Modeling turbocharger failures using Markov process for predictive maintenance

Mahmoud Rahat

Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Sweden. E-mail: mahmoud.rahat@hh.se

Sepideh Pashami

Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Sweden. E-mail: sepideh.pashami@hh.se

Slawomir Nowaczyk

Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Sweden. E-mail: Slawomir.Nowaczyk@hh.se

Zahra Kharazian

Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Sweden. E-mail: zakhkha20@student.hh.se

The advancements of the telematics and connectivity solutions have provided new opportunities for the field of predictive maintenance. The number of sensors installed on a vehicle is increasing over time, and manufacturers are looking for new ways to improve the uptime of their fleet while at the same time reducing the costs related to unexpected breakdowns. The nature of the aggregated data from vehicles is sequential, and it is interesting to investigate existing methods for modeling partially observable state sequences to detect common patterns of failure. In this paper, we introduce a new approach for predicting turbocharger failures of Volvo trucks. The first step of the method deals with modeling a sequence of readouts from each vehicle using a Markov process. To do so, we identify the most informative signals and then employ spatial similarity clustering on the readouts. We interpret each cluster as a Markov state and further convert the history of a truck into a trajectory of states. This trajectory is then aligned with repairs information to form a standard sequence labeling problem. Finally, we train a hidden Markov model (HMM) classifier for assessing the health condition of the equipment. Empirical evaluations obtained on our real-world dataset of trucks suggest that the proposed method improves the AUC score of the final system up to 6% for predicting failures of a turbocharger.

Keywords: Predictive maintenance, Markov process, Hidden Markov Model, fault detection, automotive industry, turbocharger.

1. Introduction

Nowadays, the automotive industry increasingly feels the urge to shift from conventional maintenance strategies toward more effective policies, in particular predictive maintenance. Although traditional approaches might work for some simple components like a light bulb or brake pads, it is becoming evident that complex and sophisticated equipment such as turbocharger demands a smarter and more complex solution. Following the predictive maintenance concept, we monitor signals stored by a vehicle during its operation to measure the deterioration of the equipment to change it at the right time. According to (Carvalho et al. 2019), a proper maintenance strategy should demonstrate the following properties:

- Reduce the failure rates of the component
- Minimize maintenance costs
- Maximize life expectancy of the component

Swanson (Swanson 2001) categorizes maintenance strategies into two types, distinguishing between reactive maintenance and proactive maintenance. In reactive maintenance, we allow the equipment to run until it fails. On the other hand, in the proactive maintenance, we either conduct repairs periodically (known as preventive maintenance) or monitor the equipment deterioration (known as predictive maintenance) to fix or replace the part right before failure. Predictive maintenance of the vehicle components has many clear advantages over other maintenance policies (Carvalho et al. 2019). It helps to improve the reliability of the fleet and reduce the risks and costs related to unexpected

breakdowns and uptime promises under the maintenance agreements between original equipment manufacturers and customers.

The data used in this research covers three years of aggregated information from usage patterns of almost seven thousand trucks, combined their repair's information. The usage pattern is stored bi-weekly on a dataset called Logged Vehicle Data (LVD). Each row of this dataset (known as a readout) represents measurements drawn from vehicle operation and consists of 450 signals.

Additionally, the information regarding repairs of the trucks is stored in another dataset called Vehicle Service Records (VSR). Each row of this dataset shows a repair performed on a vehicle and contains information like chassis number of the truck, the exact date of the repair, and the name of the failed component.

In this paper, we narrowed our attention down to predicting turbocharger failures of Volvo trucks. A turbocharger is a vital and expensive part of the vehicle, and its failure usually leads to a complete stop of the car. Most of the time, it fails unexpectedly, and on rare occasions. This fact adds more complexity to the problem. A consequence of that is the class imbalance issue. Considering the entire LVD dataset, only a tiny portion of the readouts have a repair associated with them. Thus, to overcome this issue, we focused on a group of vehicles that have had at least one recorded repair in the data set. Even though by applying this constraint, the number of readouts associated with a service remains less than two percent of the total number of readouts.

The proposed method employs a stochastic Markov process to represent different aspects of truck history. In this representation, we were interested in identifying the trajectory of states, which could signify distinctive patterns of the typical behavior of these trucks. To this end, the similarities between readouts in the Logged Vehicle Data are investigated using an unsupervised clustering method. We then consider each cluster as a Markov state. Since the clustering is applied to the readouts from different vehicles, the state of a vehicle is likely to change from one cluster to another during its lifetime.

The sequence produced by each truck is later aligned with the repair's information of the vehicle, allowing us to investigate the patterns of failures using the hidden Markov model. We employed a prediction horizon technique to label

the readouts. The prediction horizon indicates a period of time, ahead of a failure, when a replacement recommendation should be made for that particular part. The following section gives more details about the proposed method.

2. Proposed method

The proposed method encompasses the next steps to model the history of a truck and predict its upcoming failures:

- (i) Preprocess the readouts and normalize features
- (ii) Identify the most informative signals for the task of failure prediction of turbochargers
- (iii) Divide vehicle's readouts into clusters and assume each group as a Markov state
- (iv) Associate the repairs to the model as the hidden states and create a trajectory for each vehicle
- (v) Estimate the optimal number of states for the learned representation
- (vi) Employ a hidden Markov model classifier on the trajectories to perform failure prediction

The following paragraphs study each step in more detail.

Step (i):

Step (i) of the proposed algorithm concerns the preprocessing of input signals. Transforming the categorical data into numeric values, dropping identity features, removing duplicate attributes, imputing missing values, and normalization are some of the transformations applied to the readouts.

Step (ii):

As mentioned earlier, each readout consists of a large number of attributes, many of which are noise or unrelated to our problem. This affects the quality of the processes applied to the subsequent steps. A conventional approach to tackle this issue is to identify and remove unimportant or less related features to the task at hand, especially in the case of having a high number of attributes (Dash et al. 2002). Moreover, this step is essential since the clustering algorithm considers the same weight for all the features. Therefore, in this step, we identify the most informative signals for the task of failure prediction of turbochargers.

There are different approaches available for estimating feature importance. Since our problem is a two-class classification task (i.e., healthy or unhealthy), we decided to apply a random forest classifier (which is a supervised method) to assign relative importance scores to the features and filter them based on a threshold. As stated in (Genuer, Poggi, and Tuleau-Malot 2010, Strobl et al. 2008), the random forest is growing in popularity in many scientific fields for solving the issue of feature selection.

Random forest is an ensemble learning method that applies bagging on the data. In this technique, we randomly pick samples with replacement and train a decision tree on the selected examples for a specific number of times. Each decision tree predicts a label for the input sample, and the majority vote is taken as the final label.

Step (iii):

The next step in the proposed algorithm is to design a set of Markovian states and assign each readout to one such state. To do that, we employ a spatial similarity clustering on the readouts using the k-means algorithm.

Step (iv):

We employ the prediction horizon technique introduced in (Prytz et al. 2015, Mashhadi, Nowaczyk, and Pashami 2020, Chen, Pashami, et al. 2019) to label the readouts. The prediction horizon indicates a period (time-window) ahead of a failure that a replacement recommendation should be made for the part. Based on the prediction horizon parameter, each readout is labeled as zero if that part has had no repair in the upcoming period. Otherwise, it is set to one to imply that a replacement for that part is recommended. In other words, as stated by (Mashhadi, Nowaczyk, and Pashami 2020), each period is assigned a binary label according to the following equation:

$$\text{Label}_t = \begin{cases} 1 & \text{if failure happens in } [t, t + \Delta] \\ 0 & \text{if no failure happens in } [t, t + \Delta] \end{cases} \quad (1)$$

Where t refers to a time step and Δ represents the prediction horizon. We set the prediction horizon value to 12 weeks, which means we are interested in predicting failures 12 weeks before the incident. This selection comes from expert knowledge and a business perspective.

Step (v):

Next, we tune the number of states in the model to achieve the highest accuracy for the proposed system on the validation set.

Step (vi):

The Markov model enabled us to assign a state to every readout and, therefore, to convert the history of a truck into a sequence of states. It is interesting to study if we can infer the health status of equipment by looking at the produced sequences of states. This is possible by training an HMM classifier on the labeled trajectories of the trucks. According to the definition (Rabiner and Juang 1986), if we consider X_n the actual health status of the equipment in the time step n , and Y_n as the discrete-time observable sequence of states generated from input signals, where $n \geq 1$, the pair (X_n, Y_n) is a hidden Markov model if:

$$P(Y_n \in A \mid X_1 = x_1, \dots, X_n = x_n) = P(Y_n \in A \mid X_n = x_n) \quad (2)$$

In Eq. 2, $P(Y_n \in A \mid X_n = x_n)$ is an emission probability term which is estimated over the training sequences. The naïve assumption of the hidden Markov model is that future states depend only on the current state. The next section reviews experiments conducted to evaluate the performance of the proposed method.

3. Experiments

In this section, we perform three tests to find the optimal parameter values and assess the performance of the proposed method. In the first experiment, we identify the most informative features in the readouts set. Then we apply an unsupervised method for clustering readouts to different clusters and generating the states. In the second experiment, we use the Markov process to model each truck history and associate the failures to the model. Finally, the third approach applies the Hidden Markov Model (HMM) classifier for predicting vehicle failures based on the produced model and optimizes the number of Markovian states.

3.1 Clustering vehicle's readouts dynamically based on spatial similarities

The main focus of this experiment is to select the most informative features using a random forest classifier and cluster the readouts into a set of representative states. To do so, we first need to estimate feature importance on readouts. We split the data into train, development (validation), and test sets. A random forest classifier is employed on the training set to calculate feature importance.

Table 1. Description of the five most informative signals and their relative importance scores selected by random forest classifier.

Feature description	Relative importance
Air dryer cartridges status	12.212
Engine hours	9.008
Movement hours in a specific gear	8.485
A bin in Engine speed/torque histogram	8.093
Age of the vehicle based on the years	7.223

Table 1. represents the five most important features along with their relative importance

scores. We scaled up the scores are for better readability.

Using the scores calculated by random forest classifier, we filtered the number of attributes to fifty most informative features. This number is empirically selected since we witnessed a drop on the importance scores close to this number. The idea behind applying feature selection is to reduce the dimension of the data by identifying and removing unrelated features in order to improve the clustering performance and decrease the noise. Subsequently, we applied the k-means clustering technique on the selected attributes of readouts. Fig. 1 illustrates the distribution of the clusters, along with the five most important features.

Fig. 1. A plot displaying the results of applying clustering on LVD data. Note that only the five most important features are displayed in the figure.

A critical parameter of the k-means algorithm is the number of clusters. In the current experiment, this parameter is set to a predefined fixed value. The effect of changing this parameter on the model efficiency is further investigated in section 3.3. The next experiment evaluates the impact of the modeling process on the performance of a classifier on the task of failure prediction.

3.2 Modeling turbocharger failures using Markov process

To assess the performance of the clustering results, we need to inspect how it affects the overall performance of a classifier on the task of failure prediction. To this end, in this experiment, we model vehicle behavior by considering each cluster as a Markov process, i.e., members of each cluster represent a state. The clustering is applied to the training set. Then we query closest cluster centroid to each readout in the validation set using a k-d tree algorithm to map it to a respective Markov state. The k-d tree algorithm has been applied to many different domains for fast neighborhood search (Chen, Zhou, et al. 2019, Zellmann, Schulze, and Lang 2019). Finally, we

exploit random forest classifier to compare four possible settings regarding whether we apply feature selection and Markov modeling on the

data or not. Empirical results prove both steps to be beneficial for improving the overall performance on the failure prediction task.

Fig. 2. Results on four possible scenarios concerning with application of feature selection and clustering.

Fig. 2 displays the results produced by random forest classifier on four parameter settings with four different prediction horizon values. In two settings (A and B), we apply feature selection, while for the rest, we consider all available features. On the other hand, the Markov model has exploited only on A and C. As you can see, setting A (that encompass both feature selection and Markov modeling) outperforms other settings. The feature selection process proves to be efficient since the application of clustering on the original features (setting C) improved the results compared to other approaches. Each plot in Fig. 2 represents results for a specific prediction horizon with different values for the number of clusters. It could be seen that conclusions drawn from the experiments are valid for most of the values of the prediction horizon and the number of clusters.

3.3 Predicting future failures using HMM and Optimizing the final model

As discussed in Section 3.2, building a Markov model based on the clustering of LVD features enhances the performance of failure prediction in this context. Though, we did not elaborate on an essential parameter of the model, i.e., the number of states (clusters). Perceptibly, using a predefined value for this parameter is not

efficient, and we need to optimize it according to the model performance on the validation set.

In this section, we experiment to estimate an optimal value for the number of states. Then, we describe the history of each truck through a sequence of states extracted from the optimized Markov model. Afterward, we train a supervised hidden Markov model (HMM) classifier to model the failures of the turbocharger. Subsequently, the prediction result of this model is compared with a random forest classifier applied to the original LVD features.

Since we are modeling several different trucks with the same hidden Markov model, the implementation should be able to compute the emission and transition probabilities from not only one sequence of observations but also many parallel sequences. It is also essential to apply the Viterbi algorithm (Forney 1973) to the trajectory of each truck separately to estimate the output labels. Since existing implementations did not fulfill the requirements of our problem (i.e., accepting multiple parallel input sequences), we implemented a customized library.

Fig. 3 shows the effect of changing the number of Markov states on the performance of the model for predicting the failures. The vertical axis

displays the AUC scores gained by the HMM classifier, while the horizontal axis represents the number of states (clusters). The number of clusters changes from 1 to 2000. For each value, we run the entire system (i.e., model generation using clustering and classification using HMM) three times to compute the mean and standard deviation of the area under the ROC curve (AUC).

Fig. 3. The plot shows how the AUC score of HMM classifier changes by altering the number of Markov states

The blue points in the plot display the mean values, while the standard deviations are illustrated using a vertical green line crossing each point. This experiment is conducted using the train and development set, and the prediction horizon is set to 12 weeks.

As you can see in Figure 3, the performance of the system drops as we increase the number of clusters to more than 500. The best performance achieved using 110 clusters. Considering this value as the optimized number of states, we conducted another experiment, this time on the test dataset, and compared the HMM classifier with two random forest classifiers. Note that HMM is applied on the trajectories produced from sequences of readouts, whereas random forest is used on a single readout. Table 2 presents the final results.

To make the comparison fair, we use the same dataset split for all three approaches. In Table 2, the feature selection column indicates whether the number of attributes is reduced to the set of selected features, or the entire feature set is used. The AUC score column, depicted in Table 2, shows the final performance of each approach, which articulates that the HMM classifier outperforms other methods.

Although we were able to improve the performance of the final system by processing the sequence of trajectories instead of single readouts,

the scores are still low. Many reasons are contributing to this fact; for instance, the difficulty of the problem and having a highly imbalanced data set. The weakness of the Hidden Markov Model to memorize the intricate patterns of failures could also be contributing to the problem.

The weakness comes from the fact that HMM applies Markov assumption, which considers only the previous state and ignores the rest of the sequence of events that precede it to compute the conditional probability of the current state. Since we produced the trajectories based on distance-similarity clustering, in many cases, the state of a truck stays in the same position for a couple of time steps before changing. Combining this fact with the Markov assumption, the system is unable to make deep inference over the trajectories.

Another important factor is the labeling process. As mentioned earlier, the LVD readouts are not annotated originally, so that we assigned zero and one labels based on the prediction horizon technique. Hence, the final AUC score of the system is highly sensitive to how we allocate the labels.

Table 2. results of applying random forest and HMM classifiers

Classification method	Feature selection	Markov model	AUC
Random Forest	No	No	0.5233
Random Forest	Yes	No	0.5660
Hidden Markov model	Yes	Yes	0.5847

4. Conclusion

This paper presented a new approach based on a stochastic Markov chain for modeling sequences of readouts to detect common patterns of failure in heavy-duty trucks. We first identified related signals to the problem and then clustered the readouts based on geometric proximity criteria using the k-means algorithm. The prediction horizon technique is applied to label each state based on the repair's information. Finally, an HMM classifier decides whether the equipment needs to be changed or not. The experimental results indicate that the modeling process, as well as the sequence labeler, were able to improve the performance of the system compared to the baseline. It shows that considering the time dependency between the samples capture new

information that is not considered if we treat every sample independently. Investigating the application of the proposed approach on other components like an air compressor, air below, and clutch remains as future work. Moreover, it is interesting to explore the effect of changing the labeling technique from the prediction horizon to other approaches like the remaining useful life (Mosallam 2014, Wang et al. 2020).

References

- Carvalho, Thyago P, Fabrizzio AAMN Soares, Roberto Vita, Roberto da P Francisco, João P Basto, and Symone GS Alcalá. 2019. "A systematic literature review of machine learning methods applied to predictive maintenance." *Computers & Industrial Engineering* 137:106024.
- Chen, Kunru, Sepideh Pashami, Yuantao Fan, and Sławomir Nowaczyk. 2019. "Predicting air compressor failures using long short term memory networks." *EPIA Conference on Artificial Intelligence*.
- Chen, Yewang, Lida Zhou, Yi Tang, Jai Puneet Singh, Nizar Bouguila, Cheng Wang, Huazhen Wang, and Jixiang Du. 2019. "Fast neighbor search by using revised kd tree." *Information Sciences* 472:145-162.
- Dash, Manoranjan, Kiseok Choi, Peter Scheuermann, and Huan Liu. 2002. "Feature selection for clustering-a filter solution." 2002 IEEE International Conference on Data Mining, 2002. Proceedings.
- Forney, G David. 1973. "The viterbi algorithm." *Proceedings of the IEEE* 61 (3):268-278.
- Genuer, Robin, Jean-Michel Poggi, and Christine Tuleau-Malot. 2010. "Variable selection using random forests." *Pattern recognition letters* 31 (14):2225-2236.
- Mashhadi, Peyman Sheikholharam, Sławomir Nowaczyk, and Sepideh Pashami. 2020. "Stacked Ensemble of Recurrent Neural Networks for Predicting Turbocharger Remaining Useful Life." *Applied Sciences* 10 (1):69.
- Mosallam, Ahmed. 2014. "Remaining useful life estimation of critical components based on Bayesian Approaches."
- Prytz, Rune, Sławomir Nowaczyk, Thorsteinn Rögnvaldsson, and Stefan Byttner. 2015. "Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data." *Engineering applications of artificial intelligence* 41:139-150.
- Rabiner, Lawrence, and B Juang. 1986. "An introduction to hidden Markov models." *IEEE assp magazine* 3 (1):4-16.
- Strobl, Carolin, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. 2008. "Conditional variable importance for random forests." *BMC bioinformatics* 9 (1):307.
- Swanson, Laura. 2001. "Linking maintenance strategies to performance." *International journal of production economics* 70 (3):237-244.
- Wang, Biao, Yaguo Lei, Tao Yan, Naipeng Li, and Liang Guo. 2020. "Recurrent convolutional neural network: A new framework for remaining useful life prediction of machinery." *Neurocomputing* 379:117-129.
- Zellmann, Stefan, Jurgen P Schulze, and Ulrich Lang. 2019. "Binned kd tree construction for sparse volume data on multi-core and GPU systems." *IEEE transactions on visualization and computer graphics*.