Transactions on Audio, Speech, and Language Processing, 2012; 20(1):30–42.

[4]. Hinton, G. E., and R. R. Salakhutdinov. *Reducing the Dimensionality of Data with Neural Networks.* Science. 2006; 313(5786):504–507.

[5]. Ciresan, D., U. Meier, and J. Schmidhuber. *Multi-Column Deep Neural Networks for Image Classification.* In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE; 2012. p. 3642–3649.

[6]. Lee, H., R. Grosse, R. Ranganath, and A. Y. Ng. *Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations*. In: Proceedings of the 26th Annual International Conference on Machine Learning. ACM; 2009. p. 609-616.

[7]. Tamilselvan, P., and P. Wang. *Failure Diagnosis using Deep Belief Learning based Health State Classification*. Reliability Engineering & System Safety, July 2013; 115:124–135.

[8]. Tran, V.T., F. AlThobiani, and A. Ball. *An Approach to Fault Diagnosis of Reciprocating Compressor Valves using Teager Kaiser Energy Operator and Deep Belief Networks.* Expert Systems with Applications. 2014; 41(9):4113–4122. Available from: http://www. sciencedirect.com/science/article/pii/S0957417413010014.

[9]. Hinton, G. E., S. Osindero, and Y. W. Teh. *A Fast Learning Algorithm for Deep Belief Nets*. Neural Computation. 2006; 18(7):1527–1554.

[10]. Larochelle, H., and Y. Bengio. *Classification using Discriminative Restricted Boltzmann Machines*. In: Proceedings of the 25th international conference on Machine learning. ACM; 2008. p. 536–543.

[11]. Ackley, D. H., G. E. Hinton, and T. J. Sejnowski. *A Learning Algorithm for Boltzmann Machines*. Cognitive Science. 1985; 9(1):147–169.

[12]. Taylor, G.W., and G. E. Hinton. *Factored Conditional Restricted Boltzmann Machines for Modeling Motion Style.* Proc. 26th International Conference on Machine Learning, June 2009, pp 1025-1032. Omnipress, Montreal, Quebec.

[13]. Salakhutdinov, R., A. Mnih, and G. Hinton. *Restricted Boltzmann Machines for Collaborative Filtering.* In: Proceedings of the 24th International Conference on Machine Learning, 2007, pp. 791-798.

[14]. Zeiler, M. D., G.W. Taylor, N. F. Troje, and G. E. Hinton. *Modeling Pigeon Behaviour using a Conditional Restricted Boltzmann Machine.* In: 17th European Symposium on Artificial Neural Networks (ESANN); 2009.

[15]. Fink, O., E. Zio, and U. Weidmann. *Predicting Time Series of Railway Speed Restrictions with Time-dependent Machine Learning Techniques*. Expert Systems with Applications.2013; 40(15):6033–6040.
Available from:
http://www.sciencedirect.com/science /article/pii/S0957417413002868.

[16]. Fink, O. *Failure and Degradation Prediction by Artificial Neural Networks: Applications to Railway Systems.* ETH Zurich; 2014.

[17]. Fink, O., A. Nash, and U. Weidmann. *Predicting Potential Railway Operational Disruptions with Echo State Networks.* Transportation Research Record. Journal of the Transportation Research Board. 2013; 2374(1):66–72.

[18]. Duda, R.O., P. E. Hart, and D. G. Stork. *Pattern Classification*. 2nd ed. New York: John Wiley; 2001.

[19]. Rubinstein, R. Y., and D. P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning.* Springer; 2004.

[20]. Kline, D., and V. Berardi. Revisiting Squared-Error and Cross-Entropy Functions for Training Neural Network Classifiers. Neural Computing & Applications. 2005; 14(4):310–318.

[21]. Hornik, K., M. Stinchcombe, and H. White. *Multilayer Feed-Forward Networks are Universal Approximators*. Neural Networks. 1989; 2(5):359–366.

[22]. Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Publishing Co. Inc., Reading, Massachusetts , 1989.

[23]. Kohavi, R. *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection.* In: International Joint Conference on Artificial Intelligence (IJCAI). Morgan Kaufmann; 1995. p. 1137–1145.

[24]. Japkowicz, N., and Mohak Shah. *Evaluating Learning Algorithms: A Classification Perspective.* Cambridge University Press; Cambridge, 2011.

Olga Fink is currently a lecturer at the Zurich University of Applied Sciences (ZHAW), Switzerland. She is also research affiliate at the Massachusetts Institute of Technology (MIT), Boston. She received the Ph.D. degree in civil engineering from ETH Zurich, Switzerland in 2014 and Diploma degree in industrial engineering in 2008 from Hamburg University of Technology, Germany. Her research focuses on reliability, detection, diagnostics and prognostics, predictive maintenance, life cycle costing, machine learning and data mining techniques, particularly applied to transportation and infrastructure systems.

Enrico Zio is currently Director of the Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Electricité de France (EDF) at École Centrale Paris and École Superieure d'Électricité (SUPELEC), France and full professor at Politecnico di Milano, Italy. He received the Ph.D. degree in nuclear engineering from Politecnico di Milano and MIT in 1995 and 1998, respectively. He received his M.Sc. degree in mechanical engineering in 1995 from the UCLA and BSc in nuclear engineering in 1991 from the Politecnico di Milano. His research focuses on the characterization and modeling of the failure/repair/maintenance behavior of components, complex systems and their reliability, maintainability, prognostics, safety, vulnerability and security, Monte Carlo simulation methods, soft computing techniques, and optimization heuristics.

Ulrich Weidmann is a full professor for Transport Systems at the Institute for Transport Planning and Systems at ETH Zurich. He is Head of the Department of Civil, Environmental, and Geomatic Engineering at ETH Zurich. He received the Ph.D. degree in 1994 in Civil Engineering and his M.Sc. degree in 1988, both from ETH Zurich. From 1994 to 2004 he held several positions within the Swiss Federal Railways (SBB), including top management positions. His research focuses on passenger transport system evaluation and decision support, integration of rail freight transport systems in logistic chains, system performance and stability, and rail infrastructure management.