Predicting potential railway operational disruptions with echo state networks

Olga Fink[a,*], Andy Nash[b], Ulrich Weidmann[a]

[a]*Institute for Transport Planning and Systems, ETH Zurich, Zurich, Switzerland*
[b]*Vienna Transportation Strategies, Vienna 1070, Austria*

Abstract

European passenger rail systems are massively interconnected and operate with very high frequency. The impacts of single component failures on these types of systems can significantly affect technical and operational reliability. Today advanced diagnostic tools with broad functionalities are being added to systems and system components. These tools monitor, control the operation and support the maintenance of the highly sophisticated and interconnected components. This paper presents an approach for using a set of diagnostic event data from a passenger train exterior door system to predict the occurrence of events that might evolve into operational disruptions that impact train operation and therefore railway reliability. This approach uses a neural network algorithm with a dynamic temporal behavior (the echo state network) in combination with principle component analysis. The proposed approach exhibits a prediction accuracy of up to 99%.

1. Introduction

The reliability and stability of public transport service are among the most significant factors influencing mode choice. Unreliable service also causes higher operating costs and reduces revenue. In railway systems overall reliability is highly determined by the technical reliability of infrastructure and rolling stock. In railway systems with high frequency service, small operational disruptions can lead to cascading events throughout the network. Anticipating and preventing these disruptions can significantly improve reliability. This can be achieved by predicting impending disruption events caused by technical failures of single components or systems. Data generated by remote diagnostic systems provides an insight into the processes occurring in the single components. This data can be used to detect faults or failures, to diagnose them and to predict the considered systems remaining useful life (Vachtsevanos, 2006). In order to use diagnostic data effectively at least one parameter (which can be measured and processed cost-efficiently) must exist that directly describes the systems condition or the system condition must be able to be derived indirectly from measurable parameters based on known physical relationships.

Existing research on railway subsystem reliability using diagnostic data focuses on fault detection, predicting component failures and remaining useful life (Fararooy and Allan, 1995), (Lehrasab et al., 2002), (Roberts and Silmon, 2012). Two studies considered door systems specifically and used different types of neural networks to detect faults or degraded performance (Lehrasab et al., 2002), (Smith et al., 2010). In both studies continuously measured diagnostic data were used as indicators allowing inference on the performance of the considered systems. Example indicators are electrical current and voltage signals from the open/close cycles, closing and opening times, pressure, velocity, and airflow. In Smith et al. (2010) the input parameters were used in a regression model to derive level of performance. In Lehrasab et al. (2002) a combination of methods was applied to classify performance indicators and to indicate faults. The classification accuracy was between 66% and 90%. The network types applied in one or both studies were Multilayer Perceptrons (MLP) with different network design principles (such as cascade correlation), Radial Basis Functions (RBF) and Self-Organizing-Maps (SOM).

In addition to the continuously measured diagnostic data, the sequence of occurring predefined events and their determining parameters can also be recorded automatically. The recorded discrete events can

*Corresponding author; Email: ofink@ethz.ch; Tel: +41 44 633 27 28; Postal Address: ETH Zurich, Wolfgang-Pauli-Str. 15, 8093 Zurich, Switzerland

range from confirming an enabled function to indicating that a signal exceeds defined limits. Some of these events are used to warn the train driver or to assist the maintenance crew in fault finding and corrective actions. In all the referenced studies input signals were applied that can be directly or indirectly used to deduce the performance of the considered systems. However the event-based diagnostic data have, to the knowledge of the authors, not yet been applied for reliability prediction in railway systems.

In this research automatically generated diagnostic event data from passenger train door systems was used to derive patterns that were then applied to predict potential railway system disruption events.

The prediction was performed using echo state networks, a special type of artificial neural networks. Neural networks recognize and generalize functional relationships from representative data samples. They have been applied to many types of problems and have shown great potential in the field of reliability prediction. However, few studies have applied neural networks for reliability prediction in railway rolling stock systems (Lehrasab et al., 2002), (Smith et al., 2010), (Fink and Weidmann, 2011). In particular, discrete event diagnostic data have not yet been applied in this field for prediction and prognostic purposes. Echo state networks have shown superior performance in several benchmark studies (Jaeger and Haas, 2004) on classification and signal generation problems, but have not yet been applied to reliability prediction problems. The research questions addressed in this paper are:

1. Is it possible to predict disruption events on a component level based on automatically generated diagnostic event data?
2. Are neural networks and specifically echo state networks a suitable method to perform this task?

The next section describes neural networks and how they were used to develop the proposed algorithm to predict impending diagnostic events with the potential for causing railway operational disruptions. Section 3 presents results of applying the algorithms to the passenger train door system diagnostic data. Finally, section 4 discusses results and makes recommendations for further research.

2. Approach to predict railway operational disruptions with ESN

2.1. Neural Networks

An artificial neural network is the general term for a series of data-based methods with self-adaptive properties. Neural networks take their name from the human brain cell networks. They imitate selected features of the brains learning process. Their algorithms enable computers to deduce functional relationships from data. Neural networks consist of basic (mostly) nonlinear processing units (referred to as neurons) connected by information storing entities (referred to as weights) (Haykin, 2009). By training the values of these weights based on exemplary input-output mappings, the desired behavior is induced into the networks.

Neural networks have been applied for many purposes in many fields. They are able to recognize and generalize functional relationships from representative data samples without prior knowledge of the specific mechanisms at work or any additional assumptions (Haykin, 2009). They can then transfer the patterns and rules extracted to new input data, and are thus able to predict future behavior.

Neural networks can be subdivided into feedforward and recurrent neural networks. In feedforward networks input signals are propagated in only one direction: from input to output (Haykin, 2009). In recurrent neural networks signals are propagated in both directions since the synapse connections have cyclic forms (with at least one cyclic connection) (Mandic and Chambers, 2001). Recurrent network structures have dynamic temporal behavior.

A specific type of recurrent neural network called the echo state network (ESN) algorithm was used in this research.Similar to other recurrent neural networks, ESN are able to exhibit dynamic temporal behavior and have a memory. ESN are typically applied for modeling complex dynamic systems.

Echo state networks have demonstrated remarkable results in several benchmark studies (Jaeger and Haas, 2004). They have been applied for practical applications, such as iterated prediction of time series, signal classification and dynamic pattern recognition tasks (Verstraeten, 2009), but have not yet been applied in the field of reliability prediction.

In contrast to many neural networks, the main structural element of ESN is a reservoir rather than a layered structure. Given this structure ESN belong to the class of reservoir computing. The ESN reservoir contains a defined number of randomly and sparsely connected neurons.

For an ESN the training process works by presenting input signals ("training signals") into the reservoir to induce nonlinear responses. The single neurons exhibit an "echoed" response to the training signal and generate a variation or a transformation of the induced training signal. Subsequently, a desired output signal is determined by a trainable linear combination of all the generated response signals. In supervised learning a teacher signal is fed back to the reservoir (Jaeger, 2005). The process is illustrated schematically in Figure 1.

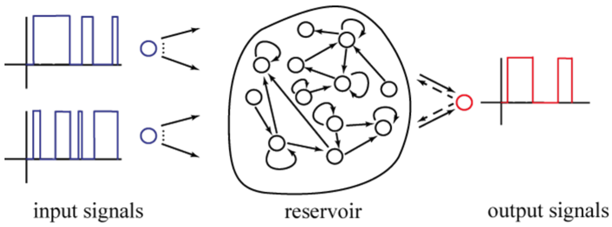

Figure 1: Functional Principle of Echo State Networks, Based on (Verstraeten, 2009)

The ESNs main difference from other neural networks is that only the weights of the reservoir output signals are trained. The weights of the connections within the reservoir are not trained but are generated randomly (Jaeger, 2005). This approach significantly reduces the learning process compared to other algorithms (e.g. backpropagation through time) (Jaeger, 2005). One of the major properties of the echo state network is the so-called echo state property. This property ensures that the network has a fading memory and thus that the influence of initial conditions, which are randomly generated, diminishes asymptotically. Therefore, at the end of the training process only learned relationships influence the output (Jaeger, 2005).

The activation function of the neurons inside the reservoir is the standard sigmoid activation function but other activation functions can also be applied. One drawback of the sigmoid activation function is that it is not possible to control the dynamics of the processes since it does not have a time constant. However, leaky integrator neurons (which can incorporate different time dependent behavior) can be included in the reservoir to address this problem (Jaeger, 2005).

Neural networks have several parameters which determine their performance and which are used to adjust the network structure and performance to the specific problem and its complexity. For example, in the case of Multilayer Perceptrons, the performance is determined by the structure of the network, in particular the number of layers and number of neurons in the layers, but also by the learning rate and the momentum (Haykin, 2009).

Echo state networks have several parameters that can be adjusted to influence their performance. Since the main structural element in ESN is the reservoir, the reservoir size is one of the main influencing parameters. The reservoirs function is to expand the nonlinearly of the input and to memorize it. The reservoir connections are usually sparse, meaning that nodes within the reservoir are only connected to a small number of other nodes (Lukosevicius, 2012). The sparsity and the spectral radius of the reservoir significantly influence reservoir performance. The spectral radius determines the width of the distribution of its nonzero elements (Lukosevicius, 2012). In addition to these parameters the scaling of the input and the scaling of the bias also influence reservoir performance. The dynamic behavior of the reservoirs can be adjusted by applying leaky integrator neurons; these have a time constant which is a further parameter that influences overall performance (Lukosevicius, 2012). As ridge regression is applied to the outputs of the reservoir, the regularization coefficient is also a crucial parameter influencing the overall performance and reducing the possibility of overfitting.

2.2. General Procedure of the Proposed Approach

Given the critical role that data plays in the process of developing and using neural networks it is not surprising that data pre-processing is critical. Figure 2 illustrates the steps involved in pre-processing data and developing the ESN algorithm used in this research. The rest of this section describes this process.

2.3. Study Data

The research goal was to see if it was possible to use automatically collected diagnostic data from a railway system component to predict the occurrence of diagnostic events with the potential to disrupt railway operations. The research focused on train doors because, while they do not have the most failures, they are extremely important to operational reliability since failures can directly influence operations and

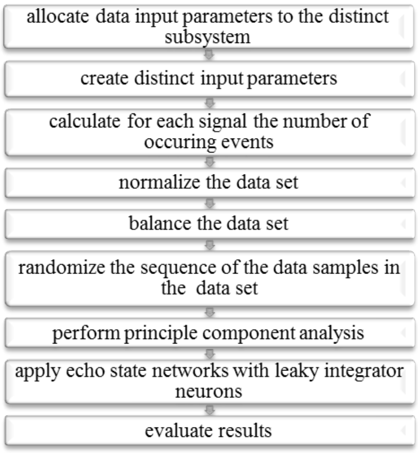

Figure 2: Developing a Neural Network Prediction Algorithm for Occurrence of Potential Operational Disruptions

schedule adherence and there are no redundant systems on the functional level. Door systems can directly or indirectly lead to delays in railway operations. 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. Therefore, anticipating failure of door systems can prevent operational disruptions and service delays and thereby lead to improved reliability. The diagnostic event data applied in this study was generated from a railway fleet consisting of 52 trainsets, each with nine cars having two doors on each side. The analysis period was 313 days (approximately ten months). Given the large fleet size, the data are considered sufficient to demonstrate the approachs feasibility. Data were collected automatically by event recorders. These begin recording parameters when a predefined diagnostic event occurs. The number and character of parameters recorded depends on the affected system. The parameters include speed, outside temperature, overhead line voltage etc. The recorded events are always assigned a time stamp, train location (i.e. train number, car number) and actual location via GPS. Diagnostic events fall into one category:

- Driver action required – high priority;
- Driver action required – low priority;
- Driver information;
- Maintenance.

Depending on the category, the corresponding events will be communicated to the driver or the maintenance crew. High priority diagnostic events are those that can result in a delay-causing event.

There are 261 distinct event codes for the door system. The codes indicate the event type and door affected. For instance, there are four different codes for one type of event (one for each door in the car). 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, 12 of which require a high priority driver action.

For door systems, high priority events that require driver action include inability to retract the steps or fully close the door prior to departure. The relevant maintenance signals include information on deviations from the normal operation of various door system components and subsystems. Additionally, the usage profiles of each door can be deduced from the diagnostic data if the points of time when the doors are enabled on train arrival are logged.

In this research, no information was available on the maintenance actions, age or actual condition of the components. Therefore it was assumed that all differences are reflected in the occurrence of the diagnostic events and that due to the size of the fleet, the dataset covers all relevant combinations of different parameters.

2.4. Data Preprocessing

The prediction problem considered in this research was defined as a classification task with aggregated dynamic patterns. Two time periods were defined: an observation time and a prediction time. This methodological approach is flexible and the time periods can be adjusted depending on the users needs. The observation time window was set at four weeks because this was considered sufficiently long for different diagnostic event patterns to evolve and given the amount of available data (ten months). The prediction time (the period to anticipate and prevent impending disruption events) was set at seven days. The four-week observation time window was used to predict the occurrence of a high priority event within the subsequent seven days. A high priority event was defined as having the potential to affect railway operations.

In order to increase the selectivity of the algorithms, only two classes were defined. This is a common approach in classification. Even if there are more than two classes, the algorithms often learn to distinguish one class from all other classes, which are grouped for the algorithm training process to one single class (Duda et al., 2001).

In this research a dynamic pattern was classified as belonging to class ”H“ (impending high priority event), if within the next seven days, starting from the selected time point, at least one of the specified high priority diagnostic events would occur. If no specified high priority events occurred during the prediction period, the time pattern was classified as belonging to class ”N“ (no occurring events). Thus, it is a binary classification task with only two classes.

It is important to understand that the classification does not distinguish when exactly the event will occur, only that it will occur within the next seven days. While this might appear imprecise, it is sufficiently precise for practical purposes and was accepted by the expert group advising this research. This simplification increases the algorithms flexibility.

The discrete event diagnostic signals (input parameters are normally called signals) were condensed to event densities, which were attributed to the distinct signals and distinct doors within the total system. In order to cover all possible combinations of diagnostic event density and generate a sufficient number of input signals, the data-patterns were generated by moving a four-week window over the 313-day study period one day at a time. As a consequence the time periods overlap and the data patterns can show high similarities.

Generally, algorithms in classification tasks tend to overweight input signals with a larger range of values than other input signals. To avoid this overweighing the input signals are normalized.

In this research normalization was not performed across the entire dataset but rather was calculated for the data subset from a single door subsystem. This was done to capture the door subsystems specific usage conditions and to make its behavior comparable to other door subsystems. In this way the values of all signals were normalized to be in the same value range.

There are several ways to normalize data. One of the simplest is to divide each value by the range of the values, which corresponds to the difference between the minimum value and the maximum value. This usually results in a value range between 0 and 1. In order to capture possible deviations from the minimum and the maximum value which are not covered by the selected dataset, some value range above the maximum value and below the minimum value are reserved to provide more flexibility for the algorithm. Therefore in this research, the value range was normalized to the range 0 and 0.833, which enables deviations from the observed maximum values in new datasets.

2.5. Balancing dataset Composition

High priority driver action diagnostic events are rare. Therefore, there are many data patterns in the class ”N“ and comparably few in the class ”H“. Thus, the datasets are not representative of the true underlying distribution of the class ”H“ and the dataset is highly unbalanced. Algorithms trained on unbalanced datasets usually have weak generalization ability. There are several approaches to handle unbalanced datasets (Duda et al., 2001).

In this research we created a balanced set of training data by selecting an equal number of data patterns from class ”H“ and class ”N“ for further analyses by omitting some of the data patterns from class ”N“. If the selected training data sufficiently represent the distribution of the entire dataset, this approach is valid and does not impair algorithm performance. Since data patterns have a high degree of similarity this approach is reasonable. Finally, after normalization and balancing the dataset, the sequence of data patterns in the dataset was randomized.

2.6. Principal Component Analysis

In large datasets variables are often correlated. This means that the information content is lower than in uncorrelated variables. Furthermore, high dimensional input data increases computational resources needed for data analysis. The principle component analysis (PCA) can be applied to reduce the dimensionality of a dataset with many correlated variables. By applying PCA the original data is transformed into a set of uncorrelated variables referred to as principle components (PC) without inducing any significant loss in variation compared to the original dataset (Jolliffe, 2004). The principal components are ordered according to the amount of the original datasets variation they explain. So the first PC will explain the largest amount of variation etc. Principal components can be considered as a lower dimensional representation of the input data. Since the data being analyzed in this research is sparse and it is assumed that the variables correlate, a principal component analysis was performed. The percentage of explained variance by the principal components was set to 95%. In this case a small loss of dataset variance was acceptable since this improved the algorithms classification performance and significantly reduced computing resources.

2.7. Network Setup and Training

An important part of the echo state network process is setting the network parameters. The ESN consisted of a reservoir with leaky integrator neurons. The applied activation function within the neurons was a standard sigmoid function. No explicit bias was included in the model. Ridge regression with a regularization term was applied to determine the optimal combination of reservoir signals that generate the desired output signal.

In this research the network parameters were determined by performing a hyperparameter optimization with grid search technique for each high priority signal (Bengio, 2000). Cross-validation was applied within the grid search process.

During training process the neural network is fed by the input signals and pertinent response signals are induced in the sparsely connected reservoir. Next, the response of the network and the teacher signals are applied to train the weights in the output layer using ridge regression. After the training process is completed, the weights and the reservoir states are fixed and the reservoir can be applied to new input data.

2.8. Evaluation of Results

After the hyperparameters have been determined, the generalization ability of the prediction algorithm using the holdout technique is validated. For this, the dataset is divided in two subsets. Both subsets are assumed to be representative for the underlying data distribution of the entire dataset and both subsets are independent (Haykin, 2009). The first subset is used to induce the signals in the ESN reservoir and to determine the output weights; this data is generally called the training dataset. The second subset is used to test how well the algorithm can generalize from data patterns that have not yet been presented to it; this data is generally called the testing dataset and is smaller than the training dataset (in this research 90% of data were used for training and 10% for testing).

The algorithms generalization ability stands in contrast to the pure memorization ability of the network. If the network simply memorizes the presented patterns from the training dataset, it will not be able to transfer the learned patterns to the testing dataset and to generalize the common features (Bishop, 2005). Overfitting occurs when the algorithm fits the training data perfectly but is not able to generalize and transfer the patterns to data that have not yet been presented to the algorithm. While there is usually a decrease in classification performance from training to testing data, a massive difference between the performance parameters for the testing and training datasets indicates a poor generalization ability and could be caused by overfitting.

In this research the analyses, evaluation procedures and all algorithms were computed in the Python programming language(Rossum, 2012). Several extension modules were also used including Numpy for array computation (Eric Jones and others., 2001), Matplotlib for graphical representations (Hunter, 2007), the Modular toolkit for Data Processing (MDP) for PCA analysis (Zito et al., 2009) and the ORGANIC Reservoir Computing Engine for echo state networks (Verstraeten, 2010).

3. RESEARCH RESULTS

The echo state network developed using the process outlined above was evaluated to assess its ability to generalize dynamic data patterns in the diagnostic event data and thereby predict potential railcar door incidents that could lead to railway system disruptions. This section summarizes how well the ESN worked.

3.1. Pre-Selection of the Considered High Priority Events

Two criteria were used to select the high priority events that were analyzed using the ESN model. First, because predicting an impending disruption event is only useful if the failures 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 analyzed. High priority events caused by different technical components or by external factors were not analyzed.

Second, for some of the distinct high-priority events, the datasets were too small to generate statistically significant prediction results. Even though the classification results were above the defined threshold and the random classification by the algorithm could be excluded, the prediction precision was not adequate.

Using these two criteria, three event types out of the 12 high priority events were selected for further analysis.

3.2. Non-Linearly Separable Problem

The first evaluation was to determine if the problem was non-linearly separable, in other words that no linear classifier could solve the task. Since the data is high dimensional it is not possible to illustrate it visually and therefore an initial principle component analysis was performed to reduce the dimensionality of the data for visualization purposes. In order to plot the data in two dimensions, the first two principal components were considered.

In this case, for one of the analyzed events the variance described by the first two principle components was 43%. The distribution of these two principal components for the datasets of both classes is displayed in Figure 3. The figure clearly indicates that the two classes, represented by their first two principle components, are not linearly separable and that the data patterns from both classes are highly overlapping within the dataset. This behavior is similar for all the analyzed events even though the principal components are not similar.

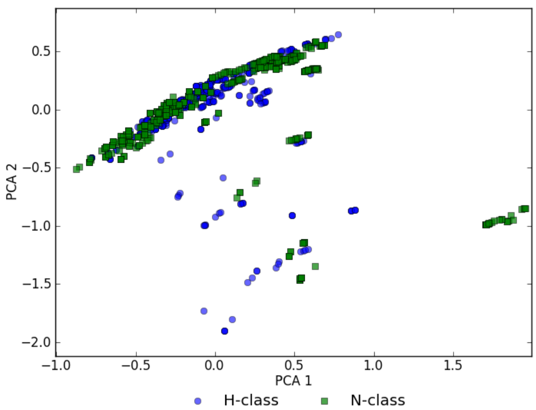

Figure 3: First Two Principal Components of a Selected dataset

3.3. Misclassification Rate

Misclassification rate is a metric commonly applied to assess the performance of classification algorithms. This is the ratio between the incorrectly classified patterns and all the patterns presented to the algorithm. It does not distinguish between the different classes and the performance of the algorithm within the selected class. A significant difference in the misclassification rate between the training and the testing dataset can indicate a poor generalization and overfitting. The algorithm developed in this research performs well with a misclassification rate between 2.2% and 3.1% for the testing datasets (Table 1). In the training datasets all the patterns were classified correctly (misclassification rate is therefore 0%). This shows that the algorithm has good generalization ability.

Another important evaluation criterion is ensuring that the classification task is not performed purely randomly. If the neural network has not learned the patterns within the data it would randomly assign the labels to the data patterns, in this case 50% of the labels would be assigned correctly for both classes. The

Table 1: Misclassification Rates of the Classification Algorithm for the Testing dataset

		High priority event category I	High priority event category II	High priority event category III
Misclassification rate	Testing	2.40%	2.20%	3.10%
Total number of patterns	Training	1,248	1,354	1,274
	Testing	125	135	127

threshold for considering the classification process as non-random is between 59.8% and 60.2% (for a 99% confidence level) for the test sample and between 53.2% and 53.3% for the training sample (depending on the size of the sample). The rate of correctly classified patterns is simply 100% minus the rate of incorrectly classified patterns. Table 1 shows that the rate of correctly classified patterns is significantly above the maximum threshold of 60.2%.

3.4. Confusion Matrices

The confusion matrix is a performance assessment parameter that can distinguish between the classes and compare the classification performance within a single class. For binary classification tasks (i.e. classification in only two classes), performance can be measured using the sensitivity and specificity parameters. Sensitivity measures the algorithm's ability to identify the positives while specificity measures the algorithms ability to identify the negatives. Positives are defined as the data samples from the class with the specified condition of interest. Negatives are data samples from the other class (i.e. which do not have the specified condition). In this research the positives are defined as data samples with a high priority diagnostic event expected within seven days.

The higher the sensitivity and specificity, the better the algorithms performance. There is usually a trade-off between the two measures and the weight given to them can be adjusted depending on whether the costs of false positives (e.g. replacing parts that may not fail) are higher than those of false negatives (e.g. events were not detected by the algorithm and caused severe service disruptions). In this case the costs for false positives and negatives are considered equally important and therefore sensitivity and specificity were weighted equally.

Table 2 presents the confusion matrix for the high priority events analyzed in this research for the testing data. The table displays relative rather than absolute values.

Table 2: Specificity Confusion Matrix and Performance Analysis for Testing dataset
Sensitivity is Marked with Dark Grey and Specificity is Marked Light Grey

		High priority event category I		High priority event category II		High priority event category III	
		H	N	H	N	H	N
Actual class	H	98.50%	1.50%	97.10%	2.90%	95.30%	4.70%
	N	3.40%	96.60%	1.50%	98.50%	1.60%	98.40%
Total number of testing patterns (class H/N)		125 (67/58)		135 (70/65)		127 (64/63)	

The data on Table 2 show that although there are some marginal differences between the sensitivity and the specificity the algorithm shows both a high sensitivity and a high specificity.

4. DISCUSSION AND CONCLUSIONS

This research shows that echo state networks can be used to predict the occurrence of high-priority events based on automatically generated diagnostic event data. More specifically, it shows that the dynamics of echo state networks are able to capture the dynamics in the time patterns of diagnostic event data and transfer these dynamics to testing data. The combination of algorithms developed in this research performed well achieving sensitivity values between 95% and 99% and specificity values between 97% and 98% for testing datasets.

For railways this means that well-designed echo state networks could predict the occurrence of future service disruptions based on an analysis of subsystem or specific component diagnostic data. The practical purpose of these predictions would be to reduce the number of operational disruption events and thereby increase service reliability. This would also enable railways to reduce maintenance costs by creating more efficient planned maintenance programs.

Despite good classification performance for the developed approach, there are some limitations. First, the model developed in research only includes diagnostic event data. The results could be improved by integrating additional data (e.g., data on actual technical failures and system disruptions). By including these additional data a relationship between high-priority events and actual disruption events could be derived and processes to anticipate the responsible technical failures could be established. In addition to these failure data, environmental data could also be integrated into the approach and the prediction quality could be improved.

Another limitation was that a moving window approach with partly overlapping time patterns was used to obtain sufficient data. The overlap was large in order to cover all possible combinations of time patterns. In a bigger dataset the overlap could have been reduced and the generalization ability could probably be increased compared to the present study.

Given the good results there are several ways to extend the research. First, the approach can be applied to other rolling stock systems and subsystems. The comparison of both results could give some insights on further performance improvement. The approach and algorithm could also be transferred to other systems with similar data or to other kinds of data, for example railway infrastructure systems. Furthermore, the time period considered in this research was comparably short due to the availability of data. In this short consideration period the evolving dynamic behavior of the data series might not have been fully captured. Therefore, the dataset should be enlarged to verify the developed algorithms generalization ability.

An important caveat regarding the proposed approach and developed algorithm is that it does not consider the time needed to react to the prediction. The approach can only predict that a high-priority event will occur within the next seven days but not when the event is going to happen within these seven days. For example, if the event occurs within the next 12 hours it might be impossible to prevent because insufficient time would be available for the rescheduling, planning and preparation processes. Therefore, for practical application it would be beneficial to include a fixed time to react to the prediction. This fixed reaction time would depend on the specific maintenance conditions (e.g., maintenance can only be undertaken on defined fixed weekdays, due to schedule adherence etc.).

One strategy might be to distinguish between three different time horizons by establishing categories ”soon”, ”mid-range” and ”long-range” for occurring events. Three fuzzy classes would contain more information. Prediction of ”mid-range” occurring events might be beneficial if special preparations for maintenance are required or adjustments to schedules of railway vehicles need longer lead-time.

Another extension of the research and its practical applicability would be to perform a sensitivity analysis for different input signals. This would be beneficial in identifying the signals with the biggest influence on the frequency of occurring events. Finally, practical applications of the algorithm require additional uncertainty analysis. This would further assess the performance of prediction results and would be a good subject for further research.

5. Acknowledgements

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

6. References

Bengio, Y., 2000. Gradient-based optimization of hyperparameters. Neural Computation 12 (8), 1889–1900.

Bishop, C. M., 2005. Neural networks for pattern recognition, reprinted 2005 Edition. Oxford University Press, Oxford.

Duda, R. O., Hart, P. E., Stork, D. G., 2001. Pattern classification, 2nd Edition. John Wiley, New York.

Eric Jones, Travis Oliphant, P. P., others., 2001. Scipy: Open source scientific tools for python.

Fararooy, S., Allan, J., 1995. On-line condition monitoring of railway equipment using neural networks. In: Advanced Condition Monitoring Systems for Railways, IEE Colloquium on. pp. 2/1–2/9.

Fink, O., Weidmann, U., 2011. Scope and potential of applying artificial neural networks in reliability prediction with a focus on railway rolling stock. In: Berenguer, G. . G. S. (Ed.), European Safety and Reliability Conference (ESREL) Annual conference 2011. Vol. Advances in Safety, Reliability and Risk Management. Taylor & Francis Group, London.

Haykin, S. S., 2009. Neural networks and learning machines, 3rd Edition. Pearson Education, Upper Saddle River.

Hunter, J. D., 2007. Matplotlib: A 2d graphics environment. Computing in Science and Engineering In Computing in Science & Engineering Vol. 9 (No. 3.), pp. 90–95.

Jaeger, H., 2005. Tutorial on training recurrent neural networks, covering bptt, rtrl, ekf and the "echo state network" approach. Tech. rep., German National Research Center for Information Technology.

Jaeger, H., Haas, H., 2004. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. Science 304 (5667), 78–80.

Jolliffe, J. T., 2004. Principal component analysis, 2nd Edition. Springer, New York.

Lehrasab, N., Dassanayake, H. P. B., Roberts, C., Fararooy, S., Goodman, C. J., 2002. Industrial fault diagnosis: Pneumatic train door case study. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit 216 (3), 175–183.

Lukosevicius, M., 2012. A Practical Guide to Applying Echo State Networks. Vol. 7700 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, Ch. 36, pp. 659–686.

Mandic, D. P., Chambers, J. A., 2001. Recurrent neural networks for prediction learning algorithms, architectures and stability. Wiley, Chichester.

Roberts, C., Silmon, J., 2012. Advanced techniques for monitoring the condition of mission-critical railway equipment. Information Science Reference, Hershey PA, pp. 341–354.

Rossum, G. v., 2012. Python setup and usage, release 2.7.3.

Smith, A. E., Coit, D. W., Yun-Chia, L., 2010. Neural network models to anticipate failures of airport ground transportation vehicle doors. Automation Science and Engineering, IEEE Transactions on 7 (1), 183–188.

Vachtsevanos, G., 2006. Intelligent fault diagnosis and prognosis for engineering systems. Wiley, Hoboken, NJ.

Verstraeten, D., 2009. Reservoir computing: computation with dynamical systems. Ph.D. thesis, Ghent University.

Verstraeten, D., 2010. Oger toolbox.

Zito, T., Wilbert, N., Wiskott, L., Berkes, P., 2009. Modular toolkit for data processing (mdp): a python data processing framework. Frontiers in Neuroinformatics 2.