OPEN ACCESS

2nd International Electronic Conference on Sensors and Applications
15–30 November 2015

www.mdpi.com/journal/sensors

Conference Proceedings Paper–Sensors and Applications

Real-Time Self Adaptable Prediction System for Mobile Mine Equipment

Gurpreet Mohaar[1,*], Ramanpreet Singh [2] and Muhtasim Maleque[3]

[1] Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. Email: mohaar@ualberta.ca

[2] Computer Science, University of New Brunswick, Fredericton, Canada. Email: b4s79@unb.ca

[3] Mechanical Engineering, McGill University, Montreal, Canada. Email: muhtasim.maleque@mail.mcgill.ca

* Author to whom correspondence should be addressed; E-Mail: mohaar@ualberta.ca.

Published: 11 November 2015

Abstract: Identifying failure signatures of machines and modeling them to predict problems well before failure occur has been of great interest to reliability and maintenance engineers, primarily because of the unparalleled advantages like improved equipment up-time, lower maintenance cost, and reduced safety risk. Production critical machinery often requires intelligent real time monitoring and an unplanned interruption can have high cost implications. To address this, we utilize the on-board sensor data and develop a near-real time prediction system to identify anomalies and failure patterns of assets. Development of such data driven system will help improve reliability engineering strategies by modeling system dynamics and predicting equipment health problems.

Keywords: Smart Maintenance; Markov process; SVD; Intelligent sensor analytics, Exhaust manifold leak.

1. Introduction

Field of Sensor analytics has seen an immense growth in past few years. There are many contributing factors to that like changes in time series database technology, inexpensive storage and processing power and most importantly rising interest of machine learning community towards interdisciplinary research. The primary goal of Intelligent sensor analytics is to detect upcoming failures and anomalies. There are many use cases such as predicting and proactively preventing equipment failure in a manufacturing plant, alerting a nurse in an electronic intensive care unit when a patient's blood pressure drops, or allowing a data center administrator to make data-driven decisions about heating, ventilating and air conditioning (HVAC).

Although there has been considerable amount of research done in the fixed plant environment, researchers have just started to scratch the surface in the area Mobile Mining equipment like Haul Trucks, shovels and auxiliary equipment. The reason being the odd problems associated with mobile equipment operation like discontinues data stream because of dead zones in the mine, sophisticated machinery where different vendors are at play making data integration an issue, asynchronous data etc. Hence a robust, scalable and self-adaptive technique is required that can work with live data feed to convert the multidimensional machine sensor data into knowledge in order to forecast the future dynamics of the system as whole and provide real time predictions.

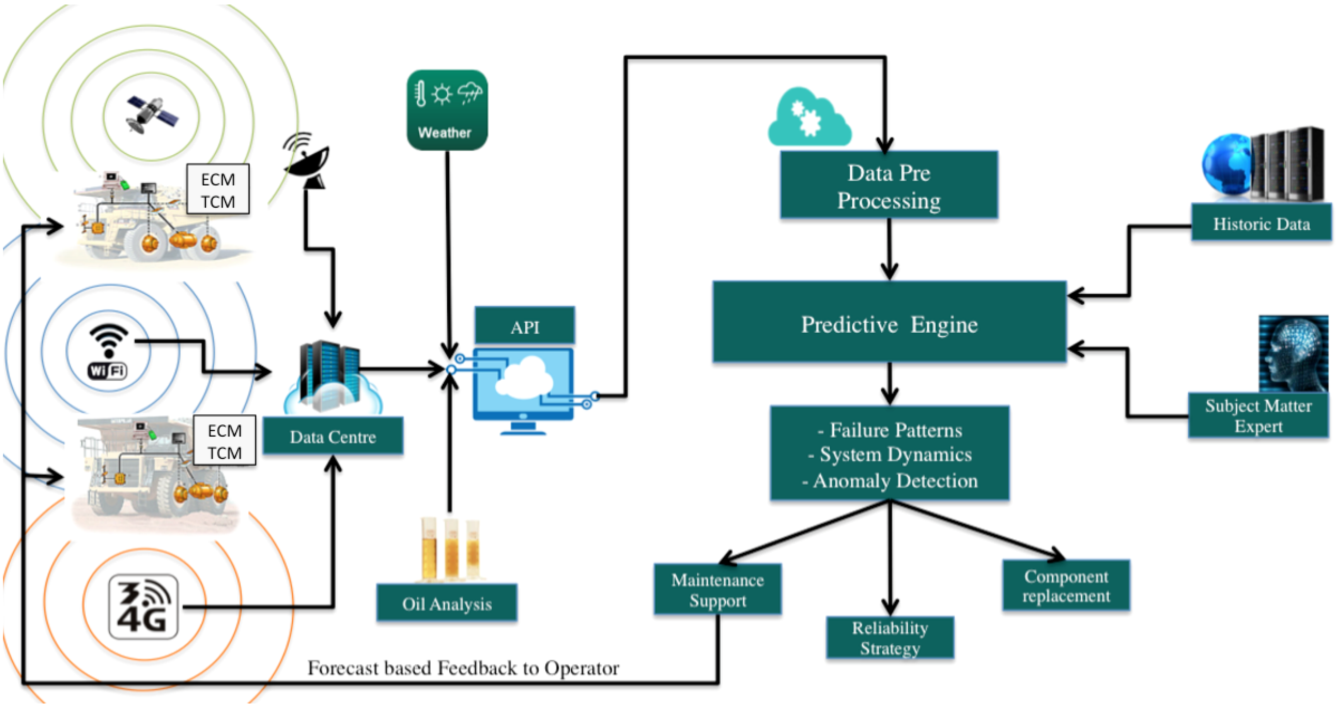

Figure 1. Proposed Architecture

Figure 1, explains the overall flow of the proposed framework from origin of data to forecast based feedback to the operations. In the world of mobile equipment, data collection is usually accomplished by connecting multiple ECMs (electronic control module) to a centralized system which processes and transmits the information to some kind of historian on a data server. In real life scenario a machine is manufactured by assembling different OEM components which makes it hard to integrate data from different ECMs primarily due to difference in time stamps. Apart from online data, static data sources like oil analysis, weather, maintenance feedback data, GPS etc are also integrated. The modeled data is fed into the prediction engine to get future dynamics, anomalies and failure patterns. This could be used to create variety of decision support tools to assist and develop maintenance and reliability technical operating envelopes and control measure around operations. Another advantage is in planning component replacements which is currently being done based on probability based failure modeling techniques like the weibull analysis [7]. This framework can be further developed to take health forecast based component replacement decisions.

2. Problem formulation and Discussion

Consider a machine with m sensors sending data asynchronously. We aim to predict short and long term behavior of the system. For this behavior analysis we regard observed data as realization of a set of underlying stochastic process ultimately responsible for what we observe. The proposed framework is divided into two phases - learning phase and self-adaptation and online prediction phase. In the learning phase, we first preform data pre-processing which involves filtering out sensor errors and creating time series data vectors by interpolating or extrapolating data points. After the pre-processing stage, data is segmented into small fixed size windows and characteristic features (f) are extracted using auto correlation coefficients. The resulting mxf matrix is decomposed using singular value decomposition (SVD) and only first K singular values are stored. Each time window is treated as a state in the Markov process and similar states are merged together using Frobenius norm comparisons. An irreducible state transition matrix and a matrix containing singular values of each state are returned.

In the online phase, real time data is passed through the prediction engine where the new time window (W) is analyzed and is either merged with a similar state (C) or added to the system as a new state and the state transition matrix is updated accordingly. If p is an irreducible regular Markov matrix, we calculate short and long range probabilities with C being the current state. A detailed framework is explained in section (3).

3. Framework

The proposed framework is divided into Learning Phase and Online Prediction Phase, which are described below.

3.1. ***Learning Phase***

The collected historical data is fed sequentially into learning module which is further broken down into 4 sub modules: Data Pre-processing, Feature extraction, Singular value decomposition and Markov chain process.

3.1.1. Data Pre-processing

Data preprocessing plays an important part in any machine learning algorithm especially in the field of sensor data analytics because of sensor related errors like missing or false values. A big challenge for data quality in mining equipment is asynchronous time stamps which come from the underlying logistics. Sophisticated mining machines such as a Komatsu 930E are assembled with various vendors involved who have their own centralized system of gathering data. The problem becomes more visible when data coming from two different systems needs to be integrated to find cross correlations at different lags as the time stamps are different and data is usually sampled and aggregated. An effective data pre-processing technique is required to deal with these problems and convert data into meaningful time series. We eliminate sensor errors by filtering out values exceeding possible max and min. We then convert data into fixed interval data vectors by linear or nonlinear interpolation based on nature of data.

3.1.2. Feature extraction

Data pre-processing is followed by another interesting step - feature extraction. Data is segmented into small fixed size windows and characteristic features are extracted using auto correlation coefficients. To compare sensor values based on their shape information, each sequence is normalized to have a zero mean and a standard deviation equal to one. Then, each normalized sequence is represented using a set of auto correlation coefficients. Considering x_k as a sub sequence with length q, its auto correlation coefficient for lag s can be estimated using (1).

$$Y_{k,s} = \frac{\sum_{t=s+1}^{q}(x_{k,t} - \hat{x}_k)(x_{k,t-s} - \hat{x}_k)}{\sum_{t=1}^{q}(x_{k,t} - \hat{x}_k)^2} \tag{1}$$

Auto correlation coefficients estimate how much a signal matches its time-shifted version [2].

3.1.3. Singular Value Decomposition

Singular value decomposition (SVD) has been effectively used for data compression in [1,8]. The task of compression begins with a matrix A consisting of numeric data, and the goal is to find a close approximation to A consisting of fewer dimensions. Because the rank of a matrix specifies the number of linearly independent columns (or rows), it is a measure of redundancy [3]. The best $rank - k$ approximation to a matrix $\mathbf{A} \in Rm \times n$ is

$$A_k = \sum_{i=1}^{k} \sigma_i u_i (v_i)^T \tag{2}$$

where $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_k \geq 0$ are the top k singular values of $\mathbf{A}$, with associated left and right singular vectors $u_i \in \mathbf{R}^m$ and $v_i \in \mathbf{R}^n$, respectively respectively. In our approach, we store and operate on the top *k* Singular values σ_i and corresponding $\mu_i$and $(v_i)^T$ vectors.

3.1.4. Markov Chain

A Markov process is a random process for which the future (the next step) depends only on the present state; it has no memory of how the present state was reached [4]. A Markov chain, studied at the discrete time points 0, 1, 2,..., is characterized by a set of states S and the transition probabilities $p_{i,j}$ between the states. Here, $p_{i,j}$ is the probability that the Markov chain is at the next time point in state j, given that it is at the present time point at state i. The matrix P with elements $p_{i,j}$ is called the transition probability matrix of the Markov chain.

Mathematically, we say that $(X_n)_{n \geq 0}$ is a Markov chain with initial distribution λ and transition matrix P if for all $n \geq 0$ and $i_0, ..., i_{n+1} \in I$,

$$P(X_0 = i_0) = \lambda_{i_0} \tag{3}$$

$$P(X_{n+1} = i_{n+1} \mid X_0 = i_0, ..., X_n = i_n) = P(X_{n+1} = i_{n+1} \mid X_n = i_n) = p_{i_n i_{n+1}} \tag{4}$$

In short, we say $(X_n)_{n>0}$ is $Markov(\lambda, P)$. Checking conditions (3) and (4) is usually the most helpful way to determine whether or not a given random process $(X_n)_{n>0}$ is a Markov Chain.

One of the goals of Markov chains is to find stationary distributions. Both irreducibility and aperiodicity are properties such that if they are fulfilled by a finite state Markov chain there exists a stationary distribution [6].

Irreducibility is the property that regardless the present state we can reach any other state in finite time [5]. Mathematically, it is expressed as:

$$\forall i, j \in S, \exists m < \infty : P(X_{n+m} = j \mid X_n = i > 0) \tag{5}$$

Each state in markov process is represented by top k singular values obtained in section 3.1.3 and similar states are merged together using euclidean distance by specifying a distance cutoff α. State transition matrix P is updated accordingly.When a new state is added to system, in order to satisfy the irreducibility condition, the Markov chain needs to be converted to irreducible chain by adding a path from the new state to all the states in the system with a very small probability.

3.2. ***Online prediction system***

In the online phase, real time data flows into a stack and depending on the size of window chosen, the stack is considered full if any of the sensors reach the count threshold. Data in stack is transferred to a pre-processing module where it is modeled into a time series. For this fixed size time window, top k singular values are calculated and the vector is then fed into the prediction engine where the new window (W) is analyzed and is either merged with a similar state (C) or added to the system as a new state and the state transition matrix is updated accordingly. If P is an irreducible regular Markov matrix, we calculate short and long range probabilities with C being the current state.

3.2.1. Short term behavior

The Probability distribution over states can be written as a stochastic row vector x with the relation:

$$x^{(n+1)} = x^{(n)}P \tag{6}$$

Where P is the state transition matrix obtained from the learning phase and $x^{(n)}$ is the current state (live). So if at time t the system is in state n, then in next time period, the system will be at state $n + 1$ with the probability $x^{(n)}P$.

Short term behavior could be used to tackle critical or high priority issues. For example, if the maximum probability of next transition is to a stage indicating fire hazard, operator on the mining equipment can be notified immediately potentially saving a life.

3.2.2. Long term behavior

In equation 6, when $n \to \infty$, we obtain steady state probability distribution of the system. This shows that the state with maximum probability is where the system will end up in long term. This can help us answer the question: Given the current health condition of equipment, what kind of problem(state) is it most likely going to end up in long term.

3.3. Case Study and Experimental Evaluation

The initial experimentation was done to predict problems related to exhaust leaks and worn turbo journal in a 797 haul truck engine. In this case study 2 years worth of synthetic sensor data was created from 6 different sensors mimicking the On-board VIMS data gathering system. The Parameters selected affecting the problem are: Exhaust temperature left front, Exhaust temperature right front, Exhaust temperature left rear, Exhaust temperature right rear, Boost pressure front and Boost pressure rear. In normal operation, front and rear engine operates at similar temperatures and pressure. Usually when there is deviation, it is commonly caused by exhaust leaks or worn turbo journals on the hot engine. We simulate time series for the chosen parameters where the exhaust temperature and boost pressure starts deviating slowly and eventually the front roughly trends 50°C hotter than the rear. The algorithm for this case study was tested for different values of:

- Fixed Window size
- Distance cutoff

After training the Markov chain, the model was tested using selected test windows. The model could categorize each window of data into different states based on the dynamics of each sensor. The test window is analyzed and is either merged with a similar state (C) or added to the system as a new state and the state transition matrix is updated accordingly. The state most similar to the test window is treated as the current state. Based on the exhaust temperature and boost pressure deviation, 10 windows were identified and windows just prior to those dates were selected as test windows. The predicted process dynamics of the system are plotted in Figure 2.

Plot 2(d) and 2(e) shows predicted and actual exhaust temperatures on front and rear respectively. Plot 2(f) shows the predicted values for front and rear, which shows a deviation indicating potential exhaust leaks or worn turbo journals. Furthermore, Plot 2(c) shows predicted values for boost pressure from front and rear showing deviations which further confirms the problem.

Our algorithm was able to proactively predict the potential problem. In a reactive environment, this problem would have caused the affected engine to run hot which may have damaged the exhausted manifolds and turbos, leading to unplanned downtime and costly repairs.Based on the forecast, maintenance is recommended to be notified to check the engine exhaust manifold for leaks and if found, manifold should be resealed with high priority.

4. Conclusions

In this paper, we proposed and analyzed a real time prediction system for mining equipment solving a variety of problems in the world of data driven reliability engineering. We utilized the concept of low rank matrix approximation for efficient storage and retrieval of real time data. Markov chain was used to model the process dynamics and to predict the short and long term behavior of the system. This self adaptable model is used for a case study to predict exhaust leaks on a CAT 797 Haul truck engine. The developed model can be combined with other data sources to create an effective decision support tool for maintenance.

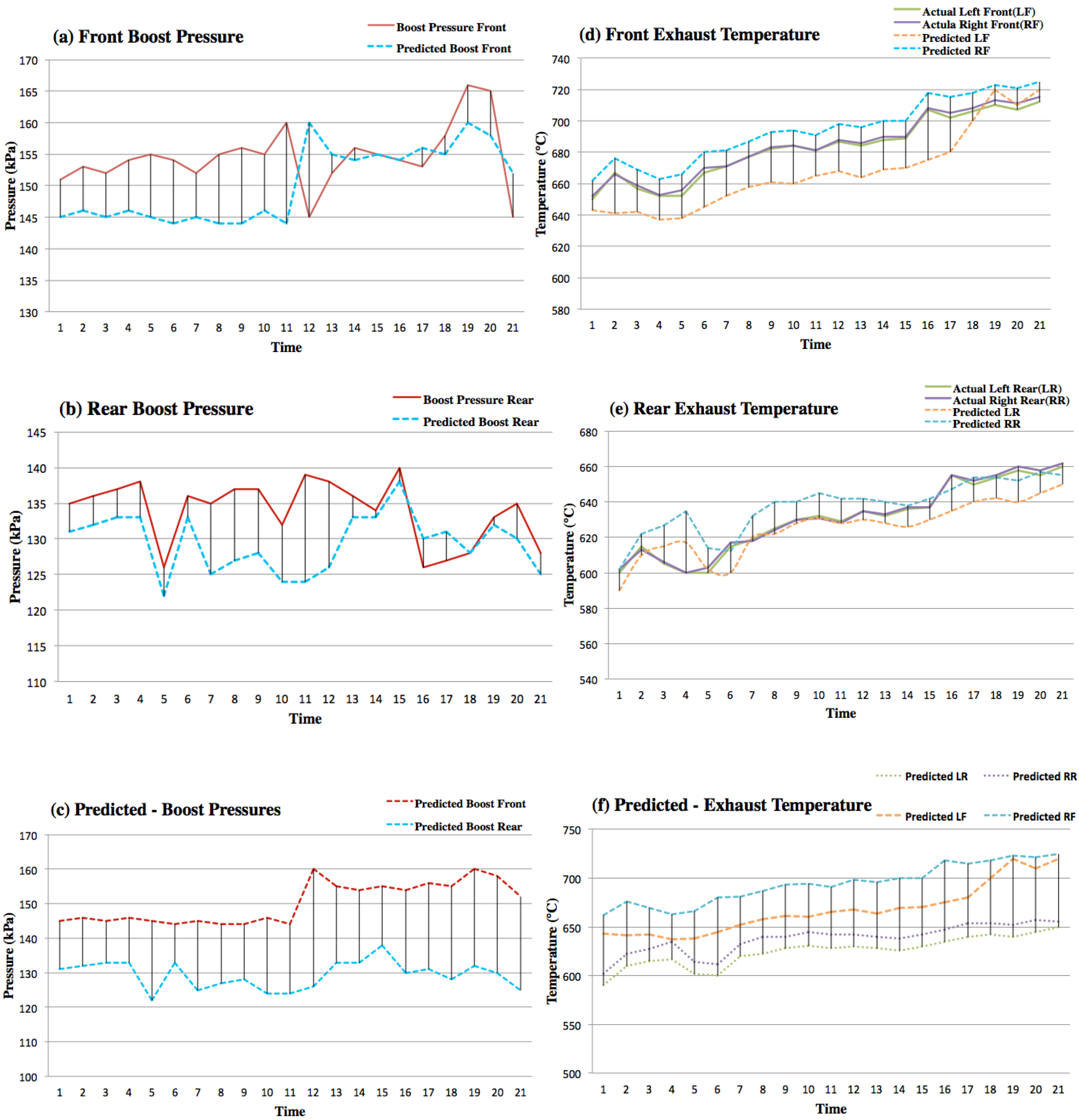

Figure 2. Experimental Results: (a) Front boost pressure vs. Time, (b) Rear boost pressure vs. Time, (c) Predicted boost pressures vs. Time, (d) Front Exhaust Temperature vs. Time, (e) Rear Exhaust Temperature vs. Time, (f) Predicted Exhaust Temperature vs. Time

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Lijie Cao. Singular value decomposition applied to digital image processing. *Division of Computing Studies, Arizona State University Polytechnic Campus, Mesa, Arizona State University polytechnic Campus*, 2006.
2. Hesam Izakian. *Cluster-Centric Anomaly Detection and Characterization in Spatial Time Series*. PhD thesis, University of Alberta, 2014.
3. Dan Kalman. A singularly valuable decomposition: the svd of a matrix. *The college mathematics journal*, 27(1):2–23, 1996.
4. John G Kemeny and James Laurie Snell. *Finite markov chains*, volume 356. van Nostrand Princeton, NJ, 1960.
5. Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.
6. Esa Nummelin. *General irreducible Markov chains and non-negative operators*, volume 83. Cambridge University Press, 2004.
7. Wallodi Weibull. A statistical distribution function of wide applicability. *Journal of Applied Mechanics*, 18:293–297, 1951.
8. Guoliang Zeng. Facial recognition with singular value decomposition. In *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, pages 145–148. Springer, 2007.