Application of Event Based Decision Tree and Ensemble of Data Driven Methods for Maintenance Action Recommendation

James K. Kimotho, Christoph Sondermann-Woelke, Tobias Meyer, and Walter Sextro

Department of Mechatronics and Dynamics, University of Paderborn, 33098 Paderborn, Germany
james.kuria.kimotho@uni-paderborn.de
christoph.sondermann-woelke@uni-paderborn.de
tobias.meyer@uni-paderborn.de
walter.sextro@uni-paderborn.de

ABSTRACT

This study presents the methods employed by a team from the department of Mechatronics and Dynamics at the University of Paderborn, Germany for the 2013 PHM data challenge. The focus of the challenge was on maintenance action recommendation for an industrial equipment based on remote monitoring and diagnosis. Since an ensemble of data driven methods has been considered as the state of the art approach in diagnosis and prognosis, the first approach was to evaluate the performance of an ensemble of data driven methods using the parametric data as input and problems (recommended maintenance action) as the output. Due to close correlation of parametric data of different problems, this approach produced high misclassification rate. Event-based decision trees were then constructed to identify problems associated with particular events. To distinguish between problems associated with events that appeared in multiple problems, support vector machine (SVM) with parameters optimally tuned using particle swarm optimization (PSO) was employed. Parametric data was used as the input to the SVM algorithm and majority voting was employed to determine the final decision for cases with multiple events. A total of 165 SVM models were constructed. This approach improved the overall score from 21 to 48. The method was further enhanced by employing an ensemble of three data driven methods, that is, SVM, random forests (RF) and bagged trees (BT), to build the event based models. With this approach, a score of 51 was obtained . The results demonstrate that the proposed event based method can be effective in maintenance action recommendation based on events codes and parametric data acquired remotely from an industrial equipment.

1. INTRODUCTION

The focus of the 2013 Prognostic and Health Management data challenge was on maintenance action recommendation in industrial remote monitoring and diagnostics. The challenge was to identify faults with confirmed maintenance actions masked within a huge amount of data with unconfirmed problems, given event codes and associated snapshot of the operational data acquired when a trigger condition is met on board. One of the challenges in using such data is that most industrial machines are designed to operate at varying conditions and a change in the operation conditions triggers a change in sensory measurements which leads to false alarm. This poses a huge challenge in isolating abnormal behavior from false alarm since it tends to be masked by the huge number of the false alarm instances recorded. In addition, most industrial equipment consist of an integration of complex systems which require different maintenance actions, further complicating the diagnostic process. A maintenance action recommender that is able to distinguish between faults with confirmed maintenance action and false alarms is therefore necessary.

Remote monitoring and diagnosis is currently gaining momentum in condition based maintenance due to the advancement in information technology and telecommunication industry(Xue & Yan, 2007). In this approach, condition monitoring data is remotely acquired and whenever an anomaly is detected, the data is recorded and transmitted to a central monitoring and diagnosis center (Xue, Yan, Roddy, & Varma, 2006). Here further analysis on the data is conducted to isolate and diagnose the faults and based on the outcome of the diagnosis, a maintenance scheme is recommended. This has led to a reduction in maintenance costs since an engineer is not required on-site to perform the trouble shooting. A number of literature based on remote monitoring and diagnosis have been published. Xue et al. (2006) combined non-parametric statistical test and decision fusion using generalized regression neural network to predict the condition of a locomotive engine. The condition was classified as normal or abnormal. Xue and Yan (2007) presented a model-based anomaly detection strategy for locomotive subsystems based on parametric data. The method involved the use of residuals between measured and model outputs to define a health indicator based on normal condition. Statistical testing, Gaussian mixture model and support vector machines were then used to evaluate the health index of the test data. Similarly the study focused on ability to detect normal and abnormal behaviors from operational data. It is evident that the use of operational data obtained remotely poses a huge challenge in fault identification and classification and there is need to develop algorithms that are capable of exploiting the operational data to not only detect abnormalities in industrial equipment, but also to classify faults and recommend maintenance action. The 2012 PHM challenge was based on the need for recommenders with this capability.

The following sections describe the data used in the challenge and data preprocessing.

1.1. Data Description

In order to develop an effective maintenance recommender, it was important to understand the structure of the data. Due to proprietary reasons, there was very little information about the sensory measurements. The data which was obtained from an industrial equipment was presented in comma separated values (csv) and consisted of the following:

1. Train - Case to Problem: a list of cases with confirmed problems, where the problem represent a maintenance action to correct the identified anomaly. This list consisted of 164 cases with a total of 13 problems.
2. Train - Nuisance Cases - a list of cases whose symptoms do not represent a confirmed problem. These are cases created by automated systems and presented to an engineer who established that the symptom was not sufficient to notify the customer of the problem. These cases constituted the bulk of the training data.
3. Training cases to events and parameters - a list of all the cases in (1) and (2) with the events that triggered the recording of the cases and a snapshot of the operating conditions or parameters. A total of 30 parameters were acquired at every data logging session. A collection of events and corresponding operational data at the point when the anomaly detection module is triggered refers to the case. The event code indicates the system or subsystem that the measurements came from and the reason why the code was generated. Some cases contain multiple data instances while some contain single data instances.
4. Test cases to events and parameters - a list of cases with the corresponding events and parameters for evaluating the recommender. The recommender should propose a maintenance action for a confirmed problem and output 'none' for unconfirmed problems.

Table 1 shows a summary of the training and testing data.

Data set	Number of Cases	Number of instances	Number of unique events
Train - Case to problem	164	27,205	358
Train - Nuisance cases	10295	1,269,243	1239
Test cases	9358	1,893,882	1246

Table 1. Summary of training and testing data.

Due to the large sizes of the training and testing data files, each of the files was broken down into smaller files of 5000 instances and loaded into MATLAB environment from where the data was converted into '*.mat' files. All the processing was handled within the MATLAB environment.

1.2. Data Preprocessing

The first step in preparing the data was to separate events and parameters associated with the train case to problem from those associated with nuisance cases. Figure 1 shows the distribution of the problems within the train-case to problems data. As seen from Figure 1, problem P2584 is more prevalent with problem P7940 having the least recorded cases.

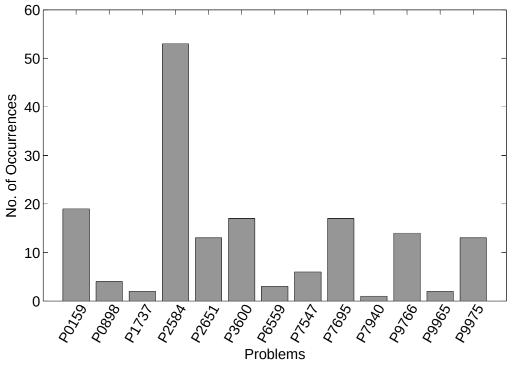

Figure 1. Distribution of problems in the train-case to problem training data

The data was unstructured in that it contained both numerical values and in some instances characters ('null'). The string 'null' was interpreted as zero (0). The data also contained some data instances with missing parameters. It was not possible to remove this data since some cases had all data instances with missing parameters. Therefore these cases were treated separately. There were also some parameters with constant values, for instance parameter P05. These parameters were removed from the data leaving a total of 26 parameters out of 30.

1.3. Data Evaluation Challenges

Since data was acquired whenever a specific condition is met onboard, rather than a fixed sampling rate, feature extraction and preprocessing of the data was quite difficult. In addition, due to proprietary reasons, the nature of the parameters recorded was not revealed. This made it difficult to define the thresholds for diagnostic purposes. Another challenge identified was the few samples or data instances in some cases, with some having only one event recorded. In such cases the data was masked by the nuisance data and made it very difficult to identify the actual problems. The small percentage of confirmed problems, 164 cases against 10295 nuisance cases would require batch training since in normal circumstances, the nuisance data would suppress the confirmed problems.

The following sections describe the methodologies employed by our team. Since ensemble of machine learning (ML) algorithms has been considered as the state of the art approach in both diagnosis and prognosis of machinery failures (Q. Yang, Liu, Zhang, & Wu, 2012), the first attempt was to use the parametric data together with an ensemble of the state of the art machine learning algorithms to classify the problems and also identify nuisance data. However, due to the close correlation between the parametric data of different problems, it was discovered that the use of parametric data together with ensemble of ML algorithms was not sufficient. The method was therefore extended to incorporate the event codes to improve classification performance.

2. PARAMETRIC BASED ENSEMBLE OF DATA DRIVEN METHODS

The first attempt was to employ an ensemble of data driven methods with the given parameters as the input and the confirmed problems as the target for training. Once the algorithms were trained, parameters from the test data were used as the input to the trained models whose output was the predicted problems. The following six data driven methods were employed:

1. k-Nearest Neighbors (kNN): In this method, the distance between each test datum and the training data is calculated and the test datum is assigned the same class as most of the k closest data (Bobrowski & Topczewska, 2004). Various distance functions can be employed but Mahalanobis distance function (Bobrowski & Topczewska, 2004) was found to perform best on the given data. The distance between data points x and y can be calculated by.

$$d(x, y) = \sqrt{(x-y)^T S^{-1}(x-y)}, \tag{1}$$

   where S is the covariance matrix of data x.

2. Artificial Neural Networks (ANN): ANN maps input data into output through one or more layers of neurons where each neuron is connected to the output of all the neurons of the preceding layer. Each neuron computes the weighted sum of its inputs through an activation function (Rajakarunakaran, Venkumar, Devaraj, & Rao, 2008). Training ANN consists of adapting the weights until the training error reaches a set minimum. A feedforward neural network consisting of three hidden layers with [110 110 80] neurons in the three layers respectively was employed. Scaled conjugate gradient (SCG) was employed as the training algorithm due to its ability to converge faster and accommodate large data.

3. Classification and Regression Trees (CART): CART predicts response to data through binary decision trees. A decision tree contains leaf nodes that represent the class name and decision nodes that specify a test to be carried out on a single attribute value, with one branch and sub-tree for each possible outcome of the test (Sutton, 2008). To predict a response, the decisions in the tree are followed from the root node to the leaf node. Pruning is normally carried out with the goal of identifying the tree with the lowest error rate based on previously unobserved data instances. Figure 2 shows a section of the classification tree with the decision nodes, where xNN represents the parameter number. The decision rules were derived from the parametric data.

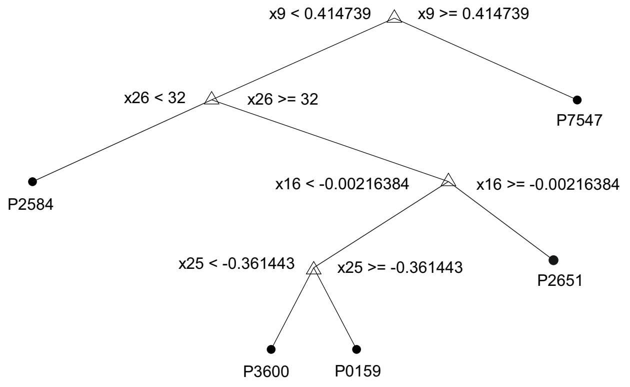

Figure 2. A section of a classification tree with decision nodes and leaves

4. Bagged trees (BT): Bagged trees involve training a number of classification trees where for each tree, a data set (S_i) is randomly sampled from the training data with replacement (Sutton, 2008). During testing, each classifier returns its class prediction and the class with the most votes is assigned to the data instance. In this study, 100 trees were found to yield the best results during training.

5. Random forests (RF): Random forests is derived from CART and it involves iteratively training a number of classification trees with each tree trained with a data set that is randomly selected with replacement from the orig-

inal data set (B.-S. Yang, Di, & Han, 2008). At each decision node, the algorithm determines the best split based on a set of features (variables) randomly selected from the original feature space. The final output of the algorithm is based on majority voting from all the trees. Figure 3 shows the construction of a random forest, where S is the original data set and S_i is the randomly sampled data set, C_i is a classification tree trained with data set S_i and N is the total number of trees. A combination of 500 trees with 50 iterations was found to yield the best results with the given data.

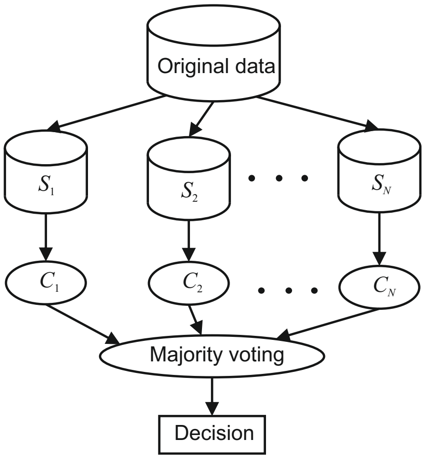

Figure 3. Construction of a random forest

6. Support vector machines (SVM): SVM is a maximum margin classifier for binary data. SVM seeks to find a hyperplane that separates data into two classes within the feature space, with the largest margin possible. For non-linear SVM, a kernel function may be employed to transform the input data into a higher dimensional feature space where classification is carried out (Hsu & Lin, 2002). Multi-classification is achieved by constructing and combining several binary classifiers. Pairwise method where $\frac{n(n-1)}{2}$ binary SVMs are constructed was employed since it is more suitable for practical applications (Hsu & Lin, 2002). A three-fold cross-validation technique was employed during training.

2.1. Training

In order to evaluate the performance of the selected algorithms, training data consisting of the 164 cases with confirmed problems and a random sample of 40 nuisance cases was used. This translated to approximately 40,000 data instances. The data was randomly permuted and split into two: 75% of the data was used for training and 25% for testing. The process was repeated with 39 other sampling instances of the nuisance data to build 40 models. The average classification accuracy of each model was computed. An ensemble of the six algorithms was then built based on weighted majority voting, where the results from algorithms with higher accuracy were given more weight. Table 2 shows the average training accuracy of the selected algorithms.

Algorithm	Average accuracy
kNN	89%
ANN	88%
CART	83%
BT	93%
RF	94%
SVM	90%
Ensemble	95%

Table 2. Classification accuracy based on training data

A look at the classification errors revealed that the majority of the errors occurred for the cases with single events. In particular, cases with event E35590 recorded the most errors. Event E35590 appears in majority of the cases, both in the cases with confirmed problems and nuisance cases.

2.2. Testing

To evaluate the performance of the algorithms on the test data, the data was supplied each case at a time to the 40 models described in the previous section and majority voting employed to select the most likely problem for each algorithm. An ensemble of the algorithms based on weighted voting was then built.

The performance of the method was evaluated using Equation 2.

$$Score = NO - NIO - NNO, \qquad (2)$$

where NO is the number of outputs, NIO is the number of incorrect outputs and NNO is the number of nuisance outputs. If the method provided an output for cases with unconfirmed output, this was considered as nuisance output. The number of outputs was based on a sample of 348 cases with an equal number of nuisance cases and cases with confirmed problems. A score of 21, with $NO = 303$, $NIO = 133$ and $NNO = 149$ was obtained based on this method. From the results obtained, it was clear that using parameters exclusively to predict the problem would not yield good results. The next attempt was to incorporate the event codes in the classification process.

3. EVENT BASED DECISION TREE AND SUPPORT VECTOR MACHINES

Since the event code indicates the system or subsystem that the measurements came from and the reason why the code was generated, a method combining event based decision tree and SVM was developed. The input to the SVM was the parametric data corresponding to events. This section describes construction of this method.

3.1. Cases with Single Events

It was observed that some cases with confirmed problems consisted of single events. A decision tree to identify these events and problems was developed. Some of the single events appeared in multiple problems. In such cases, the parametric data was used to derive the rules to differentiate between the different problems. One such event was E35590. Figure 4 shows a section of the decision tree constructed for this event. L is the number of unique events per case.

As seen in Figure 4, the decision tree was extended to include other single events appearing in the training data. Parametric data was used to derive decisions at the nodes for single events appearing in more than one case. Due to time constraints, the decision tree was not trained to identify nuisance data.

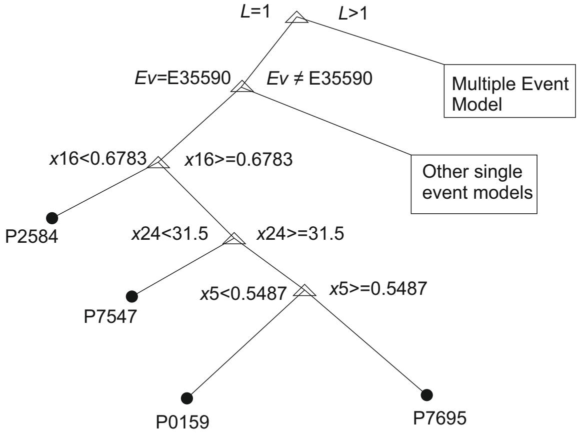

Figure 4. Construction of a decision tree for event E35590

For events appearing in multiple cases, SVM method with parameters optimally tuned using particle swarm optimization was used to train models corresponding to each event. A total of 165 models were trained to identify the possible problem given the event code and corresponding parameters. For events not appearing in the training data, the parametric data was tested with each of the models and the problem with the highest number of vote was selected.

3.2. Training

To train the method, the training data was again split into two, 75% of the data for training and 25% for testing. A 3-fold cross validation (CV) technique was employed during tuning of the parameters. PSO algorithm was employed to optimally tune the parameters of the SVM algorithm (Kimotho, Sondermann-Woelke, Meyer, & Sextro, 2013). Figure 5 shows the work flow of the SVM-based part of the method (Kimotho et al., 2013). 165 SVM models based on events that are triggered by multiple faults were trained. The prediction accuracy based on the training data was 90%.

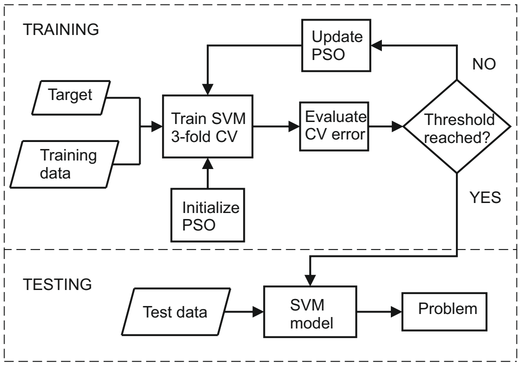

Figure 5. Work flow of the SVM based classifier with optimally tuned parameters

3.3. Testing

Testing was carried out by considering each case at a time. Based on the event code, the matching prediction model was retrieved and used to test parametric data of corresponding event. Since most cases had multiple events, the classification decision was arrived at based on majority voting.

From this method, a score of 48 was attained, with $NO = 332$, $NIO = 122$ and $NNO = 162$.

4. Event based Decision Tree and Ensemble of Data Driven Methods

In this method, event based decision tree and an ensemble of data driven methods that utilize the parametric data corresponding to the events were combined to classify the problems.

For events appearing in multiple cases, three data driven methods (RF, BT and SVM) were used to train models corresponding to each event. A total of 165 models for each method were trained to identify the possible problem given the event code and corresponding parameters. The classification decision was made by majority vote of the results from the three algorithms. For events not appearing in the training data, the parametric data was tested with each of the models and the problem with the highest number of vote was selected.

4.1. Training

To train the method, the training data was again split into two, 75% of the data for training and 25% for testing. The prediction accuracy based on the training data was 92%.

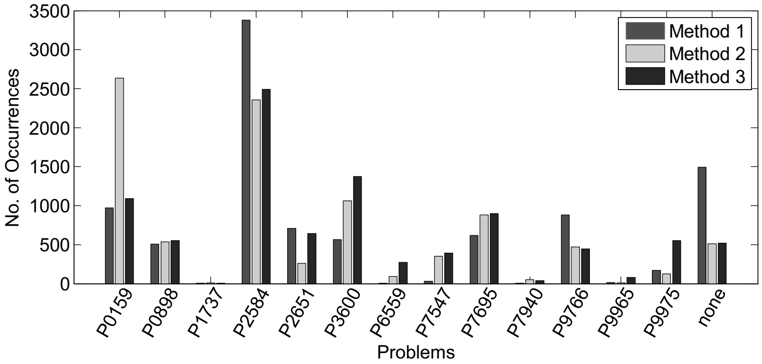

Figure 6. Distribution of classification results of the test data based on the three methods presented

4.2. Testing

Similar to the previous method, testing was carried out by considering each case at a time. Based on the event codes, the matching prediction model for each method was retrieved and used to classify the test data using parametric data as the input. The predictions from all the algorithms were combined and the classification decision made based on majority vote. With this approach, a score of 51 with $NO = 331$, $NIO = 121$ and $NNO = 159$ was obtained. This was a slight improvement compared to using event-based decision tree and SVM.

Figure 6 shows the distribution of the predictions from the three methods presented, where method 1 is the ensemble of data driven methods with only parametric data as input, method 2 is the event based method with SVM and method 3 is the event based method and ensemble of data driven methods.

5. CONCLUSION

Methodologies for recommending maintenance action, based on event codes and machinery parametric data obtained remotely have been presented. The large amount of data with unconfirmed problems (nuisance data) compared to the confirmed problems introduced a high rate of misclassification, especially when using the parametric data. However, incorporating event codes in classifying problems was found to yield better results. This led to our team being ranked position three in the 2013 PHM data challenge. The method could be further improved to reduce the number of incorrect and nuisance outputs.

ACKNOWLEDGMENT

The authors wish to acknowledge the support of the German Academic Exchange Service (DAAD) and the Ministry of Higher Education Science and Technology (MoHEST), Kenya. The authors would also like to thank Jun.-Prof. Artus Krohn-Grimberghe for the fruitful discussions on big data analysis.

REFERENCES

Bobrowski, L., & Topczewska, M. (2004). Improving the k-nn classification with the euclidean distance through linear data transformations. In *Proceedings of the 4th industrial conference on data mining, ICDM.*

Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, *13*, 415-425.

Kimotho, J., Sondermann-Woelke, C., Meyer, T., & Sextro, W. (2013). Machinery prognostic method based on multi-class support vector machines and hybrid differential evolution particle swarm optimization. *Chemical Engineering Transactions*, *33*, 619-624.

Rajakarunakaran, S., Venkumar, P., Devaraj, D., & Rao, K. S. P. (2008). Artificial neural network approach for fault detection inrotary system. *Applied Soft Computing*, *8*, 740-748.

Sutton, C. D. (2008). Classification and regression trees, bagging, and boosting. *Handbook of Statistics*, *24*, 303-329.

Xue, F., & Yan, W. (2007). Parametric model-based anomaly detection for locomotive subsystems. In *Proceedings of international joint conference on neural networks.*

Xue, F., Yan, W., Roddy, N., & Varma, A. (2006). Operational data based anomaly detection for locomotive diagnostics. In *Proceedings of international conference on machine learning.*

Yang, B.-S., Di, X., & Han, T. (2008). Random forests classifier for machine fault diagnosis. *Journal of Mechanical Science and Technology*, *22*, 1716-1725.

Yang, Q., Liu, C., Zhang, D., & Wu, D. (2012). A new ensemble fault diagnosis method based on k-means algorithm. *International Journal of Intelligent Engineering & Systems*, *5*, 9-16.