Titre: Title:	Analytics of Sequential Time Data from Physical Assets
Auteur: Author:	Ahmed Elsheikh
Date:	2018
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Elsheikh, A. (2018). Analytics of Sequential Time Data from Physical Assets [Ph.D. thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/3122/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/3122/
Directeurs de recherche: Advisors:	Soumaya Yacout, & Mohamed-Salah Ouali
Programme: Program:	Doctorat en génie industriel

UNIVERSITÉ DE MONTRÉAL

ANALYTICS OF SEQUENTIAL TIME DATA FROM PHYSICAL ASSETS

AHMED ELSHEIKH
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L’OBTENTION
DU DIPLÔME DE PHILOSOPHIAE DOCTOR
(GÉNIE INDUSTRIEL)
AVRIL 2018

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

ANALYTICS OF SEQUENTIAL TIME DATA FROM PHYSICAL ASSETS

présentée par : ELSHEIKH Ahmed
en vue de l'obtention du diplôme de : Philosophiae Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. ADJENGUE Luc Désiré, Ph. D., président
Mme YACOUT Soumaya, D. Sc., membre et directeur de recherche
M. OUALI Mohamed-Salah, Doctorat, membre et codirecteur de recherche
M. BASSETTO Samuel Jean, Doctorat, membre
M. ABDELGAWAD Hossam, Ph. D., membre externe

RÉSUMÉ

Avec l’avancement dans les technologies des capteurs et de l’intelligence artificielle, l'analyse des données est devenue une source d’information et de connaissance qui appuie la prise de décisions dans l’industrie. La prise de ces décisions, en se basant seulement sur l’expertise humaine n’est devenu suffisant ou souhaitable, et parfois même infaisable pour de nouvelles industries. L'analyse des données collectées à partir des actifs physiques vient renforcer la prise de décisions par des connaissances pratiques qui s’appuient sur des données réelles. Ces données sont utilisées pour accomplir deux tâches principales; le diagnostic et le pronostic.

Les deux tâches posent un défi, principalement à cause de la provenance des données et de leur adéquation avec l’exploitation, et aussi à cause de la difficulté à choisir le type d'analyse. Ce dernier exige un analyste ayant une expertise dans les déférentes techniques d’analyse de données, et aussi dans le domaine de l’application. Les problèmes de données sont dus aux nombreuses sources inconnues de variations interagissant avec les données collectées, qui peuvent parfois être dus à des erreurs humaines. Le choix du type de modélisation est un autre défi puisque chaque modèle a ses propres hypothèses, paramètres et limitations.

Cette thèse propose quatre nouveaux types d'analyse de séries chronologiques dont deux sont supervisés et les deux autres sont non supervisés. Ces techniques d'analyse sont testées et appliquées sur des différents problèmes industriels. Ces techniques visent à minimiser la charge de choix imposée à l'analyste.

Pour l’analyse de séries chronologiques par des techniques supervisées, la prédiction de temps de défaillance d’un actif physique est faite par une technique qui porte le nom de ‘Logical Analysis of Survival Curves (LASC)’. Cette technique est utilisée pour stratifier de manière adaptative les courbes de survie tout au long d’un processus d’inspection.

Ceci permet une modélisation plus précise au lieu d'utiliser un seul modèle augmenté pour toutes les données. L'autre technique supervisée de pronostic est un nouveau réseau de neurones de type ‘Long Short-Term Memory (LSTM) bidirectionnel’ appelé ‘Bidirectional Handshaking LSTM (BHLSTM)’. Ce modèle fait un meilleur usage des séquences courtes en faisant un tour de ronde à travers les données. De plus, le réseau est formé à l'aide d'une nouvelle fonction objective axée sur la sécurité qui force le réseau à faire des prévisions plus sûres. Enfin, étant donné que LSTM est une technique supervisée, une nouvelle approche pour générer la durée de vie utile restante

(RUL) est proposée. Cette technique exige la formulation des hypothèses moins importantes par rapport aux approches précédentes.

À des fins de diagnostic non supervisé, une nouvelle technique de classification interprétable est proposée. Cette technique est intitulée ‘Interpretable Clustering for Rule Extraction and Anomaly Detection (IC-READ)’. L'interprétation signifie que les groupes résultants sont formulés en utilisant une logique conditionnelle simple. Cela est pratique lors de la fourniture des résultats à des non-spécialistes. Il facilite toute mise en œuvre du matériel si nécessaire. La technique proposée est également non paramétrique, ce qui signifie qu'aucun réglage n'est requis. Cette technique pourrait également être utiliser dans un contexte de ‘one class classification’ pour construire un détecteur d'anomalie. L'autre technique non supervisée proposée est une approche de regroupement de séries chronologiques à plusieurs variables de longueur variable à l'aide d'une distance de type ‘Dynamic Time Warping (DTW)’ modifiée. Le DTW modifié donne des correspondances plus élevées pour les séries temporelles qui ont des tendances et des grandeurs similaires plutôt que de se concentrer uniquement sur l'une ou l'autre de ces propriétés. Cette technique est également non paramétrique et utilise la classification hiérarchique pour regrouper les séries chronologiques de manière non supervisée. Cela est particulièrement utile pour décider de la planification de la maintenance. Il est également montré qu'il peut être utilisé avec ‘Kernel Principal Components Analysis (KPCA)’ pour visualiser des séquences de longueurs variables dans des diagrammes bidimensionnels.

ABSTRACT

Data analysis has become a necessity for industry. Working with inherited expertise only has become insufficient, expensive, not easily transferable, and mostly unavailable for new industries and facilities. Data analysis can provide decision-makers with more insight on how to manage their production, maintenance and personnel. Data collection requires acquisition and storage of observatory information about the state of the different production assets. Data collection usually takes place in a timely manner which result in time-series of observations.

Depending on the type of data records available, the type of possible analyses will differ. Data labeled with previous human experience in terms of identifiable faults or fatigues can be used to build models to perform the expert's task in the future by means of supervised learning. Otherwise, if no human labeling is available then data analysis can provide insights about similar observations or visualize these similarities through unsupervised learning. Both are challenging types of analyses.

The challenges are two-fold; the first originates from the data and its adequacy, and the other is selecting the type of analysis which is a decision made by the analyst. Data challenges are due to the substantial number of unknown sources of variations inherited in the collected data, which may sometimes include human errors. Deciding upon the type of modelling is another issue as each model has its own assumptions, parameters to tune, and limitations.

This thesis proposes four new types of time-series analysis, two of which are supervised requiring data labelling by certain events such as failure when, and the other two are unsupervised that require no such labelling. These analysis techniques are tested and applied on various industrial applications, namely road maintenance, bearing outer race failure detection, cutting tool failure prediction, and turbo engine failure prediction. These techniques target minimizing the burden of choice laid on the analyst working with industrial data by providing reliable analysis tools that require fewer choices to be made by the analyst. This in turn allows different industries to easily make use of their data without requiring much expertise.

For prognostic purposes a proposed modification to the binary Logical Analysis of Data (LAD) classifier is used to adaptively stratify survival curves into long survivors and short life sets. This model requires no parameters to choose and completely relies on empirical estimations. The proposed Logical Analysis of Survival Curves show a 27% improvement in prediction accuracy

than the results obtained by well-known machine learning techniques in terms of the mean absolute error.

The other prognostic model is a new bidirectional Long Short-Term Memory (LSTM) neural network termed the Bidirectional Handshaking LSTM (BHLSTM). This model makes better use of short sequences by making a round pass through the given data. Moreover, the network is trained using a new safety oriented objective function which forces the network to make safer predictions. Finally, since LSTM is a supervised technique, a novel approach for generating the target Remaining Useful Life (RUL) is proposed requiring lesser assumptions to be made compared to previous approaches. This proposed network architecture shows an average of 18.75% decrease in the mean absolute error of predictions on the NASA turbo engine dataset.

For unsupervised diagnostic purposes a new technique for providing interpretable clustering is proposed named Interpretable Clustering for Rule Extraction and Anomaly Detection (IC-READ). Interpretation means that the resulting clusters are formulated using simple conditional logic. This is very important when providing the results to non-specialists especially those in management and ease any hardware implementation if required. The proposed technique is also non-parametric, which means there is no tuning required and shows an average of 20% improvement in cluster purity over other clustering techniques applied on 11 benchmark datasets. This technique also can use the resulting clusters to build an anomaly detector.

The last proposed technique is a whole multivariate variable length time-series clustering approach using a modified Dynamic Time Warping (DTW) distance. The modified DTW gives higher matches for time-series that have the similar trends and magnitudes rather than just focusing on either property alone. This technique is also non-parametric and uses hierarchal clustering to group time-series in an unsupervised fashion. This can be specifically useful for management to decide maintenance scheduling. It is shown also that it can be used along with Kernel Principal Components Analysis (KPCA) for visualizing variable length sequences in two-dimensional plots.

The unsupervised techniques can help, in some cases where there is a lot of variation within certain classes, to ease the supervised learning task by breaking it into smaller problems having the same nature.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AAE	Asymmetric Absolute Error
ANN	Artificial Neural Networks
ARI	Adjusted Rand Index
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
ASE	Asymmetric Squared Error
BHLSTM	Bidirectional Handshaking LSTM
CBM	Condition Based Maintenance
CH	Calinski–Harabasz
CMAPSS	Commercial Modular Aero-Propulsion System Simulation
DB*	Modified Davies–Bouldin
DDTW	Differential Dynamic Time Warping
DFT	Discrete Fourier transform
DNF	Disjunctive Normal Form
DTR	Decision Tree Regression
DTW	Dynamic Time Warping
DWT	Discrete Wavelet Transform
ED	Euclidean Distance
EM	Expectation-Maximization
ETR	Ensemble Tree Regression
FCM	Fuzzy C-Means
FFNN	Feed Forward Neural Network
GMM	Gaussian Mixture Model
GPR	Gaussian Process Regression
HC	Hierarchal clustering
HMM	Hidden Markov Model
IBL	Instance Based Learning

IC-READ	Interpretable Clustering for Rule Extraction and Anomaly Detection
IRI	International Roughness Index
KM	Kaplan-Meier
KM	K-Means
KPCA	Kernel Principal Components Analysis
LAD	Logical Analysis of Data
LASC	Logical Analysis of Survival Cuves
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MSE	Mean Squared Error
MTMDET	Ministère des Transports de la Mobilité Durable et de l'Électrification des Transports
MTS	Multivariate Time-Series
NN	Neural Network
OCC	One-Class Classification
OCCSVM	One-Class Classification Support Vector Machine
PCA	Principal Components Analysis
PDDP	Principal Direction Divisive Partitioning
PM	Predictive Maintenance
PSD	Power Spectral Density
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
SF	Scoring Function
Sil	Silhouette Index
SVM	Support Vector Machine
SVR	Support Vector Regression
TQ	Transport Quebec
UTS	Univariate Time-series

VLMTS	Variable Length Multivariate Time-Series
VLUTS	Variable Length Univariate Time-series

CHAPTER 1 INTRODUCTION

Since the beginning of the industrial revolution in the seventieth century, various physical assets came to existence and have been growing in number ever since. Maintenance evolved over the history of industry starting from replacement on failure, to scheduled maintenance, Condition Based Maintenance (CBM) (Vogl, Weiss, and Helu 2016), and finally to Predictive Maintenance (PM). Every physical asset, no matter how good its crafting quality is, undergoes degradation. The difference is rate of degradation (Jardine, Lin, and Banjevic 2006). This means that intervention for maintenance or replacement is inevitable. As the industry scale grew as well as the number and variety of physical assets increased, the need for wise management increased (Canfield 1986). Furthermore, compromising asset replacement, maintenance, and downtime cost with profit has become a critical issue (Venkatasubramanian 2005).

CBM is currently the most abundant working maintenance system due to its relative simplicity as it only requires few sensors, reduced cost as it does not require intensive data collection, and it shows noticeable improvements in reducing production costs and increasing uptime (Ahmad and Kamaruddin 2012). On the other hand, PM aims not only at detection of near failures but also at predicting the future behavior given current working conditions. Having an insight about the asset's behavior allows more efficient management and production (Lee, Ghaffari, and Elmeligy 2011; Saxena et al. 2010).

PM faces a multitude of challenges that needs to be overcome in order to obtain reliable performance (Vogl, Weiss, and Helu 2016). Firstly, data acquisition is not yet optimal in most scenarios (Hess, Calvello, and Frith 2005). This is because most transducers are affected by their surrounding environment which is inherited from their semi-conductor nature. In turn, clever processing proceeded by smart decision making is a must to overcome the inherited electronic sensor problems.

Secondly, the system complexity and interaction between its various subsystems may not be well represented by a simple model (Hess, Calvello, and Frith 2005). Also, some systems exhibit multiple modes of operation which are not controlled or intentional due to the working environment. Examples of such assets are city roads with different usage rates and weather conditions (Ben-Akiva and Ramaswamy 1993), and aircraft engines (Korson et al. 2003). These assets undergo change and sometimes end up in unknown, or unexpected operating conditions.

Uncovering groups of behavior and finding structure in the gathered data can be beneficial for decision making as well as making design precautions, and maintenance planning.

Experience is gained throughout time. Hence, to make better decisions regarding the future, one must make use of the previous experiences. Therefore, the regard for time adaptive prognostics can result in better management decisions. The growing data availability from different industrial plants, and the advancements in its acquisition techniques opens up a myriad of opportunities (Abellan-Nebot and Subrión 2009). Discovering a hidden structure in newly observed raw data can be very beneficial to proactive decision making (Jardine et al. 2015; Vogl, Weiss, and Helu 2016).

Data concern all available observations that characterize the system state condition and its behavior over time. For some applications such as machine diagnosis (Mohamad-ali, Yacout, and Lakis 2014), or product quality inspections (Tiejian Chen et al. 2016), the data collected at different time intervals can be treated as independent. However, when the sequence of observations is highly correlated (Ghasemi, Yacout, and Ouali 2010) as normally expected from degradation and maintenance processes, these observations have to be considered as time-series.

There are two main types of information that can be extracted by time-series analysis of physical-asset data; diagnostic and prognostic. Diagnosis is mainly concerned about monitoring the current state of the working physical asset giving indication of its current health state or operation state. On the other hand, prognosis is oriented towards prediction of future states of the working asset. Predictions are more insightful and hence can help achieve more efficient production (Guillén et al. 2016).

For both diagnosis and prognosis, dealing with time-series instead of instantaneous readings independently means that the relation between observations through time should be modelled and not just the mapping between the observations and the targeted diagnostic or prognostic objective. This modelling can be either explicit in the used model or implied by how the time-series is inputted to the model.

1.1 Data in Industry

Data collected by industrial firms are mainly collected in a timely manner. The data collected is some sort of observable aspect that can be either:

1- Related to the assets itself such as temperature, or

2- Related to the load exerted on the asset such as torque.

Data can be collected in a fully automatic, semi-automatic, or manual as follows:

1- Fully automatic data collection means that the sensor is placed on the assets such as a vibration sensor or near it such as an acoustic emission sensor and the sensor readings are continuously stored in some database.

2- Semi-automatic data collection means that the sensor needs to be moved to the asset to make the desire data collection. This is the case when the desired useful aspects of the asset require placement of a large number of expensive sensors such as laser sensors used for road inspection. Of course, it is not practical to place such sensors on all roads, instead the sensor is vehicle mounted and moves to each location.

3- Manual data collection requires a human expert to go and inspect the asset and estimate certain aspects for the asset based on observation or according to a certain inspection code. Another type of manual recording is the recording of human intervention to do maintenance for example.

Collection of data is vulnerable to errors due to various reasons such as:

- Interaction between different assets with each other makes the sensor gather the effect of interaction and not purely the readings for the observed asset.

- Interaction between the assets and sensors with the environment. Electronic sensors change their characteristics with temperature for example.

- Intervention of a human factor in the data collection procedure. Even in the case of fully automated data collection, a human expert sometimes intervenes to identify the observations that experience a certain damage, fault, or state of operation. The human factor is the highest source of errors in data collection. Human errors can be in the form of missing data, wrong identification, or misuse of the portable sensor, which in turn can be hard to recover or even sometimes misleading.

Handling several types of errors is considered as a pre-processing step to clean the data and restore as much as possible from the originally intended data for the subsequent analysis step. Another intermediate step prior to analysis is called feature extraction, which aims at feeding the analysis step with lesser varying information such as extracting the frequencies evident in a vibration signal instead of giving the analysis step the raw sensor recordings at each instant.

Analyzing data means either extracting useful information from the data or fulfilling a certain task such as classifying modes of operation based on historically labeled data. Data analysis can either be done offline, which means after data collection has ended the analysis uses these records to build a model, or online, which means that the analysis model is updated concurrently with the data collection.

When data analysis is given a specific task to learn then it is called supervised learning. On the other hand, if no specific task is required then it is called unsupervised learning and aims at finding interesting relations between different training records.

1.1.1 Variation in Time-Series Lengths

Recorded data usually have variable lengths, like for example when the inspected assets have variable run-to-failure spans. This variation in the length of the collected sequences of observations is handled by one of the following four ways:

1) Truncation, which means trimming all of the data to match the shortest sequence.
2) Padding, which means adding filler values to the data, like zeros for example, to match the maximum length sequence of data.
3) Windowing, which means analyzing shorter sub-sequences from the original sequence, such that these windows of observation have equal lengths.
4) Using techniques that can perform analysis on variable length sequences.

The first and second ways of handling are not suitable for sequences of large variation since the information they contain are either lost by truncation or distorted by too much padding. Nevertheless, the online type of analysis is obliged to use the windowing approach since the outcomes of the analysis are required concurrently with the monitoring process.

1.1.2 Types of Time-Series Data Available in Industry

There are several types of time-series data that can be available from industry. Here the main types of data that are encountered in industry:

1- Run-to-failure data: this type of data contains a sequence of observations that were collected as the assets started working in the system at full health, until it failed. The identification and definition of failure can be different from one system to another, even for the same asset. The identification then requires an external observer to label or annotate at a certain time the type of failure for the asset. This external observer might be a human or another device such as a computer vision camera.

2- Truncated data: this type of data is the same as the above, but there is no clear indication at what initial state the asset was when the data collection started. This is the most practical case because the asset can enter the system in a used state, undergo partial maintenance, or the manufacturing of the asset itself is not perfect, which is usually the case.

3- Data collected without annotation: this type of data has no labelling of events given from an external observer to identify certain types of failure or fatigue to the asset during its monitoring.

4- Non-event data: this type of data is either collected from critical assets that undergo extensive preventive maintenance, such as nuclear reactors, or from new assets that were not used before, such as a new power generator. Data available from this scenario usually belongs to one mode of operation, which is normal operation. This sort of data can be useful for anomaly detection, which is out of the norm behavior.

1.1.3 Types of Analysis Tasks in Industry

Supervised Learning Analysis

Supervised analysis in industry is either diagnostic or prognostic. Diagnosis is concerned with identifying the current state of the asset while prognosis tries to predict the future state(s) of the asset. Both tasks require the historical data observations to be labeled with the states or conditions that will be estimated by the analysis. In other words, supervised data analysis tries to identify relationships between the observations and the labels to perform the task on its own to decrease the reliance on human expertise.

Moreover, automated diagnosis and prognosis provide industrial systems with an around the clock inspection for any number of assets without the need of a huge number of expert technicians working in shifts.

Unsupervised Learning Analysis

This type of analysis is specifically useful for either new industries, or old industries trying to make use of historical data. Since for both cases labelling was not taken into consideration while collecting the data or there are no experts for this new type of data, both end up with just records of observations. In this case there are two types analyses to be conducted:

1- Clustering: which means grouping different records of data that show similar behavior or similar readings. Of course, similar in this context does not mean exactly the same, but rather a like.

2- Anomaly/Novelty detection: This means that it is known that the recorded data belonged only to one condition, which is typically the normal operating condition. Hence, the unsupervised data analysis task in this case is to identify if asset is in a state that is out of the norm which was seen before.

1.1.4 Potentials for Data Analysis in Industry

Industrial capital is mainly divided into materials, physical assets and personnel. Each of these components contribute to the total expenditure with a varying proportion depending on the type of industry. Data analysis can provide the decision makers with insights on how to benefit the most from each component in their industrial firm. Faults and failures can result in a lot of wasted material, costly maintenance, and prolonged downtime losses (Venkatasubramanian 2005; Vogl, Weiss, and Helu 2016). Making accurate in-time predictions allows the industry to benefit the most from their assets without neither making early unnecessary maintenance nor late replacement maintenance. The value analysis of maintenance show an exponential increase when proper intervention takes place (Haddad, Sandborn, and Pecht 2012).

A study on the impact of maintenance on competitiveness and profitability of weaving industry show that an improvement in production quality comes as a by product from better maintenance, as well as a reduction in the number of required spare parts (Maletič et al. 2014).

1.1.5 Anticipated Gains from Data Analysis

There are yet very few studies that measure the benefits of applying data analysis to industrial profit although it has been already applied in many industries.

This is due to the variability in the different types of industries and their decision-making procedures.

Nevertheless, for electronic asset management, studies show up to 22% maintenance cost reduction (Scanff et al. 2007), and a return on investment up to 300% (Feldman, Sandborn, and Jazouli 2008; Kent and Murphy 2000).

Although previous end-to-end studies are limited, there are other options for new industries willing to incorporate data analysis into their studies to identify potential profit through simulations (Gilabert et al. 2017).

1.2 Problem Statement

Many industries nowadays have managed to collect significant amount of data and are willing to explore their potential. Machine learning tools, especially those concerned with time-series analysis have been exploited throughout literature to show their capabilities of making use of data for different purposes (Javed, Gouriveau, and Zerhouni 2017; Sikorska, Hodkiewicz, and Ma 2011). There is no single machine learning technique that has shown to be suitable for all types of data in terms of type, amount, and application (Sikorska, Hodkiewicz, and Ma 2011; Wolpert 1996).

The different combinations between the analysis options, and the required tasks burdens the data analyst with a multitude of decisions to make. Furthermore, each technique has its interior parameters to tune. As a result, in many situations analysts resort to either easy or commonly known analysis techniques.

There are four main choices made by the analyst to achieve his goal:

1- The type of model to be used; whether the data is of sequential nature, labeled or not, what is the target of the analysis, and how is the model going to be used after being trained.

2- The type of data preprocessing and feature extraction to be used; more complex data preprocessing and feature extraction relives the requirement of complex analysis models.

3- The model capacity to be used given a certain amount of data; the higher the capacity of the model is, the more data it requires to be trained.

4- The choice of hyper-parameters of the selected model; almost all machine learning models require parameter tuning or make some assumptions to avoid them, but how many are they, and how sensitive the performance of the model is to these hyper-parameters.

1.3 General Objective

This thesis aims at providing data analysts with versatile tools to solve different types of problems frequently encountered in industry. This set of tools are mainly concerned with sequential data analysis which arise in most industrial applications (Jardine, Lin, and Banjevic 2006). The proposed techniques are oriented towards relieving the data analyst from the burden of making a lot of choices to perform his analysis.

This thesis proposes new techniques for sequential data analysis for solving diagnostic and prognostic problems frequently encountered in industry. These techniques target decreasing one or more of the main aforementioned choices that an analyst have to decide while performing as well as other commonly used techniques for different industrial applications. The new techniques are intended to decrease the analyst’s burden in different manners as follows:

1- Having no or a very small number of hyper-parameters to tune. This can be achieved by allowing the model to expand its capacity according to the given data or estimating the required parameters from the data.

2- Providing the results in a simple comprehensible form. This can be achieved by producing the results as a simple set of conditions for example, or by providing visualization methods for the results. This allows better understanding of the results for non-specialists and practitioners. Moreover, it simplifies hardware implementation when required.

3- Solving several problems simultaneously instead of having different analysis techniques for each objective.

1.4 Specific Objectives

To achieve the general objective, four new sequential data analysis techniques were developed to lessen the number of decisions that an analyst must make when solving various industrial applications as follows:

1- Predicting when a time-series will reach a certain threshold *without requiring any assumptions about the time-series generating process nor the threshold-reaching probability distribution*. Moreover, making the estimation *easy to comprehend* in terms of Boolean logic.

2- Predicting when a time-series will reach a certain threshold using only *partial observations*, *without requiring any explicit assumptions about the initial observation state*.

3- Proposing a distance measure for variable length time-series that focuses on the *time-series trend*, which means taking both the magnitude and the way it changes into consideration when comparing series. The new distance should be *non-parametric* to be used along with unsupervised learning techniques such as clustering and dimensionality reduction for *visualizing variable-length multivariate time-series*.

4- Proposing a completely autonomous *unsupervised rule extraction* technique that can *identify automatically the number of clusters* and represent them in *simple comprehensible Boolean logic* format. Also, using it to identify regions of normal operation to be uses as an *anomaly detector*.

1.5 Originality of Research

To the best of our knowledge none of the following was previously exploited in literature:

1- Using LAD along with adaptive stratifying thresholds with multivariate sensor readings and multiple operating conditions to estimate the RUL. Moreover, it has not been applied to Titanium composite cutting tools applications.

2- Introducing a new LSTM architecture that is bidirectional but goes a round pass on the data by allowing the processing units of one direction initialize those going back in the reverse direction.

3- Introducing a new objective function for training neural networks which penalizes early estimates less than late ones to give safer estimates.

4- Introducing a new RUL target generation approach that requires only one assumption that the RUL remains constant until a number of cycles equal to that of the minimum life span of the given dataset. It generates the target RUL using some manipulation of the given sensor readings.

5- Modifying DTW to focus on the profile of the time-series using a two-step approach. The first step is DDTW alignment using the warping path, and the second it ED distance calculation. Moreover, it has not been used along with hierarchal clustering, nor has it been applied to road maintenance management applications.

6- The clustering technique uses an N-best search method which allows controlling the amount of computations versus the anticipated accuracy of clustering.

7- Proposing a new anomaly detector for monitoring working assets using the proposed interpretable clustering IC-READ.

8- Implementing a clustering technique, that extracts logical rules in Disjunctive Normal Form (DNF) from unlabeled data using iterative data binarization using marginal histograms and has a built-in number of clusters' estimator.

1.6 Thesis Organization

The thesis is divided into eight chapters. The first chapter, which is this one, introduces the problems addressed by this thesis and its scope and objectives. Chapter 2 reviews the general framework of machine learning applications in industrial engineering, the different approaches investigated in literature, and highlights the pros and cons of each approach.

Chapters 3 and 4 focus on supervised learning techniques. Chapter 3 introduces the logical analysis of survival curves, and how to use it for Remaining Useful Life (RUL) estimation. The experiment is conducted on RUL estimation for cutting tools working on Titanium composite materials. The results show how the survival curves are adaptively estimated and how the rules can be extracted to separate long survivors from short-life ones. Chapter 4 introduces the bidirectional handshaking long short-term memory network architecture, and how it is used along with the new proposed safety oriented objective function and the proposed automatic target generation process for RUL estimation for machines with unknown health index. The experiments are run against the NASA turbo engine dataset.

Chapters 5 and 6 focus on unsupervised learning techniques. Chapter 5 introduces the profile favoring dynamic time warping distance measure for variable length multivariate time-series matching. It is shown how the proposed distance can be used for clustering and visualization. Chapter 6 introduces the IC-READ technique and compares its performance with other commonly known clustering techniques on standard datasets. Also, its performance as an anomaly detector is examined.

Chapter 7 discusses the main findings in this thesis, and Chapter 8 summarizes the findings and provides recommendations for users of the techniques developed in this thesis.

1.7 Deliverables

The outcomes of this thesis are four article papers, each covering one of the specific objectives as follows:

1- "Failure Time Prediction Using Logical Analysis of Survival Curves and Multiple Machining Signals of a Nonstationary Process"

- Authors: Ahmed Elsheikh; Soumaya Yacout, D. Sc.; Mohamed Salah Ouali, Ph. D.
- Submitted to Journal of Intelligent Manufacturing (JIMS) on 3rd of Jan 2018. The paper is accepted with minor modification.
- Abstract: "This paper develops a prognostic technique that is called the Logical Analysis of Survival Curves (LASC). The technique is used for failure time (T) prediction. It combines the reliability information that is obtained from a classical Kaplan-Meier (KM) non-parametric curve, to that obtained from online measurements of multiple sensed signals. The analysis of these signals by the Logical Analysis of Data (LAD), which is a machine learning technique, is performed in order to exploit the instantaneous knowledge that is available about the process under study. The experimental results of failure times' prediction of cutting tools are reported. The results show that LASC prognostic results are comparable or better than the results obtained by well-known machine learning techniques. Other advantages of the proposed techniques are discussed."

2- "Bidirectional Handshaking LSTM for Remaining Useful Life Prediction"

- Authors: Ahmed Elsheikh; Soumaya Yacout, D. Sc.; Mohamed Salah Ouali, Ph. D.
- Submitted to Neurocomputing Journal on 1st of Mar 2018.
- Abstract: "Unpredictable failures and unscheduled maintenance of physical systems increases production resources, produces more harmful waste for the environment, and increases system life cycle costs. Efficient Remaining Useful Life (RUL) estimation can alleviate such an issue. The RUL is predicted by making use of the data collected from several types of sensors that continuously record different indicators about a working asset, such as vibration intensity or exerted pressure. This type of continuous monitoring data is sequential in time, as it is collected at a certain rate from the sensors during the asset's work. Long Short-Term Memory (LSTM) neural network models have been demonstrated to be efficient throughout the literature when dealing with sequential data because of their ability to retain a lot of information over time about previous states of the system. This paper proposes using a new LSTM architecture for predicting the RUL when given short sequences of monitored observations with random initial wear. By using LSTM, this paper proposes a new objective function that is suitable for the RUL estimation problem, as well as a new target generation approach for training LSTM networks, which requires making lesser assumptions about the actual degradation of the system."

3- "A Profile Favoring Dynamic Time Warping for the Hierarchical Clustering of Systems Based on Degradation"

- Authors: Ahmed Elsheikh; Mohamed Salah Ouali, Ph. D.; Soumaya Yacout, D. Sc.
- Submitted to the Computer and Industrial Engineering (CAIE) Journal on the 21st of Sep 2017.
- Abstract: "Efficient management of a deteriorating system requires accurate decisions based on in situ collected information to ensure its sustainability over time. Since some of these systems experience the phenomenon of degradation, the

collected information usually exhibits specific degradation profiles for each system. It is of interest to group similar degradation trends, characterized by a multivariate time-series of collected information over time, to plan group actions. Knowledge discovery is one of the essential tools for extracting relevant information from raw data. This paper provides a new method to cluster and visualize multivariate time-series data, which is based on a modified Dynamic Time Warping (DTW) distance measure and hierarchal clustering. The modification adds emphasis on finding the similarity between time-series that have the same profiles, rather than by magnitudes over time. Applied to the clustering of deteriorating road segments, the modified DTW provides better results according to different cluster validity indices when compared to other forms of DTW distance technique."

4- "Interpretable Clustering for Rule Extraction and Anomaly Detection: A New Unsupervised Rule Induction Technique"

- Authors: Ahmed Elsheikh; Soumaya Yacout, D. Sc.; Mohamed Salah Ouali, Ph. D.
- Submitted to Expert Systems With Applications (ESWA) Journal on 28th Feb 2018.
- Abstract: "This paper proposes a modified unsupervised divisive clustering technique to achieve interpretable clustering. The technique is inspired by the supervised rule-induction techniques such as Logical Analysis of Data (LAD), which is a data mining supervised methodology that is based on Boolean analysis. Each cluster is identified by a simple set of Boolean conditions in terms of the input features. The interpretability comes as a byproduct when assuming that the boundaries between different clusters are parallel to the features' coordinates. This allows the clusters to be identified in Disjunctive Normal Form (DNF) logic where each Boolean variable indicates greater or less than conditions on each of the features. The proposed technique mitigates some of the limitations of other existing techniques. Moreover, the clusters identified by the proposed technique are adopted to solve anomaly detection problems. The performance of the proposed technique is compared to other well-known clustering techniques, and the extracted rules are compared to decision trees which is a supervised rule-induction technique."

CHAPTER 2 CRITICAL LITERATURE REVIEW

Several attempts throughout literature were made for the purpose of analysis and knowledge discovery from time-series data (Aghabozorgi, Seyed Shirkhorshidi, and Ying Wah 2015; Fu 2011). The type of analysis varies along with the required task and the way to tackle it. Machine learning is a set of techniques for approximating mapping functions between a certain input and a required output. The two main tasks in machine learning are either (Murphy 2012):

1. Supervised, where it is usually required to discriminate or make estimates for future tests based on sets of historically collected data.
2. Unsupervised, where there is no specific task required to be achieved, but rather a search for interesting information that could be extracted from the data. Hence, the objective is mostly to find groups within the historically collected data that are similar using a certain measure of similarity.

Mostly, machine learning techniques are not oriented towards data having a sequential nature. Nevertheless, there are several ways to deal with sequential data for non-sequential models, which are models that do not explicitly model the relation between observations in time (Bishop 2006). The most important approach is called windowing, which analyzes chunks of the data in sequential manner. This means that the input to the learning technique is always fixed to a certain number of previous observations in time.

The techniques used for solving these tasks can be divided into five main categories in terms of the theoretical approach used to achieve the required function approximation:

1- Statistical modelling, where a family of probability distribution functions having a set of parameters that are adapted to best represent the experimental data.

2- Discriminative modelling, where the independent variables are assumed to have a linear or non-linear relation to the output in either direct or interactive form.

3- Instance-based learning, where experimental results are archived and used as point-wise approximation for the target function.

4- Rule induction, where the target is to learn human readable rules for classification. Boolean logic in the disjunctive normal form is the most popular for this category.

2.1 Statistical Modelling

When using this type of modelling for time-series analysis, the analyst assumes that sequential data are generated from a certain probability distribution or a random process of a certain nature. The modelling tries to estimate the parameters of the hypothesis distribution. The estimation tries to maximize the joint probability of the given targets and the historical data.

Bayes theory which is one of the pillars of probabilistic modelling has the following statement

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)} \qquad 1$$

Where D stands for data given, θ stands for the hypothesized model, and P stand for the probability. We can interpret this formulation by saying: the posterior probability of our model after observing the data, is equal to the probability of the likelihood of this data being generated by our model, multiplied by our prior knowledge about this model, and divided by the probability of occurrence of this data.

Estimation of the model θ is an optimization problem. We want to maximize the posterior probability of the model given the data. In that case we can neglect $P(D)$ since it is a common factor for all models. There are two types of parameter estimation: (1) Maximum Likelihood Estimation which assumes that all parameter values are possible, and (2) Maximum A Posteriori estimation which incorporates prior knowledge about the parameters to favor certain parameters over others (Bishop 2006).

The most popular statistical modelling scheme for time-series is the Hidden Markov Models (HMMs) (Ghasemi, Yacout, and Ouali 2010; Rabiner 1989). The HMM assumes that the time-series are random process, which change their distribution over time. The HMM models assumes a set of discrete states such that each state has a different observation probability. It also assumes that the transition from one state to another is stochastic and depends solely on the previous state which is the Markov property. These assumptions are not always verifiable. Moreover, choosing the number of the discrete hidden states and their observation probability are not always clear and have to be chosen by trial and error (Aghabozorgi, Seyed Shirkhorshidi, and Ying Wah 2015).

2.1.1 Generative modelling

All statistical modelling techniques aim at approximating the likelihood of the data $P(D|\theta)$ which means modelling the joint probability of the required targets y and the inputs x, $P(x, y|\theta)$. One of the merits of this type of modelling is that by approximating the joint probability of the data and their corresponding targets, the models can be used to generate sample data having the same statistical nature as the data it was trained on. Nevertheless, this adds more burden on to the training phase, as the model will implicitly model the data distribution. Moreover, most statistical models require the calculation of some intractable integrals which then leads to the requirement of using sampling techniques (Murphy 2012).

2.2 Discriminative Modelling

Unlike generative modelling, discriminative modelling techniques aim at approximating the posterior probability of the target given the input $P(y|x;\theta)$ directly. This means that there is no explicit assumption about the likelihood distribution, which gives this type of modelling a superior performance in most cases (Ng and Jordan 2001).

The discriminant functions for the two-class case, which can be generalized to the multi-class case (Webb 2003), can be formulated as follows:

$$\gamma(\boldsymbol{x}) \begin{cases} > 0 & \rightarrow \boldsymbol{x} \in \ C_1 \\ < 0 & \rightarrow \boldsymbol{x} \in C_2 \end{cases} \qquad 2$$

Where $\gamma(\boldsymbol{x})$ is the discriminant function, $\boldsymbol{x}$ is the observed input vector, and C_1, C_2 are the classes. In the case of regression, $\gamma(\boldsymbol{x})$ approximates the target value.

The function $\gamma(\boldsymbol{x})$ is the hypothesis of how the boundary between the two classes looks like in the case of classification, or an approximation of the function itself in the case of regression. The form of $\gamma(x)$ is chosen according to the complexity of the problem at hand. The more complex the problem is, the more flexibility should be given to $\gamma(\boldsymbol{x})$ to learn a good approximation.

The famous forms of $\gamma(\boldsymbol{x})$ are as follows:

1- A linear function of the input, $\gamma(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$ where $\boldsymbol{w}^T$ is the parameter vector, and w_o is a constant bias.

2- A basis function decomposition of the input, $\gamma(\boldsymbol{x}) = \sum_{i=1}^{m} w_i \phi_i(|\boldsymbol{x} - \boldsymbol{\zeta}_i|) + w_o$, where ϕ_i is the basis function, w_i is its weight, and $\boldsymbol{\zeta}_i$ is the center of the basis function

3- A basis function decomposition of a projection of the input $\gamma(\boldsymbol{x}) = \sum_{i=1}^{m} w_i \phi_i(\boldsymbol{\eta}_i^T \boldsymbol{x} - \eta_{oi}) + w_o$, where $\boldsymbol{\eta}_i^T$ is the projection weight for the i_{th} basis, and η_{oi} is a constant.

The last form of representation corresponds to a single hidden layer Feed Forward Neural Network (FFNN) with one output neuron. Each basis function represents a single neuron in the hidden layer. If more layers are to be added, the projection itself will take the form of multiple decompositions. This provides multilayer neural networks with complex representational capabilities (Haykin 2001).

Exemplars of discriminative techniques for time-series analysis that use linear modelling are the autoregression techniques. The most popular of them is the autoregressive (AR) model, and its variants: Autoregressive Integrated Moving Average (ARIMA) (Corduas and Piccolo 2008), and Autoregressive Moving Average (ARMA) (Xiong and Yeung 2004). This type of modelling is very simple, yet it can be too simple to model complex systems. Linear modelling usually suffer from a large bias (Bishop 2006).

The second type of discriminative modelling can be applied to time-series for extracting representations. These representations depend mainly on transforming the time-series into another domain where a fixed number of descriptors/features are formulated. Examples of such are the:

1- Principal Components Analysis (PCA) (Bankó and Abonyi 2012), which uses the eigen vectors of the covariance matrix as the bases functions.

2- Discrete Fourier Transform (DFT) (Rafiei and Mendelzon 2000), which use the exponential harmonics as the bases functions.

3- The Discrete Wavelet Transform (DWT) (Barragan, Fontes, and Embiruçu 2016; Durso and Maharaj 2012), which use a multi resolution set of wavelet bases function.

Both the DFT and the DWT use handcrafted bases, while PCA is dependent on the data. Nevertheless, the PCA assumes that the data is generated from a Gaussian distribution, and the bases are orthogonal to each other. The number of components to use and which type of decomposition to use is very critical to the performance of the system. Moreover, the extracted features cause some loss of information due to the selection of the aforementioned subset which affects the overall performance of the system.

A currently evolving research direction investigates non-linear estimation using Recurrent Neural Networks (RNNs) (Tao Chen et al. 2016; Pacella and Semeraro 2007) which are global function approximators similar to the FFNN, but with feedback connections from the hidden representation back to the input. Using RNNs for time-series representation has advanced a lot the past couple of years due to the increase in computational power and the use of some new approximation models for the RNNs which made RNN training more stable and feasible (Gers 2001). These RNNs showed remarkable success in several applications, but they require a significant amount of data to be well trained. Moreover, they require a significant amount of tuning for architecture in terms of the activation functions, learning rates, learning momentum, optimization technique, number of neurons, and the number of layers.

2.3 Instance-Based Learning

This type of learning is sometimes referred to as “Lazy”, because there is actually no learning to be done. Instead it relies on the definition of a distance measure and a relatively rich database. They are also referred to as non-parametric learners because there is no parameters that are to be learned for this type of modelling. The target function is approximated by a point-wise approximation using the observations available in the database (Bishop 2006). The most popular distance measure for variable length time-series reported by (Fu 2011).

The most critical component of this type of learning is defining a distance measure that is appropriate for the task in hand. If it is chosen carefully, then it gives acceptable results specially in the case of small amounts of data. An improvement to this basic approach is to have a weighted vote among the K-Nearest Neighbors of the test observation using a neighborhood weighting function such as the inverse of the squared distance.

This type of approximation requires a lot of time during the testing phase, and the time requirements grow as the size of the database increases. Yet, there have been several attempts throughout literature to minimize the search time, it is still not practical for large datasets (Murphy 2012).

2.4 Rule Induction

This type of learning is sometimes referred to as non-metric learning (Duda, Hart, and Stork 2000), as there is neither an explicit or implicit similarity measure defined or computed during the training phase.

The output of such techniques is a set of rules that break the observation space into subspaces having the same target function. An example of such is the decision tree (Quinlan 1986), which require for rules by splitting the observation space on one of the features one step at a time starting by a root node. Another example is the LAD technique (E Boros, Hammer, and Ibaraki 2000), which is one of a set of techniques that builds rules by searching for patterns. These patterns are combinations of conditions on the features available in the dataset. The final function approximation is a weighted sum of these patterns.

This type of learning has easily interpretable results, and very easy to implement on dedicated hardware logic circuits. Yet, it is not very suitable for data with relatively high dimensionality (Bishop 2006).

CHAPTER 3 ARTICLE 1: FAILURE TIME PREDICTION USING LOGICAL ANALYSIS OF SURVIVAL CURVES AND MULTIPLE MACHINING SIGNALS OF A NONSTATIONARY PROCESS

Ahmed Elsheikh; Soumaya Yacout; Mohamed Salah Ouali

Submitted to: Journal of Intelligent Manufacturing

Abstract

This paper develops a prognostic technique that is called the Logical Analysis of Survival Curves (LASC). The technique is used for failure time (T) prediction. It combines the reliability information that is obtained from a classical Kaplan-Meier (KM) non-parametric curve, to that obtained from online measurements of multiple sensed signals. The analysis of these signals by the Logical Analysis of Data (LAD), which is a machine learning technique, is performed in order to exploit the instantaneous knowledge that is available about the process under study. LAD represents its decisions in simple Boolean logic that is easy to comprehend and implement on hardware. The proposed LASC can handle predictions for multiple operating conditions simultaneously, and it is robust against propagation of prediction uncertainties. The experimental results of failure times' prediction of cutting tools are reported. The results show that LASC give up to 27% improvements in prediction accuracy than the results obtained by well-known machine learning techniques in terms of the mean absolute error.

Keywords

Failure time prediction, Logical Analysis of Data (LAD), logical analysis of survival curves (LASC), Kaplan-Meier, pattern recognition, machine learning.

3.1 Literature review

Prognostics and health management is one of the most important tools to achieve efficient condition-based maintenance (CBM) (Guillén et al. 2016). While CBM recommends maintenance actions based on the information collected through condition monitoring, prognostics enrich the

CBM by the ability to predict failures before their occurrences so that maintenance actions can be planned in advance. One of the most important aspects of prognosis is the estimation of the Failure Time (*T*) of the working physical assets. Based on the estimated *T*, efficient maintenance actions are planned.

Failure time estimation's approaches are divided into three groups; statistical based and artificial neural networks based, which are data-driven approaches , and physics based approaches (An, Kim, and Choi 2015; Heng, Zhang, et al. 2009; Jardine, Lin, and Banjevic 2006; Si et al. 2011; Sikorska, Hodkiewicz, and Ma 2011). Since in this paper, a data-driven approach is presented, only the previous literature of failure time estimation using data-driven approaches is considered. These approaches gain popularity as the amount of available data increases.

To avoid the assumption of a specific model of degradation, Artificial Neural Networks (ANN) have been introduced to CBM (Sikorska, Hodkiewicz, and Ma 2011). ANNs have many architectures. The one that copes well with temporal data is the recurrent neural network (RNN) (Zemouri, Racoceanu, and Zerhouni 2003). An ANN, whose training targets are system survival probabilities at each time interval, was proposed by (Heng, Tan, et al. 2009). They used feedforward neural network and Kaplan-Meier estimator in order to model a degradation-based failure probability density function. In their paper, the authors predicted the reliability of a pump by using vibration signals. Self-organizing maps, which are unsupervised clustering neural networks, are used to extract signals of degradation. These signals are fed to Feed Forward (FF) ANNs, as proposed by (Huang et al. 2007). Another application of ANNs in CBM, is the use of autoencoder to extract features from rotating machining signals, for visualization and classification using Support Vector Machine (SVM) (Shao et al. 2017).The main problems that face ANNs are: (1) The sensitivity of performance to structure in terms of the number of neurons, number of hidden layers, and connections between layers. (2) The increase of computational requirements and the amount of required training data as the complexity of the ANN increases. (3) The ANN is a black-box model, which means that the model's mathematical representation provides no clear information about the physical phenomenon resulting in the classification results.

Signal decomposition and instance based learning are introduced in the literature in order to use the available data in predicting the failure time. Such decomposition is applied in a predefined set of bases, such as wavelets, where the coefficients can be further reduced using principal components analysis, as proposed by (Baccar and Söffker 2017). A semi-parametric method using subsequences of machine degradation called shapelets is introduced by (Ye and Keogh 2009). The authors built a database of these shapelets and the corresponding remaining useful life for each shapelet. The database is built using K-means, which is the parametric part, and the subsequence matching is done by using Euclidian distance, which is the non-parametric part of the method. This approach requires the choice of the length and the number of shapelets, as well the discriminative threshold. Moreover, the K-means algorithm that is used in this method is a random algorithm that is highly dependent on its initialization.

In order to avoid the above-mentioned limitations of the prediction methods, a failure time prognostic technique is presented in this paper. The well-known non-parametric Kaplan-Meier (KM) curve is used with a machine learning method called the Logical Analysis of Data (LAD). The KM curve is built from the failure times of similar non-repairable assets in order to calculate an aggregate survival estimate $S_{KM}(t)$. Since this curve does not consider the operating conditions that affect the degradation of each individual asset, the Logical Analysis of Data (LAD) exploits the available information about the degradation of each individual asset in order to generate patterns, which characterize subgroups of these assets that have similar degradation profiles. The update of the KM survival curve based on LAD generated patterns, produces individualized and adjusted KM survival curves, which are called the Logical Analysis of Survival Curves (LASC). As the asset degrades in time, the generated patterns indicate specific information about the asset degradation's profile. A similar approach was applied successfully in medicine (Kronek & Reddy 2008). Shaban et al. (2015) presented a limited experimentation with this technique by applying it to machining tool's failure based on only one degradation signal and a single operating condition. This paper extends and generalizes the same approach by using it with multi-state signals, under multiple operating conditions, and by presenting a detailed comparison and analysis of the estimation's accuracy, the prediction's quality that is obtained by the proposed approach, and by other common data-driven techniques. In the next section, the methodology of building the LASC is presented, along with its theoretical background.

3.2 Methodology

This section presents an introduction to the LAD approach, the construction of KM survival curve, and finally, the construction of the LASC, and how it uses the information that is obtained from LAD's generated patterns in order to provide adjusted KM survival curves.

3.2.1 Logical Analysis of Data (LAD)

LAD is a knowledge discovery and data analysis techniques that was introduced by Peter Hammer (Crama, Hammer, and Ibaraki 1988). It is a supervised binary or multi-classes classifier (Mohamad-ali, Yacout, and Lakis 2014). LAD finds causes that discriminates the different classes of a certain phenomenon. For example, the discrimination of classes of faults. The classification is based on patterns, which are sets of data- driven rules that are found in the monitored signals. Whenever these rules are present, then we can understand why the asset is in a certain class of phenomenon. As such, the explanatory power of LAD resides within the generated patterns, and the pattern generation algorithms are the main area of research of this approach (Endre Boros et al. 2011). In general, three methods are used for pattern generation: Enumeration-based techniques, mathematical programming algorithms, and heuristics techniques (E Boros, Hammer, and Ibaraki 2000; Hammer et al. 2004; Ryoo and Jang 2009) . The objective of these techniques and algorithms is to find the minimum number of patterns that characterize all the observations in the dataset, and which form a robust theory that is capable of correctly classifying any new observation. As most machine learning techniques, LAD is applied in two phases; the training phase at which the patterns are generated, and the testing phase at which the patterns' capacity to classify new data is tested.

3.2.2 Kaplan-Meier Survival Curve

The KM curve is a non-parametric method for calculating the survival at time t, $S_{KM}(t)$, based on the observed failures of similar systems. The survival is calculated as follows:

$$S_{KM}(t)=\prod_{ti\leq t}\left[1-\frac{d_{ti}}{Y_{ti}}\right] \qquad 3$$

Where Y_{ti} is the number of tools that are at risk of failure at time ti, and d_{ti} is the number of tools that failed after t_{i-1} and up to time t_i. This equation is used to estimate the base $S_{KM}(t)$, which aggregates all of the failure times observed in the training data without taking into account the status of the monitored signals. Hence, the interest in using KM along with LAD is to take into account the knowledge that is provided by monitored signal’s values. This merge is illustrated in the following section.

3.2.3 Logical Analysis of Survival Curves (LASC)

The proposed technique is inspired by the Logical Analysis of Survival Data (LASD) which was developed and applied successfully in the medical field by (Kronek and Reddy 2008; Reddy n.d.), and the engineering field by (Ragab et al. 2016). The novelty of the proposed technique is the reformulation of LASC to cope with the nature of industrial applications. Specifically, the degradation of physical assets. The proposed approach tracks the evolution of the degradation state, such as wear, throughout the lifetime of the physical asset and produces individualized KM survival curves at each time of analysis according to the sensors’ readings. These are placed at points of interest on the asset. This section illustrates the proposed approach.

LASC is based on the LAD approach that is presented in section 2. The idea is to update the KM curve based on the monitored signals, and by characterizing the degradation of the asset with the notion of events. These events are states of the degradation that are physically recognized by tool wear experts (Shaban et al. 2017). For each time t_k, a positive (or negative) state is defined by whether the asset has reached the predefined event of degradation or not. The measure of degradation, which is used in the experiments in section 3, is the tool wear. The left most part of Figure 3.1 shows the wear state of a subset of the tools versus time, along with the pre-defined wear events. Details about the choice of the wear events are given in (Banjevic et al. 2001).

This means that at each time of analysis t_k, such as the one indicated by the vertical blue dashed line in Figure 3.1, the tools are split into two classes. The positive class contains the tools which pass one of the pre-specified levels of degradation that indicate the wear events on the horizontal red dashed lines in Figure 3.1. The force readings and the corresponding label indicating whether the reading has passed the wear event or not is shown in Table 3.1. The positive class tools are shown in the top-middle of Figure 3.1. The negative class contains the tools that have not passed the pre-specified level of degradation. These are shown in the bottom-middle of Figure 3.1.

The next step is to generate the patterns that characterize each class by using the sensors' readings at the given time of analysis $\mathrm{t_k}$. Finally, a KM survival curve, $\mathrm{S_p(t)}$, is developed for each pattern based only on the failure times of the tools that are covered by this pattern. If the tools of a certain class are covered by more than one pattern, the survival curve is developed for the tools covered by each pattern, and then an average curve is estimated for the whole class.

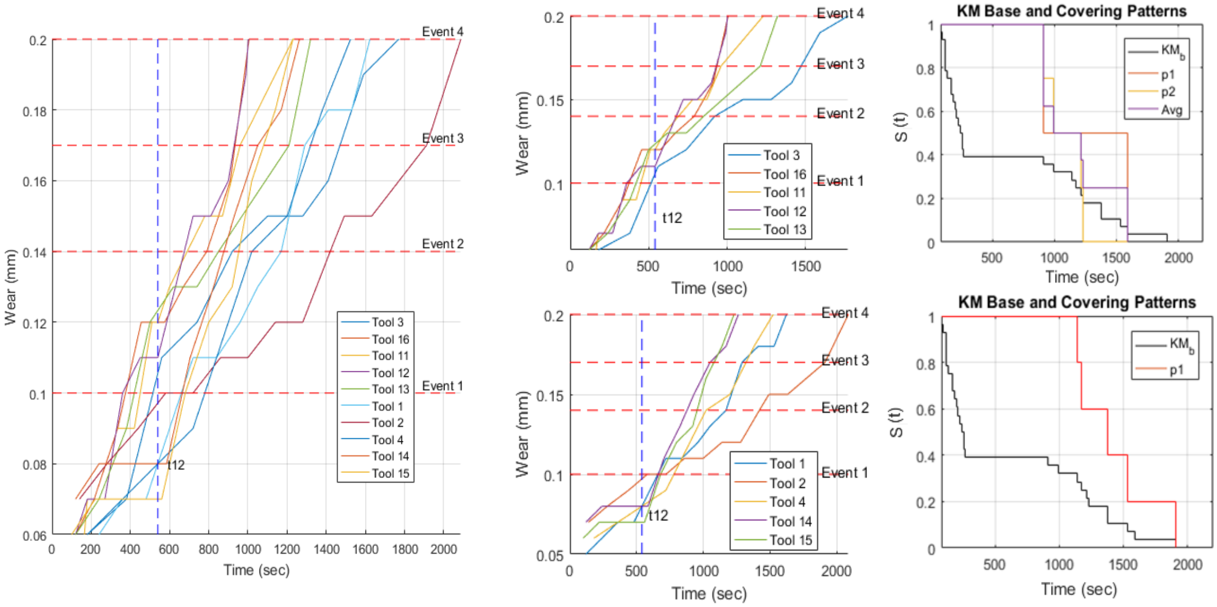


Figure 3.1 Example of analysis at time sample $\mathrm{t_{12}}$ and split using event 1; (mid-top) Tools having wear at time $\mathrm{t_{12}}$ above event 1; (mid-bottom) Tools having wear at time $\mathrm{t_{12}}$ below event 1; (top-right) KM curves constructed from the failure time of the tools covered by patterns p_1, p_2, and their average calculated using equation 4; (bottom-right) KM curve constructed from the tools covered by pattern p_1.

As the tools degrade in time, this procedure splits the original KM curve into KM curves that represent the degradation of subgroups of tools which have similar degradation profile over time. In this case, the failure time's estimation of a tool that is covered by one or more patterns is based on the failure times of specific previous systems that have similar profile of degradation, instead of the failure times of all similar tools. As the time increases, more events are experienced by the degraded tools, and more splitting of the KM curves is performed based on the pattern generated from the sensors' readings. The extracted patterns from the time of analysis $\mathrm{t_{12}}$ shown in Figure 3.1 are given in Table 3.2.

Table 3.1 Force readings and corresponding wear event for the illustrative example

Tool Number	Passed Wear Event	Fx	Fy	Fz
1	FALSE	176.3	75.2	126.75
2	FALSE	157.39	62.186	122.07
3	TRUE	166.67	56.211	94.189
4	FALSE	160.5	60.4	105.7
11	TRUE	217.5	69.7	173.3
12	TRUE	254	81.7	190.7
13	TRUE	220.6	95.333	214.53
14	FALSE	201.54	65.088	183.85
15	FALSE	201.54	65.2	202.28
16	TRUE	258.65	63.879	186.19

Table 3.2 Patterns extracted at t_{12} in the illustrative example

Positive Class		Negative Class	
Pattern	**Covered Tools**	**Pattern**	**Covered Tools**
$\boldsymbol{F_x > 163.585, F_y < 64.4835}$	3, and 16	$F_x < 209.52, F_z > 99.9445$	1, 2, 4, 14, and 15
$\boldsymbol{F_x > 209.52}$	11, 12, 13, and 16		

After the training and the testing phases, a new tool’s failure time is estimated based on specific subsets of tools that have similar degradation profiles as follows:

1. Define the events as in Figure 1 (left).
2. At the end of the kth period, collect the tool's signals and classify the tool.
3. According to the tool's classification result, the tool's failure time is estimated based on the survival curves of the patterns that cover the tool's signals at time t_k.

The following pseudo code summarizes the proposed technique

For the training phase:

1- Define the wear events, $w_i: i = 1, \dots, \omega$, where ω is the number of wear events.

2- Choose the desired times of analyses $t_k: k = 1, \dots, K$, where K is the last time of analysis.

3- At each t_k:

    a. prepare the force readings at that time, and label the tools that had wear above the event w_i by '1', and otherwise by '0'. The wear event chosen is the one that splits the data almost evenly.

    b. Train a LAD classifier that discriminates between the event tools at that time, and the event free tools.

    c. Find the KM survival curve for the tools covered by each pattern using equation 3.

For the testing phase:

1- At a desired time of analysis t_k use the corresponding LAD classifier to identify the patterns that cover that tool given the sensor readings at t_k.

2- Average the survival curves if the tool is covered by more than one pattern as shown in equation 4.

3- Estimate T using equation 5.

In the next section we present the description of an experiment that was conducted in the machining laboratory at Polytechnique Montréal. In this experiment the physical assets are cutting tools. The physical phenomenon of degradation is the tool wear.

3.3 Experimental Application

The objective of this experiment is to estimate the failure times for carbide cutting tools used for machining of titanium metal matrix composite. The monitored signals are the magnitude of the cutting forces, which are acquired by dynamometers. The failure time‘s estimation is updated from the LASC, as described below. The experiment is divided into three main steps; data acquisition, data pre-processing, and data analysis.

3.3.1 Data Acquisition

The experiments are conducted in the machining laboratory at École Polytechnique Montréal. The radial force (F_x), the feed force (F_y) and the cutting force (F_z), are acquired using a 3-component dynamometer which is attached to a 6-axis Boehringer NG 200 CNC turning center. The sensors are built-in the CNC machine, so their locations are not explicitly known and are not user adjustable. The collected signals are passed to a multichannel charge amplifier then collected by national instruments acquisition board (PXI 1000B) (Shaban, Yacout, and Balazinski 2015). The experiments are performed by using a full factorial design, with cutting speed’s values of $40, 60, \text{and } 80 \text{ m/min}$, and feed rate’s values of $0.15, 0.25, \text{and } 0.35 \text{ mm/rev}$. The depth of cut was chosen to be $0.2\ mm$.

The corresponding tool wear was measured by an Olympus SZ-X12 microscope, which measures the flank tool wear at discrete points of time through inspections. This procedure continues until the tool wear reaches a predefined maximum threshold of VBmax=0.2 mm. 28 tools were inspected throughout their lifetime. A snippet of the data is given in Table 3.3.

3.3.2 Data Preprocessing

Since the data is acquired by inspection of the wear, and the wear degradation’s profile is not the same for all the tools, the sampling instants are not exactly the same for all the tools, and they are not perfectly aligned in time. Hence, linear interpolation is used to estimate the missing values between any two successive measurements. Linear interpolation is sufficient for the wear since it is a monotonic non-rapidly changing phenomenon. If other phenomena are to be considered, then other higher order interpolations should be considered. Figure 3.2 illustrates how the wear changes with time for each of the 28 tools.

Table 3.3 A snippet of the data collected in the lab experiments

Working Age	**Wear**	**Speed (m/min)**	**Feed (mm/rev)**	***Fx***	***Fy***	***Fz***
120	*0.05*	40	0.15	120.4	51.1	116.2
240	*0.06*	40	0.15	126	50	109.4
360	*0.07*	40	0.15	140.2	59.2	113.4
480	*0.07*	40	0.15	164.4	70.6	124
600	*0.09*	40	0.15	188.2	79.8	129.5
720	*0.11*	40	0.15	227.4	92.6	136.1
:	*:*	:	:	:	:	:
1410	*0.18*	40	0.15	386.1	136.4	166.7
1530	*0.18*	40	0.15	422.1	150.6	167

3.3.3 Data Analysis

The wear events are define by four levels of degradation as suggested by (Banjevic et al. 2001) and are shown in Figure 3.2. It is the elaboration of the small example in Figure 3.1 to the complete dataset. Each event indicates a transition from one state of wear to another. The wear states are: (1) initial state with wear less than 0.1 mm, (2) regular state with wear between 0.1–0.14 mm, (3) breakage state with wear between 0.14–0.17 mm, (4) severe state with wear between 0.17–0.2 mm, and (5) the worn-out state with wear more than 0.2 mm, which is also considered as failure.

To construct the LASC at each given time of the analysis, LAD is used in order to identify the tools that passed each one of these wear events, and those that did not, as illustrated in Figure 3.1. Then a KM curve is constructed from the failure times of the tools that are covered by each pattern generated.

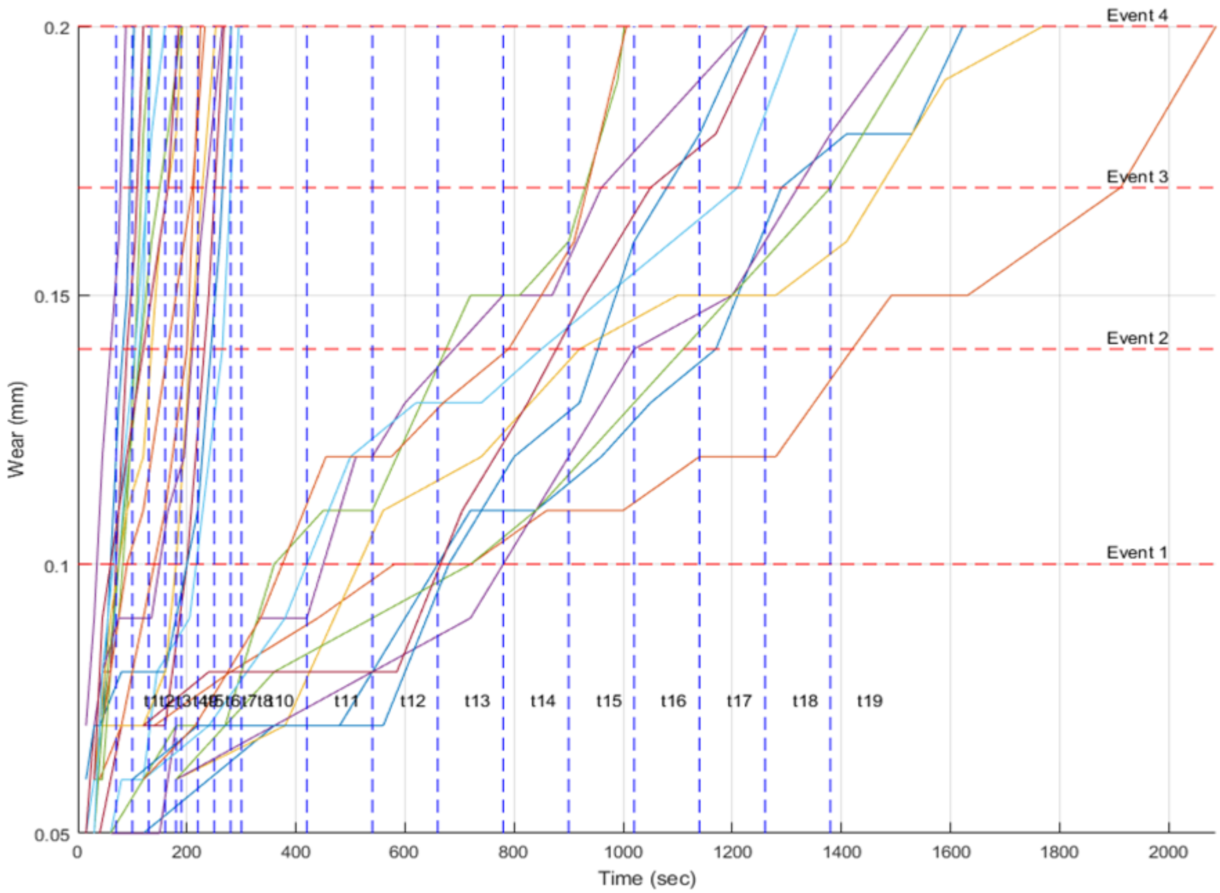


Figure 3.2 Tool wear vs. time showing wear states and analysis times, with linear interpolation between measurements. The variation in failure times is due to different operating conditions.

Whenever there is a tool signals that are covered by more than one pattern, the survival curve of that tool is calculated by using the following equation:

$$S_{Pj\_Avg}(t)=\frac{\sum_{i=1}^{Nc_j(t)} \xi_i(t)*Sp_{ij}(t)}{\sum_{i=1}^{Nc} \xi_i(t)} \tag{4}$$

Where $Sp_{ij}(t)$ is the value on the survival curve of the i_{th} covering pattern for tool j which is under inspection at time t, $\xi_i(t)$ is the weight of the i_{th} pattern at time t. The weight of a pattern is the ratio of covered observations by that pattern, and it represents its importance, the higher the weight the more important the pattern is. $N_{c_j}(t)$ is the number of patterns that covers tool j at time t.

In order to estimate the failure time of a new tool, the force readings at a given inspection time t_k are used in order to identify the current class, and the corresponding patterns that cover these readings. Then the failure time is estimated based on the corresponding KM curve according to the following equations:

$$\text{Expected remaining useful life, } \mathrm{E}(T - t_k | T > t_{k-1}) = \frac{\sum_{t_f > t_k}^{\infty} t_f[S(t_{f-1}) - S(t_f)]}{S(t_{k-1})} - t_k \text{ ,}$$

$$\text{Expected Failure Time, } T = \sum_{t_f > t_k} \frac{t_f[S(t_{f\text{-}1})\text{-}S(t_f)]}{S(t_{f\text{-}1})} \qquad 5$$

Where t_k is the analysis time, t_f is the set of 28 failure times in the training set, f={1, 2,...}, and it is assumed, without loss of generality, that $t_f > t_k$, which means that inspection always precedes a failure event. $S(t)$ is the survival probability obtained from the survival curves of the patterns that cover the classified tools. $S(t)$ in equation 5 can be any of the KM base $S_{KM}(t)$, which is given by equation 3, when estimating *T* by using classical KM and without taking into consideration the acquired signals, or the survival curve for each pattern $S_p(t)$ when the tool is covered by a single pattern, or the average of all the covering patterns $S_{Pj\_Avg}(t)$ when the tool is covered by more than one pattern as in equation 5. *T* is used to estimate the expected remaining useful life (RUL) by subtracting the time of analysis t_k. In this paper, only the calculation of the *T* is used in order to speed up and decrease the calculations of the predictions. The calculation of the RUL is thus straightforward.

3.4 Results

The testing phase of the proposed technique is accomplished by leave-one-out cross validation. This procedure is followed due to the lack of enough data for a single train-test split. Each one of the tools under test is left out once, and predictions of the failure time are compared with the actual failure time. The comparison takes the form of the average of errors for each tool. For each tool the error is calculated as the mean absolute error (MAE) between the actual and estimated times of failure for all the times of analyses. This formulation of this performance metric is given in equations 6 and 7. There are several conditions taken into consideration during the tests:

- The tools that are selected to be used in the testing phase by leave-one-out process, are those that do not have extreme values of *T*, which are either very rapid failure or very long survival. Since such extremes, if removed from the training phase, will result in a biased MAE because the classifier would not have observed a similar case during its training. This does not violate the generality of the approach, since in more practical situations, more data is supposed to be available in order to cover almost all possible scenarios.
- In the training phase, the signals were acquired at different time instants in the laboratory because the wear measurements required pausing the cutting process to measure the wear using a microscope. Hence, at any analysis time, wear values are linearly interpolated if no measurements are available at that time instant.
- Tools, under various operating conditions, namely the speed and the feed, are considered for the analysis. Hence, the tools showed a noticeable variability in their lifetime according to their operating conditions. Tools operating at higher speed and/or feed will have shorter life time. On the other hand, tools under less speed and/or feed will survive more, as depicted in Figure 3.2 where two obvious groups of degradation curves are shown. To cope with such variability in operating conditions, inspections and analyses are conducted at every 30 secs, for the first 300 seconds. Then inspection becomes less frequent at every 120 secs, to avoid redundant calculations for long-life tools, which fail less rapidly.
- There is a period when the short life tools have failed and the long-life tools are still below the first wear event. This is depicted in Figure 3.2 from time 250 sec until 358 sec. This period has no training done within it, since there are no event patterns to be identified, since all the tools during this period either are in their initial wear state, or already failed.
- For the time instances when the classified tools are covered by two or more patterns, the weighted average of the two survival curves is used as shown in equation 5.
- The performance comparison criterion is the mean absolute error (MAE) (Hyndman and Koehler 2006), which is defined as follows

$$\text{MAE}= \frac{\sum_{t_k} |\text{Actual Failure-Estimated Failure}|_{\text{t}=\text{t}_\text{k}}}{\text{number of analysis times}} \qquad 6$$

$$\text{MAE}= \frac{\sum_{\text{j}} \text{MAE}_{\text{j}}}{\text{number of test tools}} \qquad 7$$

Where MAE_j is the MAE for tool number j.

In order to assess the prediction capabilities of LASC, a comparison with other type of machine learning techniques is conducted. The techniques used are kernel methods; namely the Support Vector Regression (SVR), and the Gaussian Process Regression (GPR) because they are the most commonly used in literature (Aye and Heyns 2017; Benkedjouh et al. 2015; Datong et al. 2012; Li et al. 2017). Moreover, they can perform well with limited amounts of data, unlike neural networks which require a significant amount of data to be trained (Sikorska, Hodkiewicz, and Ma 2011). Kernel methods are widely used in the literature because they provide much flexibility for approximating arbitrary functions. However, they show some limitations due to storage and computational complexity (Murphy 2012). The comparison also includes another category of machine learning techniques, which is the non-metric Decision Tree Regression (DTR), and the Ensemble Tree Regression (ETR). The MAE results are summarized in Table 3.4.

The inputs to any of the above regressors is chosen to be w_n force readings, where w_n is termed the window size. A window size of 2 corresponds to taking the current force readings along with the previous reading. Similarly, a window size of 3 corresponds to taking the current force readings along with 2 previous readings, and so forth. The output targets of the regressors corresponds to the estimated remaining useful life after the current reading.

3.5 Discussion

From the results in Table 3.4, it is evident that LASC's performance is comparable to other machine learning techniques. The relative decrease in MAE when using LASC over other technique is calculated by $(MAE_{other} - MAE_{LASC})/MAE_{other}$. When using a window size of 3 is used for the other techniques, LASC is on the average 0.6% higher. On the other hand, the LASC' results outperform the other techniques when they use a window of size 2 with an average of 16.5% improvement. One of the advantages of the LASC technique is that it requires no windowing process to estimate the RUL. This means that the predictions are available immediately online once the machining signals are available.

Table 3.4 MAE using leave-one-out cross-validation for different RUL predictors.

Validation Tool no.	LASC	SVR	GPR	DTR	ETR	SVR	GPR	DTR	ETR
		Window size = 3				Window size = 2			
1	295.1	361.6	296.3	200.6	302.4	407.0	245.2	479.8	347.3
2	597.6	497.9	324.0	432.9	409.6	555.6	503.3	485.5	488.8
3	222.3	305.5	383.8	423.8	274.7	379.7	290.8	472.8	314.3
5	135.7	227.4	247.7	242.1	153.4	296.5	254.4	383.9	175.7
6	409.0	231.9	149.4	53.44	490.5	230.8	210.7	75.04	216.6
10	576.9	306.4	311.9	480.0	531.5	300.9	309.2	383.7	556.5
11	259.9	484.5	514.4	392.5	400.4	488.0	524.8	476.7	458.1
12	245.1	180	120.7	158.6	152.0	208.3	178.6	187.7	156.6
13	147.0	216.8	131.8	175.2	164.8	263.6	172.6	143.8	208.4
14	186.9	213.9	58.9	145.5	81.1	256.0	134.3	99.3	119.5
15	229.0	211.0	91.2	177.2	116.5	252.0	82.8	101.0	116.7
16	288.7	141.0	44.5	173.7	51.9	171.6	98.8	128.7	72.5
17	293.4	333.3	138.2	60.4	140.1	337.7	201.4	170.6	223.0
19	24.37	333.3	205.4	197.4	245.6	337.7	211.7	387.0	295.2
22	30.0	344.0	63.0	11.1	149.6	348.5	212.0	14.1	237.9
26	19.9	313.2	313.6	29.7	208.6	311.2	302.6	605.3	307.5
27	25.0	326.0	401.9	61.7	258.2	323.8	376.6	249.9	328.3
Average MAE	234.5	295.7	223.3	200.9	243.0	321.7	253.5	285.0	271.9
Relative decrease in MAE using LASC		0.2067	-0.05	-0.167	0.035	0.271	0.075	0.177	0.138

Other merits of LASC method are as follows:

- LASC provides prediction that is based on simple human-readable patterns. These patterns are interpretable because they explain why the LASC considers the current state as weary or not. They provide valuable knowledge to practitioners, such as management or maintenance personnel, in order to take evident-based decisions, and to decide on the suitable actions, before failure. Moreover, they are compact and more meaningful than those extracted by decision trees (Johannes, Dragan, and Nada 2013). This can be seen in the illustrative example patterns given in Table 3.2. The patterns covering the positive patterns do not share the same root threshold on F_x. Therefore, to model $F_x > 163.585, F_y < 64.4835$ and $F_x > 209.52$ using a decision tree, there will be more branches and conditions (Rokach and Maimon 2008).
- The state of the metal matrix turning operation is defined by the forces' readings by the dynamometers and the wear readings by using a microscope. Uncertainty is evident either due to electronic noise in the sensors, or the human readings of the wear. Nevertheless, the proposed LASC method has a survival curve, which is dependent on the current state of the tool, and estimates T at each analysis time, in a way that is completely independent from all the precedent times. This protects the proposed method from propagation of state estimation errors.
- LASC considers multiple operating conditions. The operating conditions used in data collection in terms of the speed and feed of the cutting tool are those that are common for turning of Titanium composite materials. If new data is gathered under new operating conditions, LASC searches for the patterns, and not the conditions, that cover this data, and estimate the time to failure based on these patterns. This means that even under different operating conditions, LASC can still estimate the failure time by considering the other factors.
- The proposed method does not assume any specific form of degradation or model for the time to failure T. Instead, the method uses a frequentist approach. This approach is used in Instance Based Learning (IBL), and it is based on observed data only. According to (Javed, Gouriveau, and Zerhouni 2017), IBL is more robust to uncertainties because there is no model assumed. Also, according to (Endre Boros et al. 2011), pure patterns generated from LAD are robust to noise when trained on enough data.

3.6 Using LASC for Decision Making

The predictions given by LASC are calculated for each tool after each data collection and analysis time, t_k. Figure 3.3 shows that these predictions improve as the time of the analysis comes closer to the actual failure time since the MAE is decreasing. Hence, in this section, we search for the best time t_{kop}, in order to take an action of changing the tool. This decision is based on the predictions' quality represented by the value of the predicted RUL. We search for a threshold RUL_{th}, below which the prediction of RUL is considered to be of better quality, thus a higher confidence can be given to the estimate.

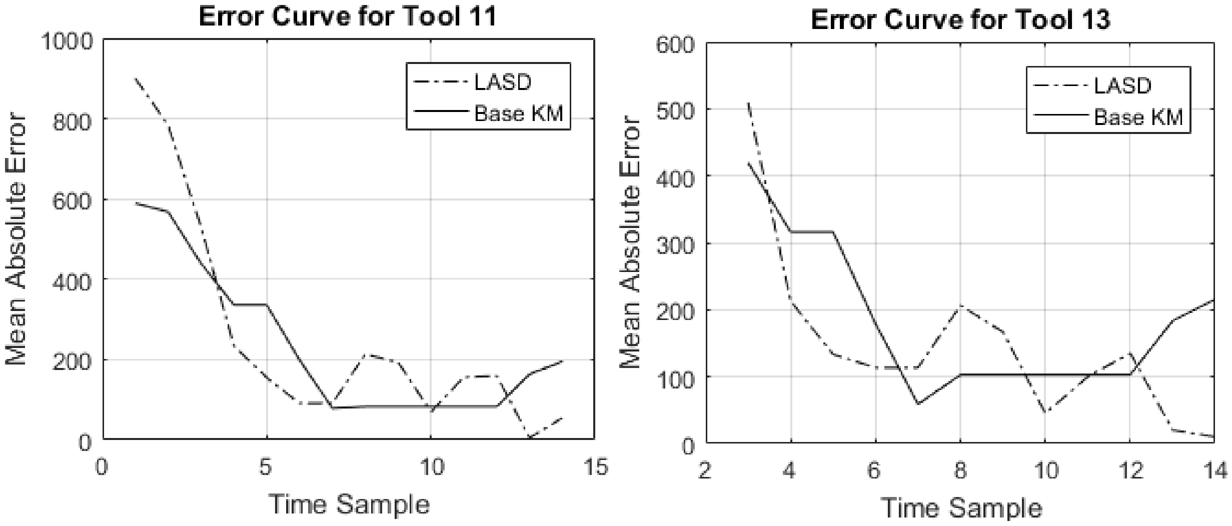


Figure 3.3 LASC and KM MAE performance vs time for tools of the classified tools

To find this threshold, a balance between three competing predictions' qualities is sought, namely;
(1) The percentage of late predictions, which is the percentage of tools in the training set that have RUL below the RUL_{th}, but their predicted failure times come after the actual failure times.
(2) The percentage of missed out predictions, which is the percentage of the tools in the training set whose RUL values are always above the RUL_{th} throughout their lifetime, and therefore no prediction is considered to be of quality until the tool failed.
(3) the MAE , which reflects the deviation from the actual failure time. These quality factors are formulated in equations 8 through 10.

For the data used in this experiment, a compromise between the three prediction quality factors is sought. The quality of the prediction is based upon the predicted RUL, which is obtained by equation 5. The threshold is determined by balancing the three competing predictions' qualities; the percentage of late predictions, the number of missed out predictions, and the MAE. A graphical illustration of the relation between decreasing the percentage of late predictions, as well as the number of missed out predictions, and the resulting MAE for different RUL thresholds is shown in Figure 3.4. The MAE is normalized to the maximum value so as to stay in the same range of percentage as the other factors. The maximum calculated MAE for the set of tested tools is 252.24 sec. The formulas used for each of the factors shown is as follows:

$$\text{Late predictions} = \frac{\text{Number of tools having RUL below } \text{RUL}_{\text{th}} \text{ and estimated T after the Actual T}}{\text{Total number of tested tools}} \% \qquad 8$$

$$\text{Missed out} = \frac{\text{Number of tools with RUL higher than threshold until T}}{\text{Total number of tested tools}} \% \qquad 9$$

$$\text{Normalized MAE} = \frac{\text{MAE of validated tools}}{\text{Maximum MAE among all tools}} \% \qquad 10$$

As shown in Figure 3.4, as the RUL_{th}increases, the number of missed predictions decrease while the number of late predictions, and the average MAE increase. On the contrary, if the predictions of better quality are taken closer to the actual T, which means choosing a smaller RUL_{th}, the number of missed predictions and the hazard of late predictions increase, while the average MAE decrease. We assume that all these factors are equally important. When a larger dataset is given, a weighting of these factors might be considered and estimated.

The summation of all qualities is shown at the top of Figure 3.4 in blue. The behavior of the total compromise curve starts off with a low value due to the low MAE, but this is not a reliable choice due to the large amount of missed out predictions. As a result of the difference in the rates of change of the increasing MAE and late predictions versus the decreasing rate of the missed out predictions, a hump in the total quality measure is formed from 140 sec to 220 sec. An almost flat region is achieved between 220 sec and 240 sec, where the 220 sec is chosen since it has a lower MAE. This is marked by a black cross in Figure 3.4. Hence, whenever the RUL of a tool under inspection is found to be less than 220 secs, it is considered as a reliable estimate and a decision of changing the tool should be taken accordingly.

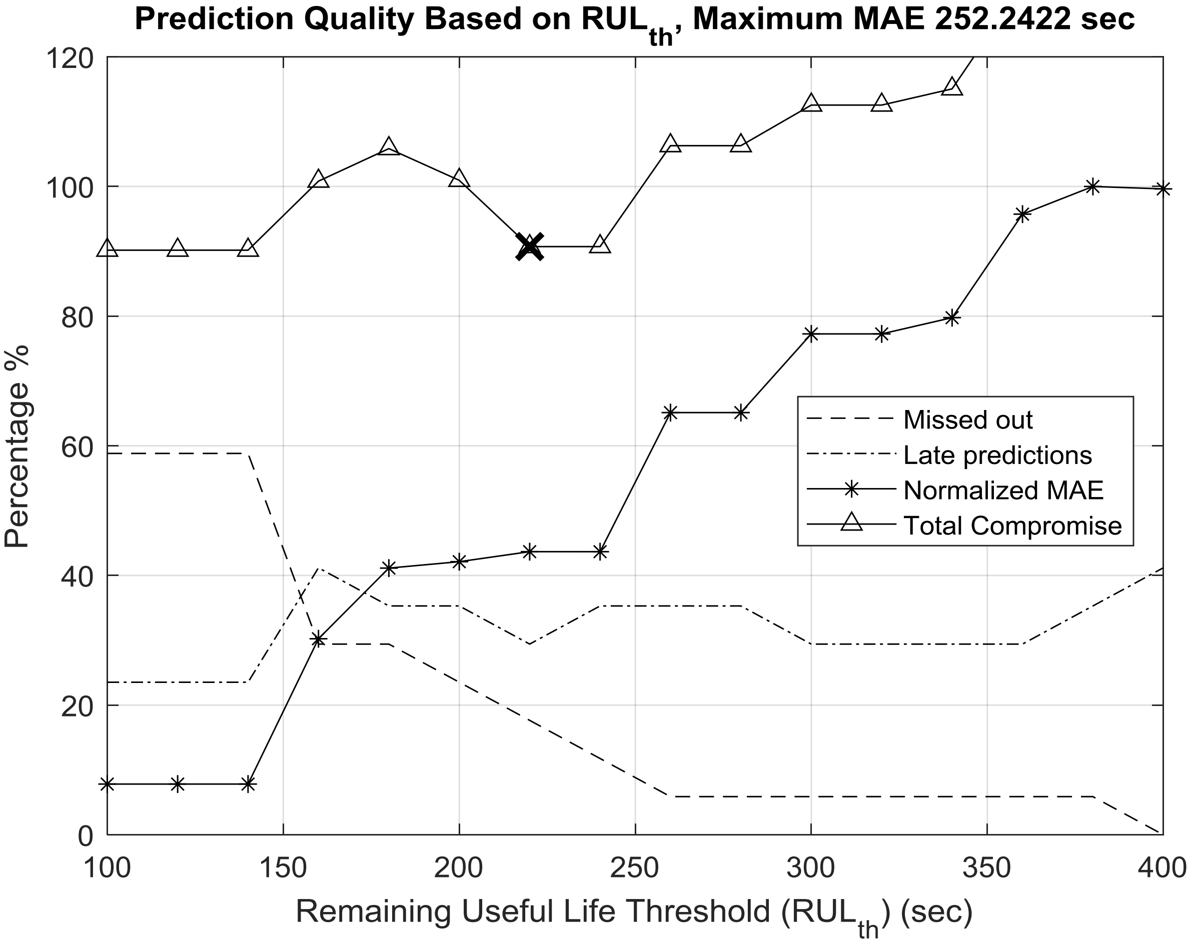


Figure 3.4 Compromise between the three decision quality factors to find the reliable decision based upon the choice of the RUL$_{th}$

3.7 Conclusion

In this paper, logical analysis of survival curves (LASC) is proposed to perform failure time estimations for carbide tools during turning of metal matrix composites. The working tool is monitored by dynamometers to measure the applied axial forces. The tools operate under various conditions in terms of the speed and feed. LASC merges a logical classifier, Logical Analysis of Data (LAD), which produces interpretable patterns that can explain to the user the causes of a certain state of the system, with the classical Kaplan-Meier (KM) survival analysis. LAD classifies the working tools at any given time of analysis into two classes according to the measured forces.

The two classes indicate whether the tool has reached one of the pre-defined states of wear or not. KM is then used to provide the survival analysis of each class. Hence, LASC allows the production of adapted KM curves that consider the measured signals instead of having a single aggregate KM curve throughout the analyses. A prediction quality assessment strategy is discussed to provide a guideline for decision making. A compromise is reached between making late predictions, missed-out predictions, and the accuracy of the predictions assessed by the MAE.

LASC can be applied to any application that has an evolving phenomenon and it is desirable to estimate when a test observation will reach a certain magnitude threshold.

Table 3.5 Summary of Potentials and Limitations of LASC

Proposed Technique	Logical Analysis of Survival Curves
Rational, Need	Predicting when a time-series will reach a certain threshold based on historical data
Potential in industry	Better scheduling for replacement maintenance
Benchmarks used	Remaining Useful Life for carbide cutting tools used for machining of titanium metal matrix composite
Average performance improvement against benchmark	27% decrease in the Mean Absolute Error of prediction
Scalability of the algorithm	Can scale up to 10,000 observations per analysis time, and number of indicators up to 20 sensor readings
Computational Complexity	Requires d comparison operations, 1 summation, and 1 division operation per prediction, where d is the number of sensors.
Extendibility to other potential applications	Can be used for any type of time-series data that show a certain trend and the time to reach a certain threshold value is of interest
Inputs to the algorithm	Multivariate time-series data labeled with the desired threshold value on its magnitude
Output from the algorithm	Prediction of the time remaining until the time-series reaches in a certain magnitude threshold
Parameters to set / decided upon	*The number of intermediate events which can either be known apriori, or simply the mean at each time of analysis. *The times of analysis which can be chosen to be at every time data is available, or according to a certain tolerance in the rate of predictions required.

CHAPTER 4 ARTICLE 2: BIDIRECTIONAL HANDSHAKING LSTM FOR REMAINING USEFUL LIFE PREDICTION

Ahmed Elsheikh; Soumaya Yacout; Mohamed Salah Ouali

Submitted to: Neurocomputing Journal

Abstract

Unpredictable failures and unscheduled maintenance of physical systems increases production resources, produces more harmful waste for the environment, and increases system life cycle costs. Efficient Remaining Useful Life (RUL) estimation can alleviate such an issue. The RUL is predicted by making use of the data collected from several types of sensors that continuously record different indicators about a working asset, such as vibration intensity or exerted pressure. This type of continuous monitoring data is sequential in time, as it is collected at a certain rate from the sensors during the asset's work. Long Short-Term Memory (LSTM) neural network models have been demonstrated to be efficient throughout the literature when dealing with sequential data because of their ability to retain a lot of information over time about previous states of the system. This paper proposes using a new LSTM architecture for predicting the RUL when given short sequences of monitored observations with random initial wear. By using LSTM, this paper proposes a new objective function that is suitable for the RUL estimation problem, as well as a new target generation approach for training LSTM networks, which requires making lesser assumptions about the actual degradation of the system. The average performance improvement shows 18.75% decrease in the mean absolute error of prediction on the NASA turbo engine dataset.

Keywords: Remaining useful life prediction, bidirectional handshaking, long short-term memory, asymmetric objective function, target generation.

4.1 Introduction

Remaining Useful Life (RUL) prediction is the attempt to predict the remaining period of normal operation, at a certain level of performance, for a physical system (Sikorska, Hodkiewicz, and Ma 2011). There has been numerous research on RUL production in the literature. Nevertheless, there is no one approach that is universal because of the variability in the physics of different systems, their surrounding conditions, initial working conditions, and the physics of the acquisition devices (Javed, Gouriveau, and Zerhouni 2017), [4](Hess, Calvello, and Frith 2005; Venkatasubramanian 2005). The initial condition of the working asset or its subcomponents affect the asset's RUL. Poor initial conditions put the asset at a higher risk of earlier failure, which in turn means shorter RUL (Hess, Calvello, and Frith 2005).

Previous research can be mainly classified into physical modelling approaches and data-driven approaches (An, Kim, and Choi 2015; Javed, Gouriveau, and Zerhouni 2017). Physical modelling achieves the best performance if it can be accurately formulated. This is an almost impossible task as a result of the large amount of interacting variables that can affect the physical system directly and indirectly, which are mostly unknown (Sikorska, Hodkiewicz, and Ma 2011). Moreover, these variables interact with each other in highly complex, non-linear ways that cannot be anticipated (Javed, Gouriveau, and Zerhouni 2017).

Accurate RUL prediction allows users of a physical system to benefit the most from its working lifetime, and to make efficient decisions regarding maintenance and replacement (Si et al. 2011). In turn, this means more profit and fewer losses due to unexpected faults or failures (An, Kim, and Choi 2015). Due to the random nature of the system behavior, there is always the possibility that the predicted RUL will either be earlier or later than the actual lifetime. In this case, it is safer to have earlier rather than later predictions to avoid catastrophic failures (Saxena et al. 2010).

Data-driven approaches offer good approximations for a system's failure mechanism based on historical data. These approaches vary in their capacity to learn complex systems (Sikorska, Hodkiewicz, and Ma 2011). The monitored signals captured by sensors are acquired at a certain rate and in a chronological order from the working system. Therefore, processing sequential data such as this and continuously predicting the RUL is a problem that is suitable for sequential modelling. Data-driven approaches handle sequential data differently; either intrinsically, such as Hidden Markov Models (HMM) (Rabiner 1989), by windowing such as Convolutional Neural Networks (LeCun and Bengio 1995) and most machine learning techniques, or by transformations to compress variable length sequences in a set of informative features (Fu 2011). The latter two approaches do not harness much of the sequential information as a result of their limited scope of observation in time and compression of information. On the other hand, models that are sequential in nature capture sequential dependencies between various observations in time in a more compact manner.

One of the most common sequential modelling techniques is the Recurrent Neural Networks (RNN) (Haykin 2001). The main advantage of RNNs over HMMs is that HMMs have a finite, discrete set of states to represent the system, while RNN theoretically has no such limitations (Gers, Schraudolph, and Schmidhuber 2002). RNN and its more recent variations such as Long Short-Term Memory (LSTM) networks (Hochreiter and Urgen Schmidhuber 1997) and the Gated Recurrent Unit networks (Cho et al. 2014) have shown some success in various domains that have a sequential nature (Greff et al. 2017), or that can be processed sequentially (Le, Jaitly, and Hinton 2015).

There is very limited work in the literature using RNN and its variants for RUL prediction (Sikorska, Hodkiewicz, and Ma 2011), and even fewer publications that actually report performance and scores on real test scenarios. Very recent publications, as in (Wu et al. 2017), use LSTM and discuss how the data is prepared for the RUL prediction task, but they use a subset of the training set as the test data, which means they know the full testing sequence to allow it to produce detailed results about the performance of the LSTM over time. Nevertheless, they did not report or prepare the network to work with short sequences of sensor observations. Also, another very recent attempt was the use of an ensemble of Echo State Networks (ESN) (Rigamonti et al. 2017). However, none of the previous publications report results using bidirectional RNNs, that were first introduced by (Schuster and Paliwal 1997), which process input sequences in both directions. This architecture has shown improvements in different recognition applications that are sequential in nature (Graves and Jaitly 2014). Yet, this type of architecture requires certain modifications to suit the RUL prediction problem, which is different than recognition.

All of the aforementioned sequential modelling techniques are supervised and therefore require training on targets. Since the RUL cannot be assumed to be degrading linearly at every working cycle, especially if the working asset starts at a new condition, some papers have made assumptions about the nature of the actual RUL by using the piece-wise degradation function, which starts constant and degrades according to a certain power function (Heimes 2008; Wu et al. 2017). The point of inflection from constant to degradation is determined by inspection (Heimes 2008) or by tuning a detector model (Wu et al. 2017). Such assumptions restrict the model to making predictions given a full history of the working asset until failure. This implies that partial maintenance procedures causing the asset to start at an improved health condition are not taken into account, and such maintenance actions induce only minimal repair.

This paper presents four main contributions in order to predict the RUL:

1- Predicting the RUL from short observation sequences with random starts, since the initial condition of physical systems is usually unknown due to manufacturing deficiencies, replacements of parts of the system, and non-ideal maintenance. Hence, the network is trained to anticipate such requirements.

2- Proposing a new safety-oriented objective function for the LSTM network to train the network to favor safer, earlier prediction rather than later prediction, since all well-known objective functions are either symmetric, such as the Mean Squared Error (MSE) and Mean Absolute Error (MAE), or rectified such as hinge loss (Crammer and Singer 2001), and both types are not suitable for making safe predictions.

3- Proposing a new target RUL generation procedure for the training process of the network. Instead of relying on the manual examination of the data (Heimes 2008) or tuning other models (Wu et al. 2017), the proposed procedure for generating predicted RUL uses the given sensor readings for defining an approximation for the actual RUL.

4- Proposing a new bidirectional LSTM network architecture that suits the RUL prediction problem. Since there are no intermediate predictions required for a given sequence of observations, and only the RUL needs to be predicted after the given sequence is observed, then there is no need to process the sequence simultaneously in both directions.

Instead, the proposed architecture processes the observed sequence in both directions sequentially by processing the sequence in the forward direction, and then using the LSTM final states to initialize the backward processing cells. This architecture forces the network to obtain two different yet linked mappings of the observation sequences to the desired RUL. The reason is that each cell is a function of the current input and the previous state, and both are different for LSTMs processing in opposite directions, unlike all-forward LSTM architectures, where all LSTMs have the same input sequence but different previous states.

4.2 The Long Short-Term Memory Cell

An LSTM cell was proposed to overcome the limitations of training the classical RNN (Hochreiter and Urgen Schmidhuber 1997). Instead of having the output of the RNN cell be a non-linear function of the weighted sum of the current inputs and previous output, the LSTM uses storage elements to pass information from the past outputs to current outputs.

The LSTM has three control signals, such that each is a non-linear function activated by a weighted sum of the current input observation x_t and previous hidden state h_{t-1} as shown in equations 11-16. The forget gate f_t decides whether to retain or forget the previous state c_{t-1} of the LSTM. The input gate i_t decides whether to update the state of the LSTM using the current input or not, and the output gate o_t decides whether to pass on the hidden state h_t to the next iteration or not. The new state c_t stored in the LSTM is the sum of the new gated input a_t and the gated previous state c_{t-1} as shown in equation 15. Figure 4.1 illustrates how the gates and inputs interact.

$$i_t = \sigma(W_i x_t + H_i h_{t-1} + b_i) \qquad 11$$

$$o_t = \sigma(W_o x_t + H_o h_{t-1} + b_o) \qquad 12$$

$$f_t = \sigma\left(W_f x_t + H_f h_{t-1} + b_f\right) \qquad 13$$

$$a_t = \tanh(W_a x_t + H_a h_{t-1} + b_a) \qquad 14$$

$$c_t = f_t \odot c_{t-1} + i_t \odot a_t \qquad 15$$

$$h_t = o_t \odot \tanh(c_t) \qquad 16$$

Where W_*, H_*, and b_* are the trainable weights and biases, respectively, for each gating signal indicated by $*$, h_{t-1} is the hidden layer activation of the previous iteration, while h_t is the current hidden layer activation. Similar to the hidden states, the cell states c_t, c_{t-1} are defined. The current input is x_t, and the gate activations are f_t, a_t, i_t as described previously. Finally, $\odot$ is the elementwise multiplication operator.

The training of the LSTM using backpropagation is much more stable than the classical RNN and can theoretically retain information for prolonged periods of time (Hochreiter and Urgen

Schmidhuber 1997). This is beneficial when attempting to make forecasts about the future by learning from long sequences of historical data.

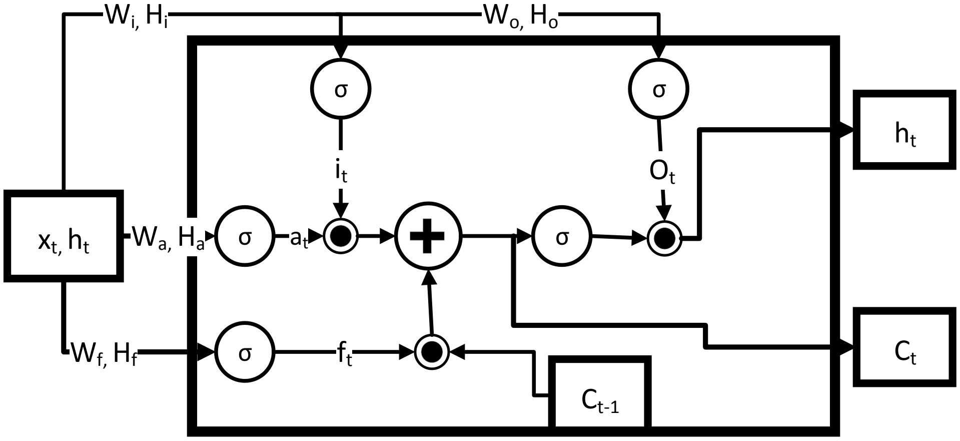


Figure 4.1 LSTM cell diagram

4.2.1 Bidirectional LSTM

Bidirectional RNNs are a modification to the conventional RNNs to process sequences of observations in both directions, starting from the first input observation to the last, and starting from the last observation back to the first (Schuster and Paliwal 1997). This requires the processed sequence to be buffered into windows of observations to allow processing in both directions. At any point in time, the network makes use of the earlier observations processed by the forward LSTM cells until the point of prediction, as well as the upcoming observations processed by the backward LSTM cells to make the prediction. This approach is beneficial when making intermediate predictions, like recognizing phonemes in speech recognition (Graves and Jaitly 2014). This is different from the prediction task, where predictions are required ahead of the whole given sequence. This requirement is the trigger for the new proposed architecture for bidirectional LSTMs.

4.3 The Proposed Methodology

4.3.1 Bidirectional Handshaking LSTM (BHSLSTM)

The aim of the proposed approach is to extract as much information as possible from a given subsequence of observations to make predictions about the RUL of the system. Processing the given sequence in the forward direction, which means in the same sequence as they appeared in the system through several LSTM cells, produces a summary vector, which contains the final output after passing the given sequence through the cells.

The proposed handshaking approach initializes another set of LSTM cells that process the given sequence in reverse order, starting with the last observation and ending up with another summary vector at the first observation. This allows the LSTM network to have more insights when identifying the trend of the sequence in both directions. Moreover, the handshaking procedure allows the learning process to be collaborative between the forward and backward units, and therefore provides better results.

Figure 4.2 illustrates the architecture when using a single LSTM cell in the forward direction and another for the backward direction. The figure shows how the final state of the forward processing cell initializes the state of the backward processing cell.

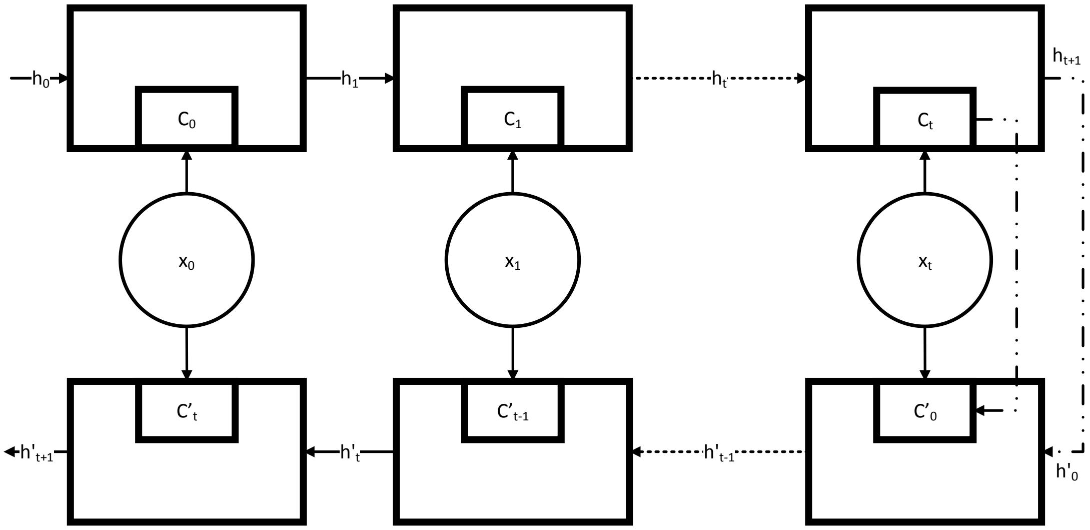


Figure 4.2 BHLSTM diagram for a single forward and a single backward LSTM cells

4.3.2 Safety-Oriented Objective Function

The objective function is the error function at the output of the network that needs to be minimized. The networks are trained by minimizing the error between the network predictions and the actual RUL in the training data. As mentioned earlier, the main goal of RUL estimation is to anticipate upcoming failures before they occur to avoid unnecessary downtime, as well as additional maintenance costs. That is why the Scoring Function (SF) proposed by (Saxena, Ieee, et al. 2008) favors early estimations rather than late ones, as shown in the following equation, by assuming that $d = RUL_{pred} - RUL_{actual}$:

$$SF = \begin{cases} e^{-\frac{d}{\alpha_1}}, & d < 0 \\ e^{-\frac{d}{\alpha_2}}, & d \geq 0 \end{cases}, \quad \alpha_1, \alpha_2 \in \mathbb{R}^+ \qquad 17$$

In order to comply with the requirements of the scoring function, and in general any safety measure for prediction, the network is trained with an objective function that penalizes unsafe predictions at a much higher cost. The SF is not a suitable objective function, as raising the error measure to an exponential is not a well chain-rule-differentiable function because the exponential term must be calculated for every weight update, which makes it computationally expensive. It also can either overflow, resulting in very large updates, or vanish, resulting in no updates at all due to the multiple applications of the exponential term.

Therefore, in this paper, we propose using an approximation of this function as shown in Figure 4.3 which also favors earlier predictions rather than later ones. This approximation is more suitable unlike conventional objective functions, which are either symmetric, such as the MSE and MAE functions, or rectified, which means that they are one-side, favoring those such as the hinge functions. When $d = RUL_{pred} - RUL_{actual}$, the proposed asymmetric objective functions are defined as follows, the Asymmetric Squared Error (ASE)

$$ASE = \begin{cases} \alpha_1 d^2, & d < 0 \\ \alpha_2 d^2, & d \geq 0 \end{cases}, \quad \alpha_1, \alpha_2 \in \mathbb{R}^+ \qquad 18$$

And the Asymmetric Absolute Error (AAE)

$$AAE = \begin{cases} \alpha_1 |d|, & d < 0 \\ \alpha_2 |d|, & d \geq 0 \end{cases}, \quad \alpha_1, \alpha_2 \in \mathbb{R}^+ \qquad 19$$

4.3.3 Target RUL Generation

The target RUL used for training the network is another challenge, since the actual state of health of a system is not given and is usually unknown. Hence, different suggestions about the nature of degradation were given by different authors (Heimes 2008; Wu et al. 2017).

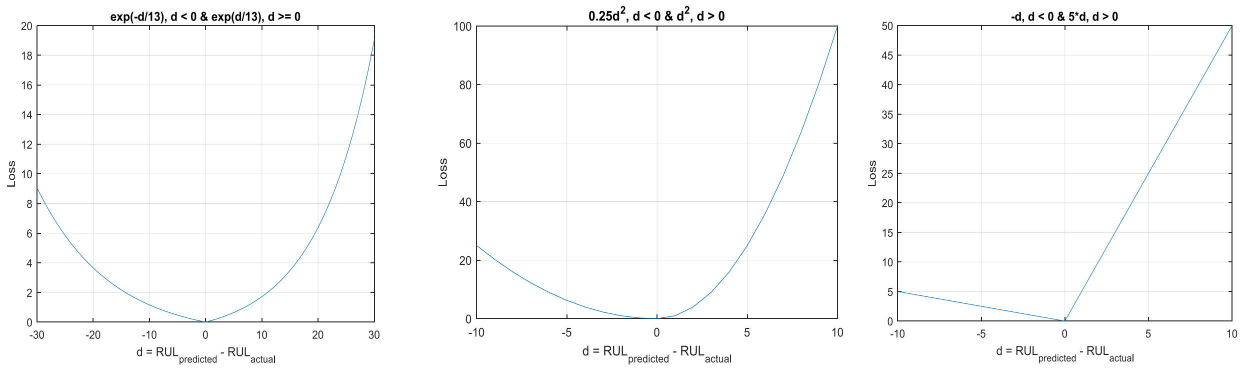


Figure 4.3 Comparison between the scoring function $\alpha_1 = 10, \alpha_2 = 13$ (left), the proposed asymmetric squared objective function $\alpha_1 = 0.25, \alpha_2 = 1$ (middle), and the asymmetric absolute objective function $\alpha_1 = 1, \alpha_2 = 5$ (right).

The core idea for their assumptions is that for healthy systems, the degradation is not noticeable, and all system behaviors look alike, making the RUL estimation an impractical task. Therefore, it is assumed that the RUL will be piecewise continuous, such that at the beginning of life, the RUL is constant and then starts decreasing.

The rate of decrease and the point where the system starts degrading are the two main concerns. The author in (Heimes 2008) made an assumption based upon a manual study of the data that the system will start degrading at 130 cycles and will degrade linearly afterwards. The authors of (Wu et al. 2017) suggested using an anomaly detector to identify the point where the system starts degrading, which signals the start of degradation after three triggers, and they tried several power law degradation functions. They had to tune the anomaly detector, the number of triggers, and the power of the degradation function parameters using a grid search until they achieved good validation results.

Our proposed target preparation is based on the behavior of the sensors' readings of the system. This is a real physical phenomenon that the previous researchers tried to approximate. We start the preparation of the target by taking the moving average smoothed version of all the sensor readings that show a trend, scale them down, and shift their values to start from 1 and end at 0. The sensor readings that show an upward trend are inverted before this step by subtracting all of the sensor readings from the maximum value. Now, each sensor reading has a value ranging from 1 to 0, which is assumed to represent the ratio of the current health condition relative to the starting health condition at 1 and failure at 0. The lifespan of an asset is the working time, starting from a healthy condition until failure in a run-to-failure dataset. Hence, the RUL at any instant of the working time is a fraction of the lifespan, which is proportional to the health condition at that instant. To get the values to correspond with the remaining useful life, each of these ratios are then multiplied by the lifespan, which is the number of life cycles that the working asset took before failure starting from a healthy condition, as given in the training data. Next, for each sensor, there is the estimation of the actual RUL. The final RUL is chosen from the minimum RUL among all of the estimations to favor safer early predictions.

Finally, all RUL values above the minimum lifespan in the training set are truncated to that value. This truncation was shown to be useful in several previous works (Heimes 2008; Riad, Elminir, and Elattar 2010; Wang et al. 2008). Figure 4.4 illustrates this procedure for one of the sensors' readings.

In summary, the point at which the system starts degrading and the degradation profile are dependent on the sensor readings and requires no tuning or any assumptions about the function of the degradation.

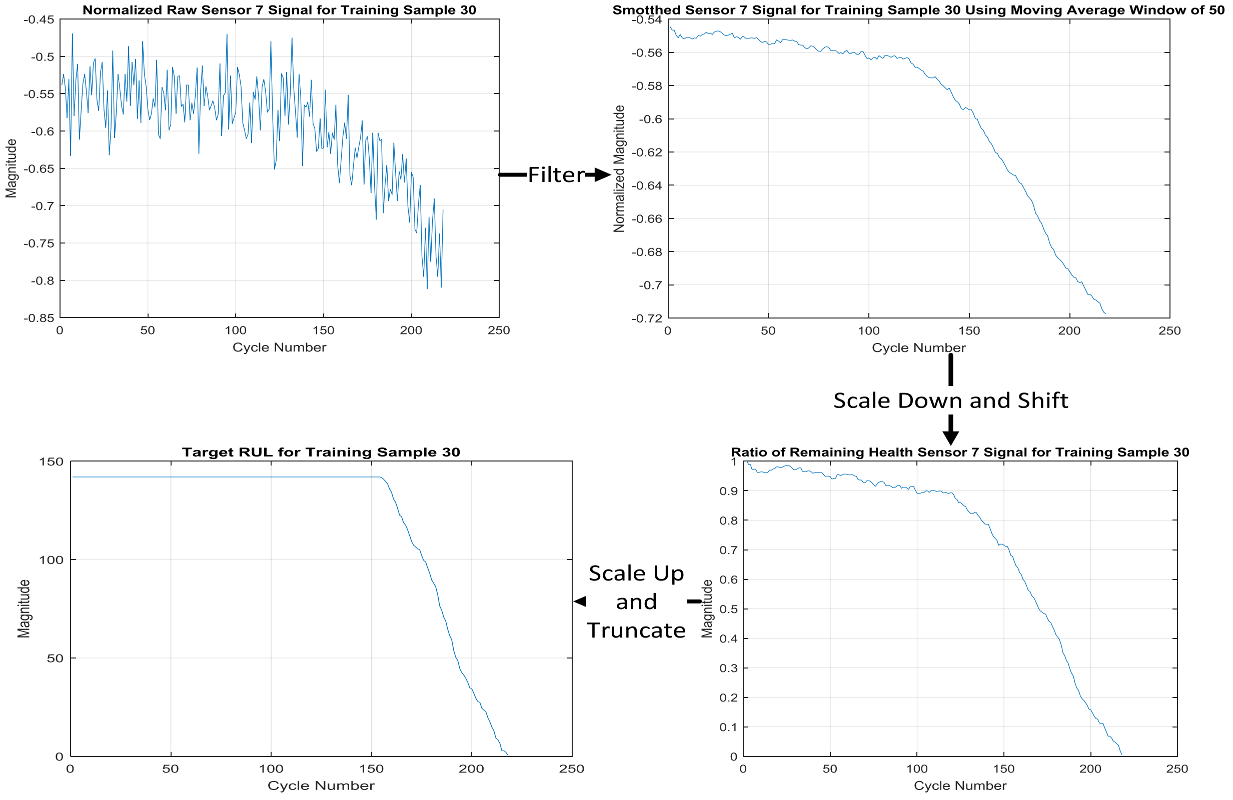


Figure 4.4 RUL target generation procedure.

4.3.4 Data Preparation

Since the network should be prepared to make predictions given a short sequence of observations, the training data must accommodate such requirements. That is why the training sequences are chunked into consecutive overlapping windows of observation sequences, each with a length equal to the minimum anticipated input sequence, which corresponds to the maximum desired number of cycles before the first RUL prediction is given. Each of these input sequences has an output corresponding to the RUL. These windows will allow the network to experience observation sequences from systems starting at different initial health states. The RUL predictions are made using only a single window of observation, so after each window with a certain number of observations, the RUL is predicted without making use of any other observation that came before or after the window. This approach relieves the RUL prediction model from the requirement to have all observations start from full health until the moment of prediction, which is sometimes not available.

4.4 Experiments and Results

4.4.1 Benchmark Dataset Overview

The dataset used for the experiments is the NASA turbofan engine degradation simulation dataset, known as the Commercial Modular Aero-Propulsion System Simulation (CMAPSS) (Saxena, Ieee, et al. 2008). The dataset has four simulation settings. Two of them have single operating conditions and the other two have multiple operating conditions. The datasets are given in the form of training and testing subsets. The training sets have run-to-failure series so the tests stop some time prior to complete failure, thus it is required to forecast the RUL. The time-series contain 26 sensor readings, as well as operating condition indicators as described in (Saxena, Ieee, et al. 2008).

The datasets used in the experiments are the 1st and 3rd, which contain only one operating condition. The 3rd dataset has multiple failure modes. These datasets are chosen since they can be generalized to multiple operating conditions if the operating modes are given by training a model for each operating condition.

Only the sensors that show trends were considered in the analysis, namely sensors number 4, 7, 8, 11, 12, 13, 15, 17, 20, and 21 as described in (Wu et al. 2017). Another piece of information is added to the sensor readings, which is the forward difference of these readings. Since all sensor readings show a trend, such series cannot be considered stationary; hence, we use the forward difference as a detrending procedure to make the analysis of the time series more feasible (Qi and Zhang 2008). The raw sensor readings, as well as the difference, are used as the input to the network. There is no feature extraction step as has been proposed by some of the works of literature such as (Guo et al. 2017; Saxena, Ieee, et al. 2008).

4.4.2 Performance Measures

In order to properly assess the performance of different network architectures, objective functions, and target RUL approximations, a set of performance measures should be first defined.

The measures used in this paper are the asymmetric SF, discussed in (Saxena, Ieee, et al. 2008), which penalizes late predictions more than early ones. Assuming that $d = RUL_{pred} - RUL_{actual}$, then the scoring function is defined as follows

$$SF = \begin{cases} e^{-\frac{d}{13}}, & d < 0 \\ e^{-\frac{d}{10}}, & d \geq 0 \end{cases} \qquad 20$$

The RUL predictions are considered correct if d is in the range [-10, 13] as discussed in (Saxena, Celaya, et al. 2008; Saxena, Ieee, et al. 2008). The percentage of correct RUL is formulated as follows, when the correct indicator $I_c(d) = 1, -10 \leq d \leq 13$, and 0 otherwise

$$Percentage = \frac{1}{n}\sum_{i=1}^{n} I_c(d_i) \qquad 21$$

Where n is the number of testing samples. The Mean Absolute Error (MAE) is defined as follows

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |d| \qquad 22$$

4.4.3 Performance Evaluation Using Different Network Architectures

Table 4.1 and Table 4.2 illustrate the performance measures of different network structures on the validation and training sets for datasets 1 and 3, respectively. The sequences are chosen to be of length 30, and the training set is windowed using this size. A random 20% subset is selected as the validation set. Each structure is run three times and the performance of the ones that perform the best run in the validation set is used to evaluate the performance of the testing set.

The number of LSTM cells in each architecture is selected to be the same: a total of 256 cells for dataset 1 and 512 for dataset 3. The number of LSTM cells is inspired by the results in (Wu et al. 2017). Increasing the number of cells increases the representational capacity of the network to approximate complex functions, yet makes it prone to overfitting. On the other hand, decreasing the number of neurons can cause underfitting, and that is why the number of neurons is estimated by trial and error on a validation set. The number of neurons in the feedforward network on top of the LSTM cells has 1024 neurons for dataset 3, and 512 for dataset 1. A final linear neuron is placed at the output of the network to predict the RUL. Since the focus in this paper is to show the effect of the architecture of the network on the accuracy of the prediction, the number of LSTM cells and feedforward neurons remained constant across the experiments.

For the experiments in Table 4.1 and Table 4.2, the ASE objective function is used with $\alpha_1 = 0.25$, and $\alpha_2 = 9$, which were chosen by trial and error using the validation set performance of dataset 1. Early stopping is used with a tolerance of a maximum of 2 epochs without improvement in the training performance. The batch size used for training is 32. The optimizer used is the RMSprop algorithm (Mukkamala and Hein 2017). The training uses Keras (Chollet 2015) with a TensorFlow (Abadi et al. 2015) backbone.

In the testing phase only the last 30 observations, which is the minimum observation sequence required to be tested, are considered for the prediction task in order to assess the performance, given short sequences of random initial wear.

The results in Table 4.1 and Table 4.2 illustrate that the BHSLTM outperforms the other structures on the test set performance with an average decrease in MAE of 15.5% and 22% for datasets 1 and 3 respectively, although it is not always the best on the training dataset, which means that this structure can generalize better than the other structure.

To assess the effectiveness of the state initialization of the backward processing cells by the final state of the forward processing cells used in the proposed architecture, an experiment is conducted on dataset 1 without the proposed handshake. The results are given in Table 4.3. They show a 25.6% increase in MAE, 16% decrease in accuracy, and a 54% increase in the scoring function.

Table 4.1 Dataset 1 performance measures for different network architectures.

Model	Train Loss	Train SF	Validation Loss	Validation SF	Test SF	Test Accuracy	Test MAE
BHSLSTM (128)	4554.56	218.77	3670.12	152.288			
	5589.12	287.31	5092.67	301.78			
	4593.86	235.17	4323.77	145.54	376.64	63	11.7
2 Layers LSTM (128)	4515.8	213.67	3961.57	224.97			
	5094.9	253.22	5056.98	335.82			
	4884.82	228.97	4032.45	151.4	1120.1	50	14.96
1 Layer LSTM (256)	5273.7	244.1	6186.3	378.1	464.8	57	13.7
	7543.53	512.6	7839.87	652.16			
	5129.47	237.5	6195.96	450.9			
BLSTM (128)	5182.67	243.47	5452.87	375.47			
	4926.5	221.93	4475.2	213.23	542.6	58	13
	5288.86	241.2	4377.4	243.95			

Table 4.2 Dataset 3 performance measures for different network architectures.

Model	Train Loss	Train SF	Validation Loss	Validation SF	Test SF	Test Accuracy	Test MAE
BHSLSTM (256)	6409	485	8747	1677			
	7914	766	6196	670			
	5767	464	5373	527	1422	52	15.79
2 Layers LSTM (256)	6550.89	507	5747	414.45	4931	45	18.36
	7584.9	1096	5828	519			
	5859	366.14	6668.7	530.9			
1 Layer LSTM (512)	8458.5	699	10213	502.4	8158	42	23
	10907.65	997.56	9238	1181.9			
	11435	1160.25	10633	1480			
BLSTM (256)	7947.42	581.52	7158.19	321.5	4715.3	43	19.855
	8400.29	756.75	7061.85	581.25			
	6962	538	7944	867.73			

Table 4.3 Bidirectional LSTM performance without the proposed handshake procedure for dataset 1.

Train Loss	Train SF	Validation Loss	Validation SF	Test SF	Test Accuracy	Test MAE
5680.26	285.75	7161.5	596.55	581.5	53	13.59
4745.56	216.18	4038.54	202.35			
4827.67	219.8	4653.3	245.56			

4.4.4 Performance Evaluation Using the Mean Squared Error with BHSLTM

To assess the performance improvements when using the proposed asymmetric objective function, another set of experiments using the proposed BHLSTM is conducted on both datasets using the MSE objective function.

Table 4.4 BHLSTM performance using the MSE objective function for dataset 1.

Train Loss	Train SF	Validation Loss	Validation SF	Test SF	Test Accuracy	Test MAE
152.57	113	201.76	111.22	2407.8	26	20.4
145.36	106.16	165.58	144.156			
161.98	126.3	229.69	177.1			

Table 4.5 BHLSTM performance using the MSE objective function for dataset 3.

Train Loss	Train SF	Validation Loss	Validation SF	Test SF	Test Accuracy	Test MAE
181.8	230.4	224.05	389.39	6875	28	35.27
193.32	287.15	413.183	324.72			
142.17	140.68	200	273.54			

By comparing the performance of the BHLTSM in Table 4.1 and Table 4.2 to those in Table 4.4 and Table 4.5 a significant drop in performance on the test datasets is noted. For dataset 1 a 75% increase in MAE is noted and 49.75% increase for dataset 3. As a means for a visual assessment of the effect of the objective function on the estimation of the RUL, Figure 4.5 illustrates the estimations of a BHLSTM network trained with ASE and MSE along with the actual target RUL for two of the training engines as well as the error distributions. The predictions at each cycle use only the previous 30 sensor readings, not the whole sequence up to the point of forecast.

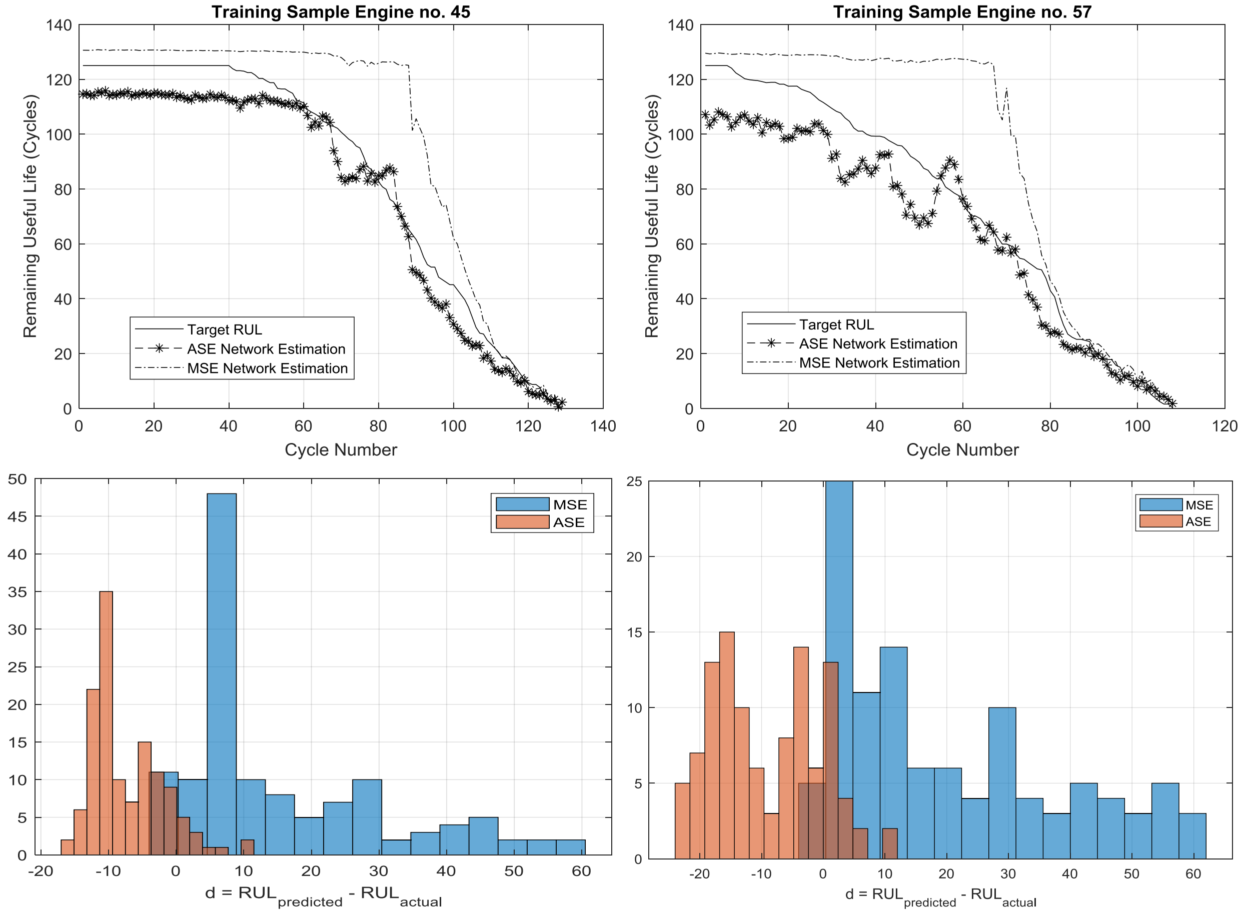


Figure 4.5 Comparison between RUL forecasts using ASE and MSE objective functions.

Figure 4.5 shows two merits for the ASE objective function trained network. First, its forecasts are mostly safe and come before the actual RUL. Even the forecasts that come after the actual RUL, like around cycle 80 for engine number 45 and cycle 60 for engine number 57, are not too late. The second merit is that the network trained using the ASE objective approximates the actual RUL curve much better than the other network using the MSE objective function.

4.4.5 Performance Evaluation Using the Piecewise Linear Target RUL

Finally, to assess the performance improvement when using the proposed target RUL generation procedure with the fixed piecewise RUL proposed by (Heimes 2008), the BHLSTM is trained with such RUL for dataset 1 and the results are given in Table 4.6.

Table 4.6 BHLSTM performance on dataset 1 using the piecewise linear RUL.

Train Loss	Train SF	Validation Loss	Validation SF	Test SF	Test Accuracy	Test MAE
11668.16	1672.75	10300.94	1806.76			
10044.38	1478.14	10413.44	1384.8			
11344.18	1720.16	10288.53	2072.7	616.1	54	14.03

Again, by comparing the results of the BHLSTM in Table 4.1 with Table 4.6 we can see that the performance of the test set deteriorates by a 23% increase in MAE, 6% decrease in accuracy, and a 63.5% increase in the score function.

4.5 Discussion and Conclusion

In this paper, a new Long Short-Term Memory (LSTM) network architecture, objective training function, and training target generation are proposed. These proposals aim at solving the problem of making Remaining Useful Life (RUL) predictions for physical systems using short sequences of observations with random starting health conditions more efficient.

The proposed Bidirectional Handshaking LSTM (BHLSTM) network architecture uses a bidirectional sequence processing approach in a sequential manner. The forward processing LSTM units pass their final states to the backward processing units instead of simultaneously partially processing the sequence up to the time of prediction, which is more suitable for making intermediate predictions rather than making a single prediction after observing a sequence, such as in the case of RUL prediction. The summary vectors of both directions of the BHLSTM are concatenated for the higher classification layers.

This paper also proposed a new, asymmetric objective function that penalizes late predictions more than earlier ones, thereby ensuring safer predictions. This is in contrast to the commonly used mean squared error objective function, which is a symmetric function.

Finally, the proposed target generation for the RUL training requires no assumptions about the degradation function part, nor the point at which the degradation starts. The proposed approach uses the sensor readings to estimate the health of the system, which is then mapped to the target RUL. This decreases the amount of parameter tuning required by previous works of literature. Each of the proposed approaches can be used along with other types of neural networks, as well as for training other machine learning models.

Table 4.7 Summary of Potentials and Limitations of BHLSTM

Proposed Technique	Bidirectional Handshaking LSTM
Rational, Need	Predicting when a partially observed time-series will reach a certain threshold based on historical data
Potential in industry	Better scheduling for maintenance for assets that undergo partial maintenance, or enter the system in a used state
Benchmarks used	Datasets 1 and 3 from the NASA turbofan engine degradation simulation dataset, known as the Commercial Modular Aero-Propulsion System Simulation (CMAPSS)
Average performance improvement against benchmark	18.75% decrease in the Mean Absolute Error of prediction
Scalability of the algorithm	This technique requires a sufficient amount of data to be trained. So, the more the better.
Computational Complexity	$O(d^3NT)$ where d is the number of sensor readings, N is the number of LSTM cells, and T is the number of observations in the partially observed time-series
Extendibility to other potential applications	Can be used for any type of time-series data that show a certain overall trend with intermediate changes and the time to reach a certain threshold value is of interest
Inputs to the algorithm	Multivariate subsequences of time-series data labeled with the desired threshold value on its magnitude
Output from the algorithm	Prediction of the time remaining until the time-series reaches in a certain magnitude threshold
Parameters to set / decided upon	The training algorithm, the number of neurons, the number of layers, and the type of activation function for the network. A grid search can tune the required parameters on the validation dataset.

CHAPTER 5 ARTICLE 3: A PROFILE FAVORING DYNAMIC TIME WARPING FOR THE HIERARCHICAL CLUSTERING OF SYSTEMS BASED ON DEGRADATION

Ahmed Elsheikh; Mohamed Salah Ouali; Soumaya Yacout

Submitted to: Computer and Industrial Engineering Journal

Abstract

Efficient management of a deteriorating system requires accurate decisions based on in situ collected information to ensure its sustainability over time. Since some of these systems experience the phenomenon of degradation, the collected information usually exhibits specific degradation profiles for each system. It is of interest to group similar degradation trends, characterized by a multivariate time-series of collected information over time, to plan group actions. Knowledge discovery is one of the essential tools for extracting relevant information from raw data. This paper provides a new method to cluster and visualize multivariate time-series data, which is based on a modified Dynamic Time Warping (DTW) distance measure and hierarchal clustering. The modification adds emphasis on finding the similarity between time-series that have the same profiles, rather than by magnitudes over time. Applied to the clustering of deteriorating road segments, the modified DTW provides better results according to different cluster validity indices when compared to other forms of DTW distance technique.

Keywords

Dynamic time warping, Multivariate time-series clustering, Pavement management.

5.1 Introduction

The growing availability of data from different industrial plants and advancements in their acquisition techniques prompt further exploration (Abellan-Nebot and Subrión 2009). Discovering a hidden structure in newly observed, raw data can be very beneficial for proactive decision making (Jardine et al. 2015; Vogl, Weiss, and Helu 2016). The data contains observations that characterize the condition of a system state and its behavior over time. For some applications, such as machine diagnosis (Mohamad-ali, Yacout, and Lakis 2014), or product quality inspections (Tiejian Chen et al. 2016), the data collected at different time intervals can be treated as the independent system state condition. However, when the sequence of observations is highly correlated (Ghasemi, Yacout, and Ouali 2010) as is normally expected from degradation and maintenance processes, these observations have to be considered as time-series, which is a set of values collected by order of observation in time (Barragan, Fontes, and Embiruçu 2016). This paper focuses on revealing structures by clustering sequentially correlated data where the profile of the time-series is of interest. The profile describes how the series changes in magnitude from one time to the next. This is highly evident in degradation data, where mining for trends of degradation can be beneficial to decision making.

Clustering depends mainly on the definition of a distance measure. Dynamic Time Warping (DTW) is one of the most well known time-series distance measures (Fu 2011). It has been used for Multivariate Time-Series (MTS) classification (Górecki and Łuczak 2015) and in different applications with data of a sequential nature, such as speech recognition (Myers, Rabiner, and Rosenberg 1980; Sakoe and Chiba 1978), online handwriting recognition (Efrat, Fan, and Venkatasubramanian 2007), and sign language recognition (Kuzmanić and Zanchi 2007). The DTW presents a solution to match variable length MTS that may contain shifts, expansions, or contractions in time (Górecki and Łuczak 2015; Kruskall and Liberman 1983).

The classical DTW favors values of magnitude rather than the signal profile. Another variant of the DTW is to take the first order derivative between the time samples of the series and use it as input to the DTW. This variant groups the time-series with the same profile together, and is called Derivative DTW (DDTW) (Keogh and Pazzani 2001). However, DDTW completely disregards the magnitudes of the observations of the matched series.

It has been previously shown that neither the classical DTW nor the DDTW can consistently perform well on its own. A weighted average between both distances was suggested by (Górecki and Łuczak 2013). The optimal value of this weighting factor can be found by trying different values for the factor and then assessing the classification performance using cross-validation, until the best performing weight is found. This is an exhaustive search procedure, which is time consuming and must be adjusted for every task or when new data is available for a certain task.

Moreover, the evaluation of that weighting factor is less straightforward for an unsupervised task since there are no known target labels for the available task.

To allow DTW to favor time-series profiles rather than the magnitudes of the observations at different time samples in the series, the proposed modification aims to find the best alignment between the matched series using the warping path of the DDTW and re-align the original series, then compare them using Euclidean Distance (ED). This way, the series are properly aligned according to their profiles and then their magnitudes are taken into account without requiring extra parameters. Series alignment was used with the classical DTW for the purpose of series averaging by (Abdulla et al. 2003; Petitjean, Ketterlin, and Gançarski 2011).

This paper demonstrates the advantage of using the profile favoring DTW along with hierarchal clustering to identify useful patterns in time-series data. Several attempts throughout the literature have been made for the purpose of uncovering a structure from time-series data (Aghabozorgi, Seyed Shirkhorshidi, and Ying Wah 2015; Fu 2011; Warren Liao 2005). Statistical modelling assumes that sequential data is generated from a random process of a certain nature. Examples of such statistical modelling include Hidden Markov Models (HMMs) (Ghasemi, Yacout, and Ouali 2010; Rabiner 1989) and the time-series Bayes model (Van der Heijden, Velikova, and Lucas 2014).

Another group of algorithms depends mainly on transforming the time-series into another domain where regular clustering and pattern recognition techniques can be directly applied. Principal components analysis (PCA) (Bankó and Abonyi 2012), discrete Fourier transform (DFT) (Rafiei and Mendelzon 2000), and discrete wavelet transform (DWT) (Barragan, Fontes, and Embiruçu 2016; Durso and Maharaj 2012) are some examples of such transformations.

Last but not least, there are regression algorithms that try to find a parameterization for future samples given current and previous samples. The most popular are the autoregressive (AR) model, and its variants: Autoregressive Moving Average ARMA (Xiong and Yeung 2004) and Autoregressive Integrated Moving Average (ARIMA) (Corduas and Piccolo 2008). A currently evolving research direction is to find a non-linear parameterization of the series using Recurrent Neural Networks (RNNs) (Tao Chen et al. 2016; Pacella and Semeraro 2007).

All of these algorithms require parameter estimation, as well as the knowledge of the number of underling processes that generate the data that is observed. If the processes that are observed have different modes of operation, then a model or a transformation of each mode should be considered independently. Examples of such multi-modal behavior appears for machining tools (Aramesh et al. 2014), city roads with different usage rates (Ben-Akiva and Ramaswamy 1993), and aircraft engines (Korson et al. 2003). These assets undergo change and sometimes end up in unknown or unexpected operating conditions. Uncovering groups of behaviors and finding structure in the gathered data can be beneficial for decision making that concerns design precautions and maintenance planning.

The approach discussed in this paper uses a non-parametric unsupervised method to uncover distinct groups from time-series' databases. The method applies a modified (DTW) distance measure, which favors time-series' profiles along with hierarchal clustering to uncover the groups. These groups of time-series of similar profiles are very useful for planning maintenance tasks for physical assets. The proposed approach aims to minimize the analyst's interaction with any parameter selection while revealing useful information from the data.

Section 2 characterizes the different types of time-series clustering and the scope of the proposed method, and explains its advantages by different numerical and visual illustrations. Section 3 shows the results for a real application on the pavements' health data along with a performance assessment using different cluster validity indices, which are adopted for the time-series clustering task. Section 4 concludes with the findings of the paper.

5.2 Time-series Clustering

Previous approaches for time-series clustering can be grouped into three main categories based on the underlying theory (Aghabozorgi, Seyed Shirkhorshidi, and Ying Wah 2015; Warren Liao 2005):

1) Model-based approaches, which model the time-series by using a set of parameters under certain assumptions, such as HMM, ARIMA, and RNN, which have been discussed in the previous section.

2) Feature-based approaches, which are essentially the transformation methods such as PCA and DWT that are discussed in the previous section. Both approaches describe the time-series in a

set of parameters or projections, which have a fixed dimensionality. These sets can be presented in a vector form for the conventional clustering method.

3) Shape-based approaches, which use the original series or subsequences of it without any transformation to find the distance between different time-series. The proposed approach falls under this third category.

There is another categorization scheme for clustering methods based on the objective. This categorization identifies three main objectives for time-series clustering (Fu 2011):

1) Whole series clustering, which considers each time-series as a single entity. This is challenging because not all time-series have the same number of observations.
2) Subsequence clustering, which targets clustering subsequences or windows of the original time-series.
3) Point clustering, which clusters the series observations according to their proximity in time as well as their values.

The proposed approach is of the whole series clustering type. According to (Aghabozorgi, Seyed Shirkhorshidi, and Ying Wah 2015; Barragan, Fontes, and Embiruçu 2016), Variable Length Multivariate Time-Series (VLMTS) clustering has not been well exploited in the literature. The proposed method is motivated by this problem.

The clustering procedures are mainly split into three categories (Everitt et al. 2011):

1) Density estimation using probabilistic models such as the Gaussian Mixture Model (GMM).
2) Iterative optimization clustering such as K-Means (KM) and Fuzzy C-Means (FCM), which iteratively try to divide the data into a given number of clusters.
3) Hierarchal clustering, which result in a series of partitions starting from the individual samples up to one cluster containing all the data. This hierarchy can be visualized using a graphical tree representation called the dendogram. In this paper, the hierarchal clustering is chosen to be the

grouping method, as it provides more intuition about the data, and the number of clusters can be estimated in a relatively more straightforward way than other methods.

Distance measures are an essential part of the clustering procedure for the second and third categories. The first category instead calculates a probability of membership, which can also have a distance measure interpretation (Bishop 2006).

5.3 Dynamic Time Warping (DTW)

Instead of matching one-to-one the value of the series' samples, which is applicable only when using fixed length sequences or subsequences, DTW searches for the best warping path. This warping path is the best alignment between the two matched series such that the minimum accumulated distance between their observations is achieved. The distance between the series' observations is any of the commonly used measures such as Euclidean Distance (ED), absolute difference, cosine distance, or Hamming distance for discrete data. The ED is chosen in this paper for the experiments discussed in this work.

The formula for the DTW distance between two time-series X and Y, assuming that series $X = (X_1, \dots, X_n)$, and $Y = (Y_1, \dots, Y_m)$ are two MTS with lengths n and m respectively, where X_i, and Y_j are the multidimensional vector observations of the series, is as follows:

$$DTW(X,Y) = \min_P \sum_P d_P(X_i, Y_j), \; \begin{cases} i = 1, \dots, n \\ j = 1, \dots, m \end{cases} \qquad 23$$

Where P is the warping path, which is the list of observation pairs (one from each series) that are the most similar, and $d_P(X_i, Y_j)$ is the distance between the time samples X_i of series X and Y_i of series Y along the warping path. The ED $d_P(X_i, Y_j) = \sqrt{\sum_{k=1}^{n_d} (X_{ik} - Y_{ik})^2}$, where X_{ik} and Y_{ik} are the k_{th} dimensions of the observation vectors, and n_d is the number of dimensions of all observations.

This is a combinatorial search problem that seeks to find the optimal match between the two series. Since this is computationally intensive and its complexity increases exponentially with the lengths of the two series, a dynamic programming solution is used with some specific constrains to ensure that the sought warping path is an acceptable match between the two time-series, as follows:

1) Boundary conditions: since the aim is to match two whole time-series, then the first and last observations in both time-series must be included in the path, otherwise one of them can be matched to the sub-section of the other.
2) Monotonicity condition: since time is an evolving variable, matching should be from present to present, or present to future. The path does not allow going back in time to find a match with previous samples.
3) Step size condition: since time samples define each series, then it is not acceptable to skip any sample from either series. This means that the warping path must go through all time steps without skipping any time instant.

Figure 5.1 illustrates the procedure using a Univariate Time-series (UTS). Each grid point in the shown matrix represents the ED between the i_{th} observation in series X and the j_{th} observation in series Y. As indicated in the legend, the darker grid points represent smaller distance, which means higher similarity. The line shown in red indicates the warping path. Horizontal parts in the warping path indicate expansions in time-series X relative to Y. On the other hand, vertical parts in the path indicate compressions in time-series X relative to Y.

5.3.1 Finding the minimum Warping Path

Dynamic programming is a procedure that decreases the computational complexity of search problems from the exponential with series length to linear. This is done by defining accumulators to store intermediate computations, which in turn saves re-computations.

Assuming the same series defined for equation 23, the procedure is as follows:

1) Calculate the pairwise distance matrix $d_{n\times m}$ between every pair of observations X_i and Y_j. This is depicted in Figure 5.1 by the color scale matrix. The values are indicated by the legend.

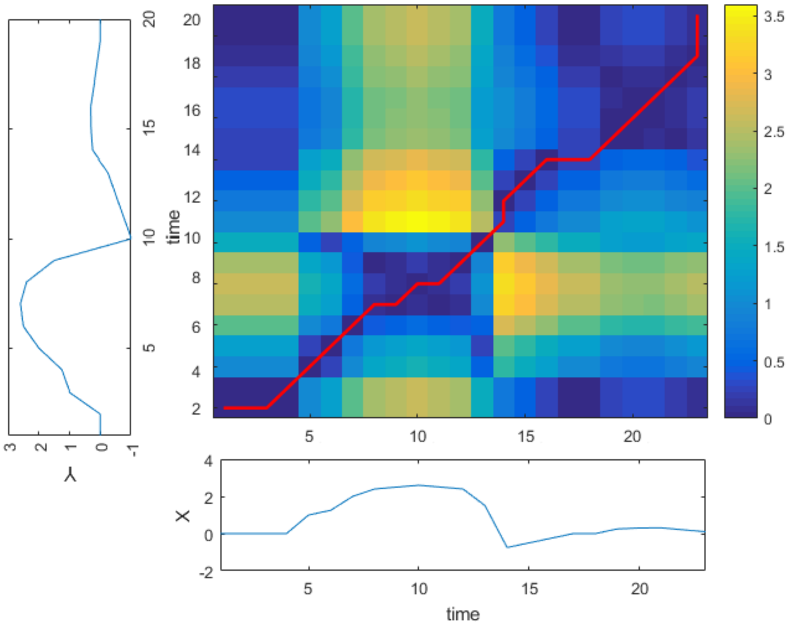


Figure 5.1 Warping path between two univariate time-series.

2) Calculate the accumulated distance matrix $D_{n\times m}$ according to the following equation

$$D_{i,j} = \min\{D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1}\} + d_{i,j} \qquad 24$$

3) The warping path $P = (p_1, \dots, p_L)$ is found by backtracking, i.e. starting at the terminal location $p_L = (n, m)$ and tracing back the path with minimum accumulated distance.

$$p_{l-1} = \begin{cases} (1, m-1), \;\; if\ n = 1 \\ (n-1, 1), \;\; if\ m = 1 \\ \operatorname{argmin}\{D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1}\}, o.w. \end{cases} \qquad 25$$

5.3.2 Profile Favoring DTW

The first step in the proposed method is to find the warping path as illustrated in equations 23 to 25, but instead of using the time-series themselves, the first-order difference is used. Assuming any series X as defined above, the first-order difference of X is defined as

$$X_d = \left(X_{d_1}, \dots, X_{d_{n-1}}\right) \mid X_{d_i} = X_{i+1} - X_i, \qquad i = 1, \dots, n-1 \qquad 26$$

Where X_d is the first-order difference of the series X, and n is the number of observations of the series. The sequence of observations characterizes the evolution of the time.

The second step is to calculate the distance between the aligned original series using any distance measure such as the ED.

A simple example to illustrate the proposed method is given in Figure 5.2. This figure shows three UTS; $S1 = (1,4,4,1,1)$, $S2 = (2,5,5,2,2,2)$, and $S3 = (5,2,2,3,3,2)$. The behavior of $S1$ is similar to $S2$ but with lower magnitude and a fewer number of samples.

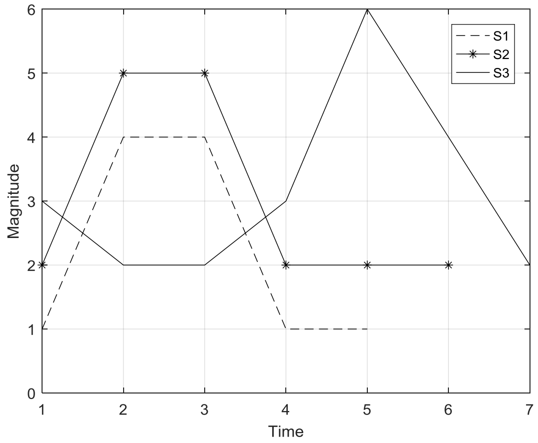


Figure 5.2 Illustrative example for comparing DTW distances for UTS.

The three series are matched by using three variants of the DTW; (1) the classical DTW illustrated in this section and identified by equation 24, (2) the DTW with alignment that uses the warping path identified by equation 25, then matches the two aligned series by using ED, and finally, (3) the proposed profile favoring DTW, which uses the warping path of the first order derivative of the series given in equation 26, and then matches the two aligned original series by the ED. Table 5.1 summarizes the results of this synthetic example. The proposed distance identified the two series $S1$ and $S2$ have the same profile in order to be closer to each other, unlike the other two classical DTW distances.

Table 5.1 Comparison of DTW distances for the UTS synthetic example.

Series \ Distance measure	DTW	DTW Alignment	Profile Favoring DTW
S1, S2	6	2.4495	2.4495
S1, S3	9	3.6056	5.1962
S2, S3	4	2	5.6569

5.4 Hierarchal Clustering

To achieve the desired knowledge discovery in a VLMTS dataset, the proposed distance measure needs to be used within a clustering method to uncover hidden structures in the data. Hierarchal clustering (HC) is a type of clustering that produces various levels of clustering of the data available. In this paper, HC is chosen because it provides some insight about the structure of the data in space, which in turn provides better intuition about the expected number of clusters (Everitt et al. 2011). It is not like partitional clustering, which requires prior knowledge of the number of clusters and enforces a certain shape for encompassing the cluster members (Barragan, Fontes, and Embiruçu 2016).

5.4.1 Procedure for Bottom-Up Hierarchal Clustering

The input to this method is the pairwise distance between all sample points in the dataset. The method starts by grouping the most similar pair of entities together according to a given similarity measure and treats them as a cluster in the next iteration. The method iterates until it ends up with a single group containing all data samples (Murphy 2012). The similarity between groups is measured using one of three main links between any two groups G_1 and G_2:

5.4.1.1 A) Single link

This is defined as the distance between the two closest members, which are the most similar according to the chosen distance measure, whether it is DTW, DTW alignment, or Profile favoring DTW, of each group. It can be defined as

$$D_{SL}(G_1, G_2) = \min_{g \,\epsilon\, G_1,\ g' \,\epsilon\, G_2} d_{g,g'} \qquad 27$$

Where $D_{SL}(G1, G2)$ is the single link distance between the two groups $G1$ and $G2$, and $d_{g,g'}$ is the distance between any two members g and g' of the groups $G1$ and $G2$, respectively. This tends to produce large contiguous clusters. Nevertheless, single link is vulnerable to noisy data. This vulnerability is termed chaining, which means it tends to link clusters together more than required in the presence of noise.

5.4.1.2 B) Complete link

Complete link is defined as the distance between the most distant pairs of each group. It can be defined as

$$D_{CL}(G_1, G_2) = \max_{g \in G_1,\ g' \in G_2} d_{g,g'} \qquad 28$$

This measure favors compact clusters, which have low variance amongst its members. Complete link is more immune to noise, but might break large clusters into smaller ones.

5.4.1.3 C) Average link

Average link is a compromise between the two types of linkage above. It can be defined as

$$D_{avg}(G_1, G_2) = \frac{1}{n_{G_1} n_{G_2}} \sum_{g \in G_1} \sum_{g' \in G_2} d_{g,g'} \qquad 29$$

Average link is used for the experiments in this paper since it is a balance between the other two links.

5.4.2 Effect of Profile Favoring DTW on Clustering

In order to more clearly demonstrate time-series clustering, we provide an example for the visualization of a synthesized dataset from the UCR Time-series Classification Archive (Y. Chen et al. 2015). This type of analysis allows the analyst to see the effect of the distance measure used on the space where the time series is being clustered. This example uses three types of real UTS with the same range of magnitudes, but slightly different profiles and variable lengths. Samples from each time-series are shown in Figure 5.3.

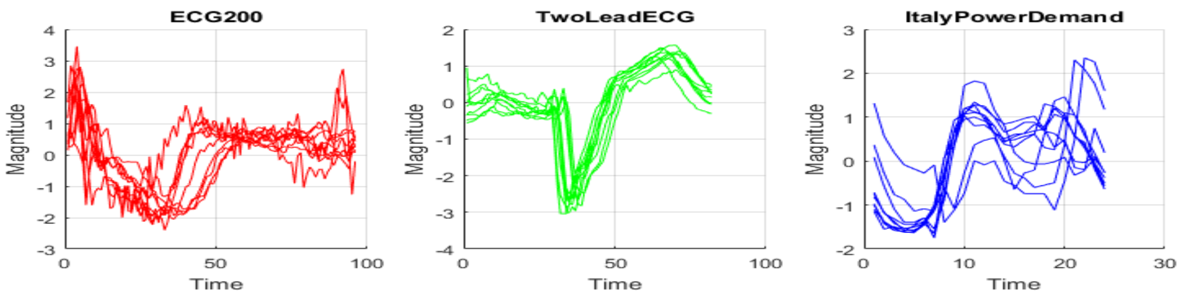


Figure 5.3 Example UTS with different modes of behavior.

Any distance measure can be interpreted as a kernel (Scholkopf 2001). Kernels can be simply thought of as a similarity measure between samples, which is the exact opposite of the notion of distance. The conventional DTW was used as a kernel for SVM in (Bahlmann, Haasdonk, and Burkhardt 2002). Principal Components Analysis (PCA), which is an approach for dimensionality reduction (Bishop 2006), is represented in kernel form and termed Kernel PCA (KPCA) (Bernal De Lázaro et al. 2015). Hence, the effect of the modified DTW is visualized by using KPCA with the DTW kernel. This allows even VLMTS to be visualized in a two-dimensional plot.

The scatter plot of the three UTS modes of behavior that are shown in Figure 5.3 is shown in Figure 5.4 with different forms of DTW, namely the conventional DTW, the DTW with alignment, and the proposed profile favoring DTW, which were discussed in the previous section. The color labels represent the actual ground truths. It is clear how the proposed DTW, namely the profile favoring, better discriminates the series, which have different modes. It identifies that there are three modes. Cluster validity indices confirm this conclusion is reached in the next section.

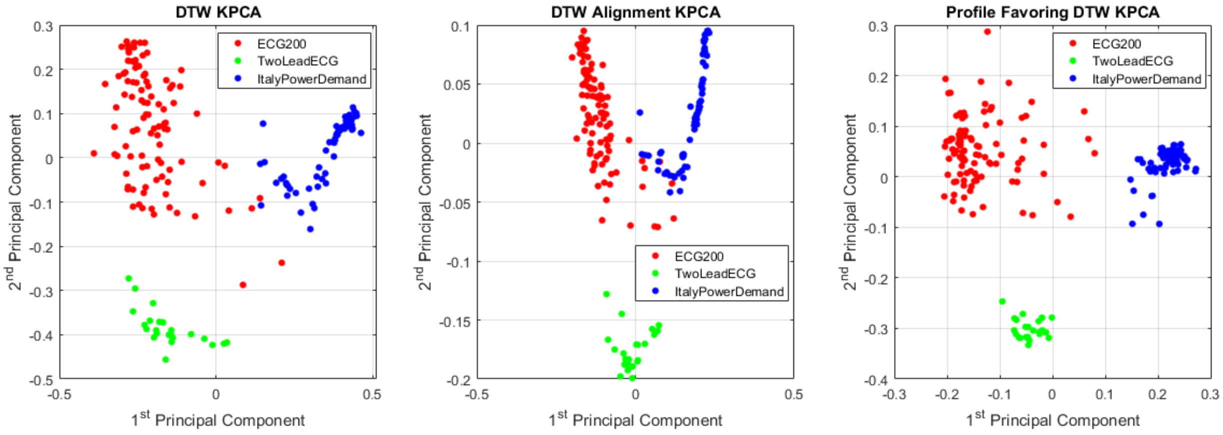


Figure 5.4 Visual Comparison between the variants if the DTW distances in the low dimensional KPCA projection.

Figure 5.5 shows the dendrograms of the HC using the average link of equation 30 for the different variants of the DTW distance. This is another way to depict how the series can be grouped together. It is important to note that theses dendrograms do not correspond to the distribution depicted in the KPCA plots. This is due to the fact the KPCA only retains a fraction of the similarities between the samples (Bernal De Lázaro et al. 2015). The heights of the branches in the dendogram indicate the distances between samples. The red dots indicate the potential cluster splits. The potential splits are placed between groups of samples that are separated by large distances from each other. From the three dots that are obtained when using the profile favoring DTW, along with average link HC shown in equation 30, it can be concluded that the proposed modification reveals the three UTS modes. Other samples that do not fall in any of the groups that are indicated by the red dots are considered as outliers of the nearest cluster. The potential splits are not clear for the DTW alignment distance because the distances between different groups of samples are very similar. This indicates that the samples seem closer to each other using this distance, which confirms the visual result of the KPCA.

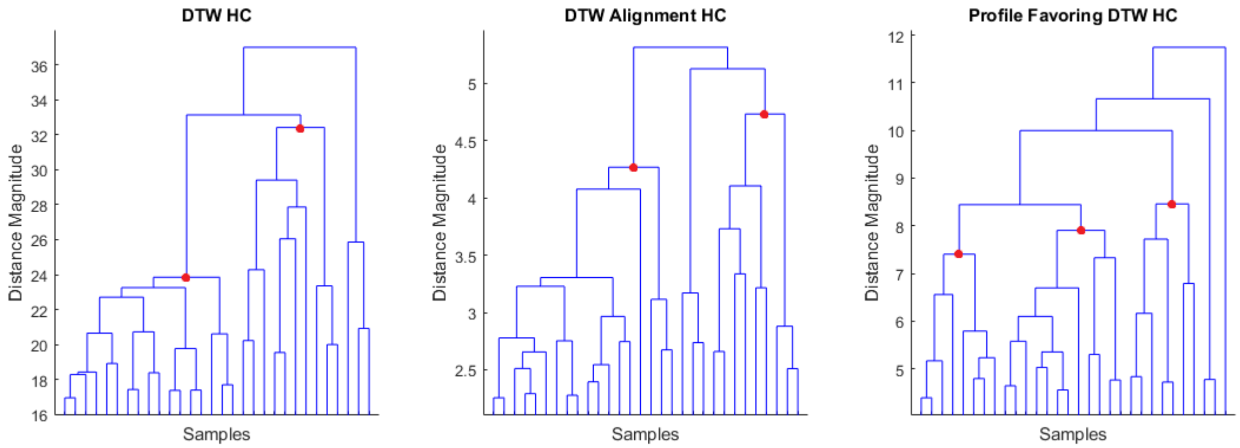


Figure 5.5 Comparison between the dendograms of the DTW distances.

In the next section, an example that is based on a real application is presented, and cluster validity indices are introduced in order to show how the proposed profile favoring DTW distance measure is applied to real Variable Length UTS (VLUTS) and VLMTS data.

5.5 Example of an application

The example of an application concerns the clustering of road segments that have a similar degradation profile over the time. The data is provided by the Ministère des Transports de la Mobilité Durable et de l’Électrification des Transports (MTMDET) du Québec. A subset of 21 road degradation profiles is used to validate the proposed clustering method. Each road segment is characterized by three main performance measures (PM) that indicates its degradation state (Sayers and Karamihas 1998), namely:

1) The International Roughness Index (IRI): which measures the amount of variation and the decrease in smoothness in a road segment.

2) Rutting: which is the amount of depressions in a road segment due to tires.
3) The total number of cracks in the road segment.

Figure 5.6 shows the first performance metric (PM1) versus the observation sequence of the 21 road segments. This data assesses the need for a profile favoring DTW, since there are degradations with different profiles yet with similar magnitudes, and others having similar profiles yet with large magnitude differences.

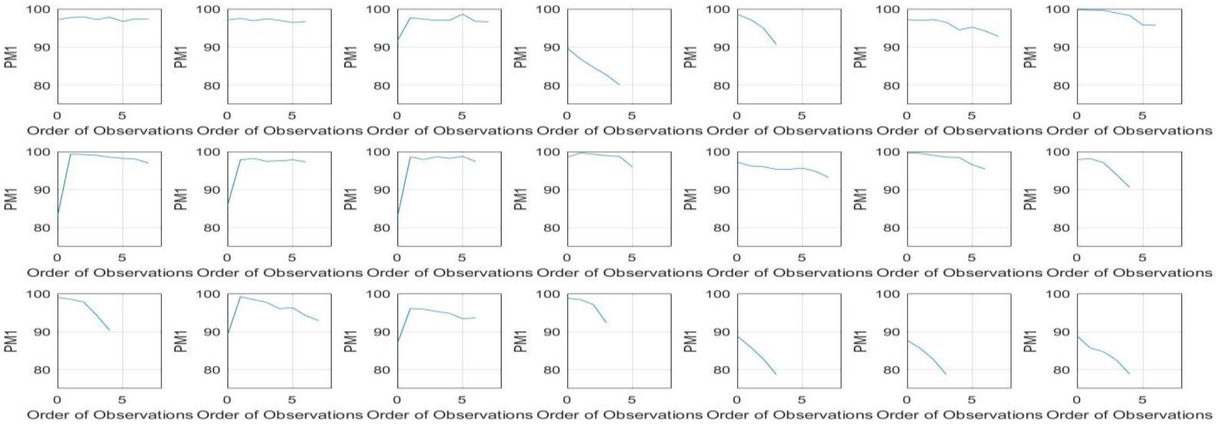


Figure 5.6 Metric of real data on the degradation of road segments versus observation sequence.

The clusters of similar road segments are formed based on hierarchical clustering with the variants in the DTW distance. The validity of the obtained clusters is evaluated by using three validation indices as well as the visual plots of the clusters, as was explained in the previous section. The validation indices used are the Silhouette (Sil) index (Rousseeuw 1987), a variation of the Davies–Bouldin (DB) index (Kim and Ramakrishna 2005), and the Calinski–Harabasz (CH) index (Caliński and Harabasz 1974). These indices are the three top performing validity indices as suggested by the extensive review in (Arbelaitz et al. 2013). These indices were modified to suit the time-series clustering problem. Since the mean of time-series of different lengths is not well defined (Fu 2011), we use the medoid instead of the centroid, which is a sample that has a minimum distance to all of the other time series samples in a given dataset S. More formally,

$$Med = \underset{i}{\operatorname{argmin}} \sum_{x_j \in S \backslash x_i} d(x_i, x_j)\,, x_i \in S \qquad 30$$

Where Med is the medoid, and x_i, x_j are time series samples in the dataset S

5.5.1 Cluster Validity Indices

For a given distance measure $d(x, y)$ between any two time-series, $C = \{c_1, \dots, c_k\}$ is a set of k clusters for a total of N road segments, and each cluster has members x_i with a count of N_{c_i} and medoid $\bar{c}$ then the indices are defined as follows:

5.5.1.1 Silhouette Index (Sil)

This index identifies how each road segment is in tact with its cluster. This is done by calculating the average distance between each sample and all other samples within its cluster, and subtracting it from the minimum average distance between that sample and its closest neighboring cluster. The formulation is as follows:

$$Sil(C) = \frac{1}{N} \sum_{c_i \in C} \sum_{x_i \in c_i} \frac{\beta_{x_i} - \alpha_{x_i}}{\max\{\beta_{x_i}, \alpha_{x_i}\}} \qquad 31$$

Where $\alpha_{x_i} = \frac{1}{N_{c_i}} \sum_{x_j \in c_i} d(x_i, x_j)$, and $\beta_{x_i} = \min_{c_j \in C \setminus c_i} \left\{ \frac{1}{N_{c_i}} \sum_{x_j \in c_j} d(x_i, x_j) \right\}$. The value of this index ranges from -1 to 1, where -1 indicates that the average distance within a given cluster is more than the average distance between distance the samples of that cluster to the samples of others of neighboring clusters suggesting erroneous clustering. On the other hand, a value of 1 indicates that the average distance within a cluster is much less than the average distance between the samples of a cluster to the samples of other neighboring clusters suggesting compact, and well separated clusters.

5.5.1.2 Modified Davies–Bouldin Index (DB)

This measure calculates the ratio between the scatter within each cluster, which is the average distance between samples in a cluster and its centroid, and the distance between cluster centroids. The formulation is given by:

$$DB_{mod}(C) = \frac{1}{N_{c_i}} \sum_{c_i \in C} \frac{\max_{c_j \in C \setminus c_i} \{\sigma(c_i) + \sigma(c_j)\}}{\min_{c_j \in C \setminus c_i} \{d(\overline{c_\iota}, \overline{c_J})\}} \qquad 32$$

Where $\sigma(c_i) = \frac{1}{N_{c_i}} \sum_{x_i \in c_i} d(x_i, \overline{c_\iota})$. The smaller the value of this index, the more precise the clustering is.

5.5.1.3 Calinski–Harabasz Index (CH)

This index calculates the ratio between the scatter of cluster centroids around the global mean to the scatter within each cluster. The formulation is as follows:

$$CH(C) = \frac{N-k}{k-1} \frac{\sum_{c_i \in C} N_{c_i} d(\overline{c_\iota}, \mu)}{\sum_{c_i \in C} \sum_{x_i \in c_i} d(x_i, \overline{c_\iota})} \qquad 33$$

Where μ is the global mean of all the samples. The value of this index increases when the clusters are compact and are separated from each other.

These clustering validity indices are not similar since each one is concerned with a different aspect of the clusters as illustrated, and also as reported in (Arbelaitz et al. 2013). That is why decisions and performance assessment cannot rely on a single index. Hence, a vote is made among the indices to identify cluster validity. The vote is in favor of the majority, which is at least two when using the above three indices.

5.5.2 Results Using a Single Road-Performance Metric

The hierarchal clustering used was not cut based on a prespecified number of clusters. Instead, a threshold based on the inconsistency measure of the hierarchal clustering is used to identify the proper number of clusters (Bailey et al. 2007). The idea is to measure the scatter between links of the hierarchy, this scatter corresponds to distances between potential clusters at every level, and to choose the best cut, which maximizes this scatter. The inconsistency measure is calculated as follows:

$$inconsistency = \frac{h_l - \mu_l}{\sigma_l} \qquad 34$$

Where h_l is the height of the link at level l of the hierarchy, μ_l is the average height of links at level l, and σ_l is their standard deviation.

The validity indices for clustering using the DTW variants using only the first road-performance metric PM1 clustering are summarized in Table 5.2.

In general, for any hierarchal clustering procedure when the selected threshold for the inconsistency - which is also termed cutoff - of the hierarchy decreases, the number of resulting clusters increase.

In general, the profile favoring shows overall better validity values than the other conventional distance measures, which are the DTW and the DTW with alignment. The reason is that two or more of the validity indices are always in favor of the proposed method. For each of the DTW variants, a vote is made to select the most proper set of clusters from the results of using different cutoffs.

By noting that the DB index identifies higher quality clusters by lower values unlike the other two indices, we illustrate an example of how to identify the best number of clusters using the validity indices for each of the DTW distance variants.

In the case of the profile favoring DTW distance, the CH and DB agree that 7 clusters are better than 5, while the Sil and CH agree that 7 clusters are better than 14. For the other two conventional distances, the Sil and DB agree that 14 clusters are better than 6 for DTW and 14 clusters are better than 7 for DTW with alignment. This arises from the fact that the clusters derived from the conventional distances suffer from high non-homogeneity; this is why using those distances makes it more valid to break away clusters.

A visualization of the results for the clusters of PM1 series identified by the validity indices using cutoff 1.0 are shown in Figure 5.7, Figure 5.8, and Figure 5.9. Each sub-plot in these figures shows the members of a single cluster. By comparing the members of each DTW distance variant, it can be noted that the clusters that are found by using the proposed profile favoring DTW are the most similar in terms of behavior. Examples of inconsistent clusters are clusters 2 and 6 using the DTW distance, and cluster 5 using the DTW with alignment distance.

5.5.3 Results Using Multiple Road-Performance Metrics

Table 5.3 summarizes the results of the validity indices using all three road performance metrics. Of course it is not possible to depict how three variables evolve with time on a 2D plot as in the UTS case.

Table 5.2 Single-metric road-degradation data cluster validity summary of results.

DTW					DTW with Alignment					Profile Favoring				
Clustering Cutoff	Silhouette	Calinski - Harabasz	Davies - Bouldin	No. of Clusters	Clustering Cutoff	Silhouette	Calinski - Harabasz	Davies - Bouldin	No. of Clusters	Clustering Cutoff	Silhouette	Calinski - Harabasz	Davies - Bouldin	No. of Clusters
1.1	0.57068	4.0017	6.9754	6	1.1	0.57727	4.2584	17.926	5	1.1	0.58533	4.0966	17.994	5
1	0.57068	4.0017	6.9754	6	1	0.56951	7.0524	8.1108	6	1	0.5769	8.1856	4.4816	7
0.9	0.57068	4.0017	6.9754	6	0.9	0.56951	7.0524	8.1108	6	0.9	0.5769	8.1856	4.4816	7
0.8	0.57068	4.0017	6.9754	6	0.8	0.5756	8.2463	4.4822	7	0.8	0.5769	8.1856	4.4816	7
0.5	0.57256	7.1594	3.4479	14	0.5	0.62065	7.5228	2.0096	14	0.5	0.57158	6.5967	1.9938	14

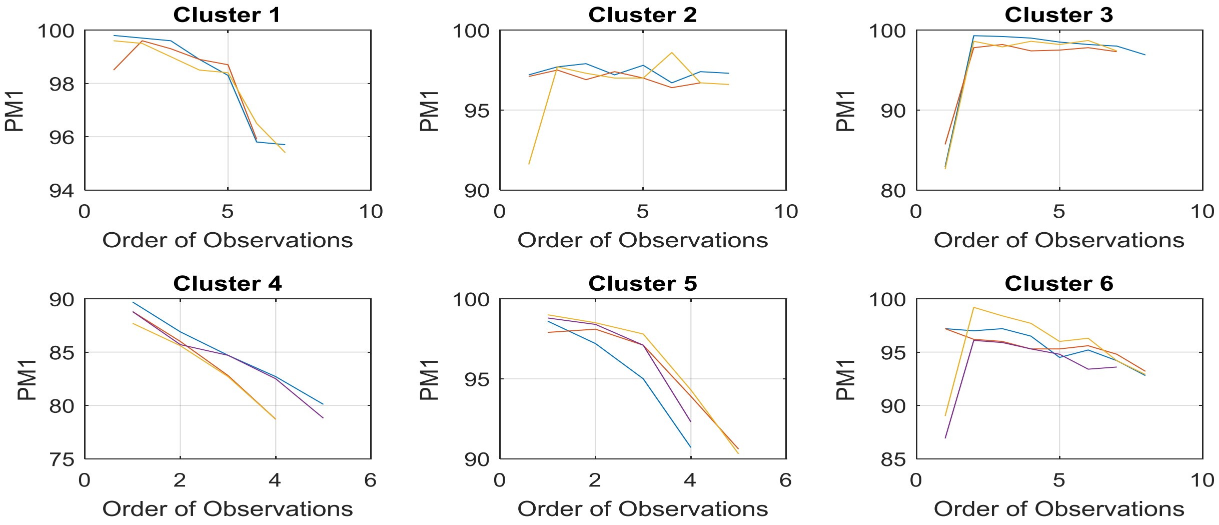


Figure 5.7 DTW clustering results.

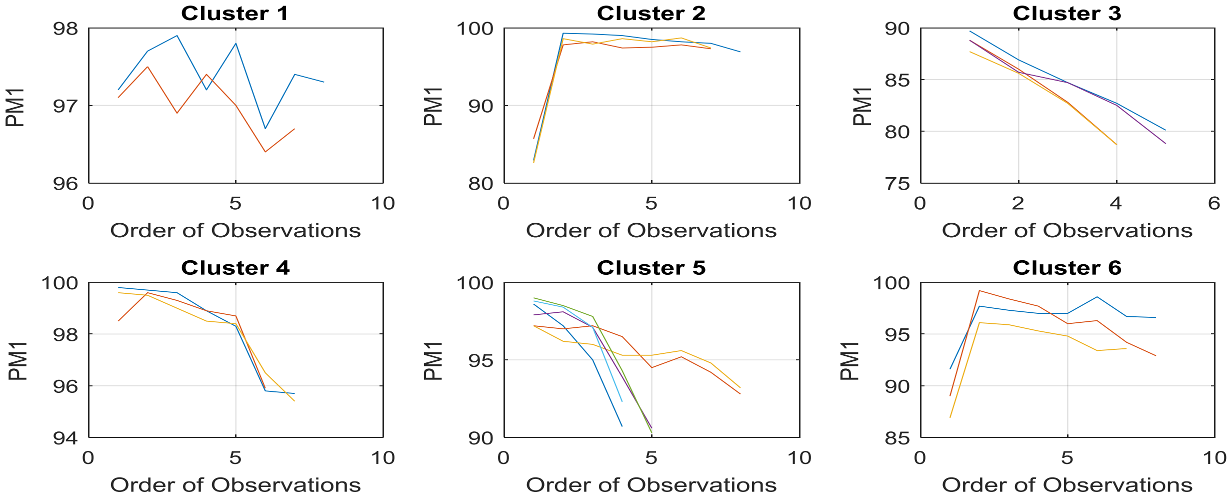


Figure 5.8 DTW with alignment clustering results.

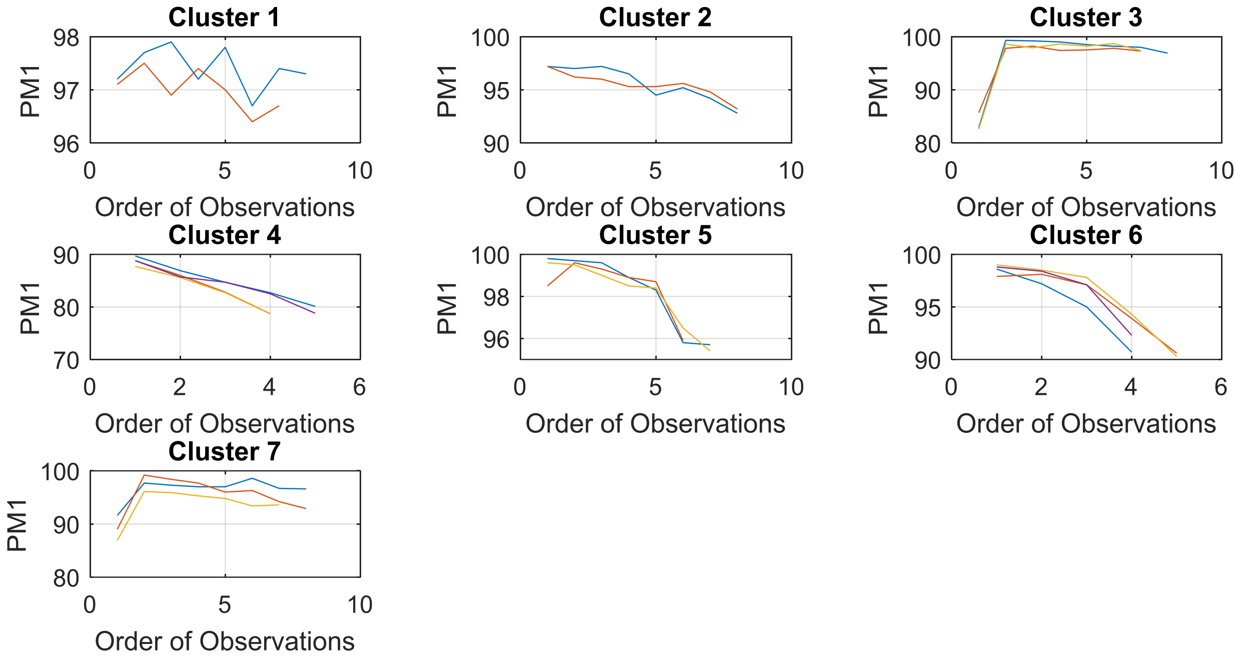
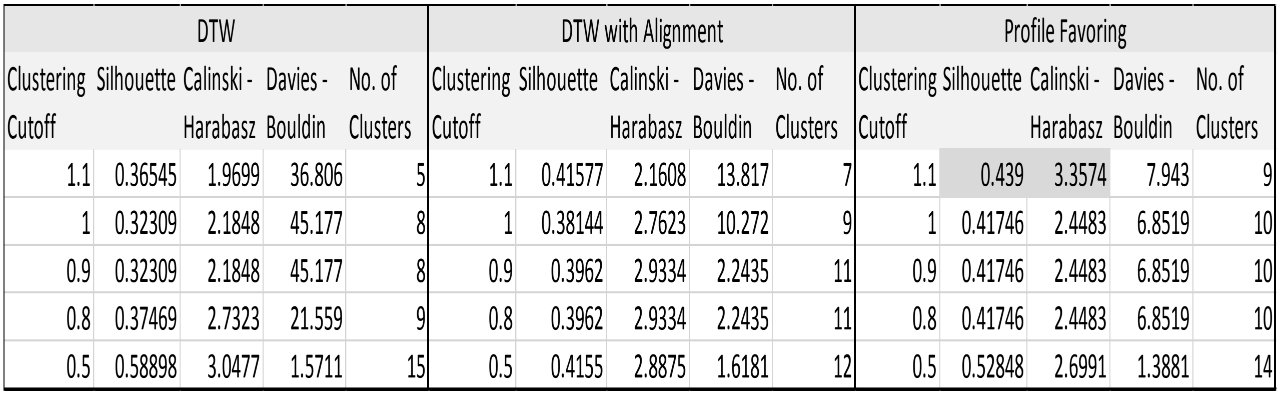


Figure 5.9 Profile favoring DTW clustering results.

This is a VLMTS problem, which is challenging to many time-series clustering techniques that cannot deal with variable lengths or multivariate observations together. Again, the overall performance of the profile favoring DTW is better than the other distance measures and the number of clusters is almost the same for different cutoffs, and therefore, the validity indices. This indicates well-separated clusters in the profile favoring DTW distance space.

Table 5.3 MTS data cluster validity summary of results.



DTW					DTW with Alignment					Profile Favoring				
Clustering Cutoff	Silhouette	Calinski - Harabasz	Davies - Bouldin	No. of Clusters	Clustering Cutoff	Silhouette	Calinski - Harabasz	Davies - Bouldin	No. of Clusters	Clustering Cutoff	Silhouette	Calinski - Harabasz	Davies - Bouldin	No. of Clusters
1.1	0.36545	1.9699	36.806	5	1.1	0.41577	2.1608	13.817	7	1.1	0.439	3.3574	7.943	9
1	0.32309	2.1848	45.177	8	1	0.38144	2.7623	10.272	9	1	0.41746	2.4483	6.8519	10
0.9	0.32309	2.1848	45.177	8	0.9	0.3962	2.9334	2.2435	11	0.9	0.41746	2.4483	6.8519	10
0.8	0.37469	2.7323	21.559	9	0.8	0.3962	2.9334	2.2435	11	0.8	0.41746	2.4483	6.8519	10
0.5	0.58898	3.0477	1.5711	15	0.5	0.4155	2.8875	1.6181	12	0.5	0.52848	2.6991	1.3881	14

5.6 Conclusion

In this paper, a new Dynamic Time Warping (DTW) distance for time-series matching is proposed. The proposed method targets applications that need to identify physical assets with a similar profile over time. The goal is to identify time-series that have the same profile and not only those that have closer magnitudes at different observations.

The proposed distance is used with unsupervised clustering methods, namely hierarchal clustering, visualization and dimensionality reduction methods such as kernel PCA. The performance of the proposed measure is evaluated using three cluster validity indices adopted for the time-series classification task as well as visualization.

The proposed profile favoring DTW is shown to outperform the other variants of the DTW in all of the assessment methodologies used; namely, the validity indices and the KPCA visualization. Moreover, it does not require any interaction from the analyst or parameter learning from the data since it is not required to tune any parameters, which makes it an easy to use analysis tool.

The proposed clustering method is applied to real degradation data on pavement using single and multiple performance metrics to maintain the condition of the pavement. The performance of the proposed method shows superiority to other classical DTW variants. The performance is evaluated using a vote between three cluster validity indices, as well as a visualization of the results.

The proposed method assumes the observations of the time-series are collected at equal time intervals. In some cases, the sampling intervals of the observations are not equal. In this case, it is more representative to calculate the derivative with respect to time instead of using the difference between observations, as this will take the variable rate of change of the series into account.

Table 5.4 Summary of Potentials and Limitations of PFDTW

Proposed Technique	Profile Favoring Dynamic Time Warping
Rational, Need	Cluster sequences of variable length observations based on trends, each of which can have multiple factors
Potential in industry	Scheduling for assets' maintenance based on the type of degradation
Benchmarks used	Transport Quebec road performance data
Average performance improvement against benchmark	Better cluster validity indicators as well as better representation of the variable length time-series in fixed 2D spaces.
Scalability of the algorithm	The algorithm can be used for any amount of data and number of sensor readings
Computational Complexity	The computational complexity during retrieval is $\mathcal{O}(T^2N^2)$ where T is the average time-series length, and N Is the number of training samples.
Extendibility to other potential applications	Can be used for grouping records of time-series data into groups having similar trends
Inputs to the algorithm	Multivariate time-series data with variable lengths with no specific labels
Output from the algorithm	Either using the distance measure within a clustering algorithm to group time-series with similar trends, or as a kernel within PCA to project the variable time series data into 2D fixed space.
Parameters to set / decided upon	The parameters required by the clustering technique it is used along with such as the number of clusters or cut-off threshold if used with hierarchal clustering.

CHAPTER 6 ARTICLE 4: INTERPRETABLE CLUSTERING FOR RULE EXTRACTION AND ANOMALY DETECTION: A NEW UNSUPERVISED RULE INDUCTION TECHNIQUE

Ahmed Elsheikh; Soumaya Yacout; Mohamed Salah Ouali

Submitted to: Expert Systems with Applications

Abstract

This paper proposes a modified unsupervised divisive clustering technique to achieve interpretable clustering. The technique is inspired by the supervised rule-induction techniques such as Logical Analysis of Data (LAD), which is a data mining supervised methodology that is based on Boolean analysis. Each cluster is identified by a simple set of Boolean conditions in terms of the input features. The interpretability comes as a by-product when assuming that the boundaries between different clusters are parallel to the features' coordinates. This allows the clusters to be identified in Disjunctive Normal Form (DNF) logic where each Boolean variable indicates greater or less than conditions on each of the features. The proposed technique mitigates some of the limitations of other existing techniques in terms of the parameters that require tuning, and the comprehensibility of the resulting clusters. Moreover, the clusters identified by the proposed technique are adopted to solve anomaly detection problems. The performance of the proposed technique is compared to other well-known clustering techniques such as the DBSCAN and Density Peaks, and the extracted rules are compared to decision trees which is a supervised rule-induction technique. The results show that its performance is 20% on average higher in accuracy across 11 UCI benchmark datasets to the most known clustering techniques, and its interpretability is comparable to rules extracted by supervised techniques.

6.1 Introduction

Identifying data of similar nature is one of the most important topics in data mining, and clustering is one of its well-known tools. Grouping data into clusters requires a measure of dissimilarity between any two observations. It also requires a measure of the scatter, which is the amount of dissimilarity within a given group of data, and between different groups. The target of clustering is to find groups of observations that minimize the scatter within a single group of observations and maximize it between different ones (Everitt et al. 2011).

There are numerous techniques, throughout the literature, to perform clustering. These techniques can be grouped into five main categories (Kantardzic 2011):

A. Partitional clustering: The algorithms of this category divide the observation space into a predefined number of clusters. These algorithms iteratively group the observations into their nearest cluster center, and then moves the center to minimize a certain objective function, such as the sum of mean squared differences between each center and the observations of its cluster. One of the most used algorithm of this category is the k-means
B. Hierarchal clustering: There are two sub-categories of this category; the agglomerative, which is a bottom-up approach, and the divisive, which is a top-down approach. The idea is to iteratively merge or split into clusters the given observations until a certain stopping criterion is met, for example a given number of clusters or a threshold on the distance between clusters is reached.
C. Density-based clustering: The algorithms of this category grow clusters based on a certain definition of a neighborhood to each observation, for example a hyper-spherical neighborhood of some radius r around each observation. All linked neighbors are considered to be a cluster.
D. Model-based clustering: the algorithms of this category use probability modelling, typically mixture models. Given a certain type of distribution, and a number of mixtures, the clusters are estimated by the Expectation-Maximization (EM) algorithm in an iterative method until the clusters are formed.
E. Neural network clustering: this is a special type of neural networks that is called self-organizing maps. It clusters the data by an iterative procedure that is called competitive learning. The procedure updates the locations of a given number of centroid based on a scaling factor that changes with the distance to the introduced observation, the distance to the neighboring centroids, and the iteration number.

Each of the clustering techniques examined in literature have their strength points and their limitations. These limitations come from the fact that some assumptions must be made about the solution. These assumptions mainly include (Everitt et al. 2011; Kantardzic 2011):

- Assuming certain geometrical shapes for the generated clusters. Examples of such are the portioning that is used by the algorithms of the partitional clustering that is described in A), and the model-based clustering techniques that is described in D). This portioning imposes hyper-spherical shapes in the case of k-means, and hyper-elliptical shapes in the case of Gaussian mixture models, respectively.
- Assuming a certain threshold for proximity between cluster members, beyond which observations are considered to belong to another disjoint cluster. Examples of such techniques are the hierarchal and the density-based clustering techniques, which are described in B) and C), respectively.
- Assuming the values of some parameters that can highly affect the clustering results such as the self-organizing map neural network that is described in E), and which requires defining the learning rate, the neighborhood size and its learning factor, and the decay factor of the learning rate with the number of iterations.

None of these techniques leads to interpretable clusters, that can be easily represented in the form of simple thresholding rules on each feature. This simple representation is very useful to give insights to decision makers and non-specialists involved in any firm to take proper actions, or to anticipate and take precautions from certain events (Chung and Tseng 2012). From a practical point of view, interpretability of the generated patterns helps the practitioner in understanding the physical phenomenon that leads to a specific clustering. It also permits domain experts to verify their assumptions and to discover hidden knowledge which cannot be discovered by human expertise alone. Some examples of supervised rule extraction applications are found in engineering (Shaban, Yacout, and Balazinski 2015), and in medicine (Alexe, Blackstone, and Hammer 2003). A lot of research on rule extraction from other black-box supervised learning approaches throughout literature, some of which can be found in (Barakat and Bradley 2010) for support vector machines, and in (Hailesilassie 2016) for deep neural networks.

The focus in this paper is on finding an unsupervised clustering technique that offers human interpretability to the generated clusters, where each cluster is identified by a simple set of Boolean conditions in terms of the input features. The interpretability comes as a byproduct when assuming that the boundaries between different clusters are parallel to the features' coordinates. This allows the clusters to be identified by a Disjunctive Normal Form (DNF) logic, where each Boolean variable indicates greater or less than conditions on each of the features. This property can be found in some supervised classifiers such as decision trees (Quinlan 1986), and rule induction approaches such as the Logical Analysis of Data (LAD) (E Boros, Hammer, and Ibaraki 2000). Searching for interpretability in the case of unsupervised problems is more challenging since there are no given discriminating target labels.

The interpretation property was studied in the unsupervised domain. Previous researches can be grouped into two categories:

- Clusters that are formed of hyper-rectangles instead of hyper-spheres as in the case of k-means, or by assuming that the data come from a mixture of almost rectangular probability distributions, for example a uniform distribution, in order to allow an overlap between different rectangles (Andrew, Moore, and Pelleg 2001). In this case, the interpretability is attained because the boundaries between different clusters are parallel to the features' coordinates. Consequently, the generated patterns are interpreted as zones of features' values that characterize each cluster. This approach requires defining the number of clusters, an assumption of the probability distribution, and the clustering is not reproducible since iterative fitting methods, such as EM, use random initialization. Hyper-rectangles are used for supervised classification problems and are reviewed in (Hasperué, Lanzarini, and De Giusti 2012).
- Divisive clustering techniques, which start with the complete set of observations as a single cluster, and iteratively split the space into clusters. The most known technique in this category is the bisecting k-means (Steinbach, Karypis, and Kumar 2000) which iteratively splits the cluster with the largest scatter. The scatter is estimated from the diameter of the cluster. The new centers are then found using k-means with $k = 2$, and the Principal Direction Divisive Partitioning (PDDP) (Boley 1998) which splits the direction that has the largest eigenvalue by using Principal Components Analysis (PCA).

The latter two divisive clustering techniques suffer from four main limitations:

(a) The greedy algorithms that are usually used to find a suboptimal solution for a search problem, in a straightforward manner, simply take the best split in terms of the assumption on the scatter, and once a split is made there is no way to go back and consider another split that may result in a better global optimum.

(b) The measures of the quality of split impose certain geometry on the revealed clusters, which restricts the capability of the technique to reveal clusters of arbitrary geometry.

(c) The stopping criterion requires prior knowledge about the number of iterations or the threshold on the splitting criterion, which is usually not available and might require many trials and errors.

(d) The splits are not as simple as parallel-to-the coordinates rules that are used for rule-induction. The number of these splits is not prespecified and grows according to the demands of the data, and hence do not have a certain geometrical distribution to fit to the data. This type of rules can be easily formulated as a combination of thresholds or ranges on different features, which can be easily interpreted by management or non-specialists.

In this paper, the proposed technique for obtaining Interpretable Clustering for Rule Extraction and Anomaly Detection (IC-READ) is a novel approach for divisive clustering. It mitigates the above four limitations that are described in (a) to (d). There are three main advantages to the proposed technique over the previous ones:

1) The technique keeps track of n-best candidates instead of just the best candidate. This means there is not just a single possible path where an error made at any step can largely deviate the solution from the global optimal solution. The greedy algorithm is a special case of the n-best search with $n = 1$. In this way, the proposed technique mitigates problem (a).
2) The technique saves exploring many unnecessary candidate splits by identifying only low-density regions as possible candidate splits. This mitigates problem (b).
3) The technique has an automatic selection capability for the best number of clusters without interference from the analyst. This property mitigates problems (c) and (d).

Moreover, the technique can be used for one-class classification tasks, which is used for anomaly detection. This is presented in detail in section 6.4.

The proposed technique is inspired by the general supervised LAD rule induction technique (E Boros, Hammer, and Ibaraki 2000). LAD has two main steps of concern:

- Binarization: this step converts the given data into binary format by finding cut-points. The cut-points in LAD are identified by transition in the labels of the given data from one class to another. In the case of unsupervised learning there are no given labels, so the proposed binarization identifies candidate cut-point through the change in density for every feature and this binarization changes for each split to benefit from the previous split. This step is discussed with further details in section 6.1.

- Pattern generation: this is done in LAD by using an enumeration technique (Endre Boros et al. 2011) and the patterns are chosen by order of covered observations. In the case of the proposed IC-READ a measure of clustering quality is used to choose among the candidate patterns. The clustering quality measure in general measures how close observations of the same cluster are together and how well are they separated from other clusters. This step is discussed with further details in section 6.2.

LAD stops looking for patterns when the observations of each class is completely covered by the generated patterns. For the case of unsupervised IC-READ, a cluster validity measure is chosen as the stopping criteria for the splitting procedure. This is explained with more details in section 6.2.

The rest of the paper is organized as follows. Section 6.2 introduces the proposed IC-READ technique explaining how the rules are extracted and the clusters are formulated. Section 6.3 shows experimental results for some benchmark datasets and compares the performance to other commonly used classifiers. Section 6.4 illustrates how to use the proposed technique to do anomaly detection, and compares its performance to another commonly used anomaly detector.

6.2 Interpretable Clustering for Rule Extraction and Anomaly Detection (IC-READ)

The proposed clustering technique follows a divisive approach, which is in the category of hierarchical clustering that was presented in point (B), in section 1. The hierarchy is formed by iteratively splitting on one of the feature dimensions in a top-down approach. This type of divisive clustering is called monothetic (Everitt et al. 2011). The proposed approach is composed of three phases; identify the split candidates, select the best-split, check validity of the clusters identified.

6.2.1 Identifying Split Candidates

Since the clustering problem has no specific target labels to split upon, a common approach that exists in the literature is to put a cut-point between every two successive observations and to consider it as a candidate split as in (Chavent, Lechevallier, and Briant 2007). This is an exhaustive search procedure, and even for supervised rule induction where the targets are known, this search poses challenges (Thabtah, Qabajeh, and Chiclana 2016). Our proposed approach decreases this huge number of candidate cut-points by calculating the histogram of each dimension and identify the candidate cut-points to be the midpoints of regions of low density located between high densities and consider those only as candidates. The reason for this is that low-density regions are considered as candidate regions to be intra-cluster, while high-density regions are inter-cluster. This is the basic assumption of the proposed technique.

Unlike supervised rule-induction techniques that identify all candidate splits for all features all at once, splitting on the marginal histogram of each feature dimension can get obscured by the correlations found at higher dimensions. This might limit the ability of the clustering technique to find good clusters. The proposed approach is an adaptive procedure which iteratively identifies a set of new candidate cut-points after each divisive step in the proposed clustering technique.

The example given in Figure 6.1 shows a comparison between the all-at-once-split approach and our one split-at-a-time approach. Figure 6.1 illustrates this problem for a synthetic two-dimensional set of observations. The observation space shown in Figure 6.1 at the top, along with the marginal histogram of each feature dimension.

The marginal histogram of the first feature, x_1 , fails to identify the best low-density regions around $x_1 = 0.5$ between the top two clusters, and at $x_2 = 2$ between the two bottom clusters, respectively. This marginal histogram of $x_1$identifies a low-density region around $x_1 = 1$, because it is obscured by the correlation between x_1, and x_2. Nevertheless, the correct low-density regions can be identified after the first split on $x_2 = 0.5$, since each of the top and the bottom clusters will be separated, and their marginal distributions will have a clearer view in the x_1 feature dimension.

The clusters identified using candidate splits directly from the raw observation space are shown in Figure 6.1 in the middle, where some of the observations of the upper and lower clusters are wrongfully grouped because the split procedure is non-adaptive and the split in which all the low densities once, and uses them for later splitting. This is a consequence of having the low density at the vertical split comes at $x_1 = 1$, which is the lowest density point from the perspective of the marginal histogram of the raw data for x_1, and the low density at the horizontal split at $x_2 = 0.5$.

The clusters identified using the adaptive candidate split selection procedure, which will be explained in the following section, are shown in Figure 6.1 bottom. This split shows almost perfect

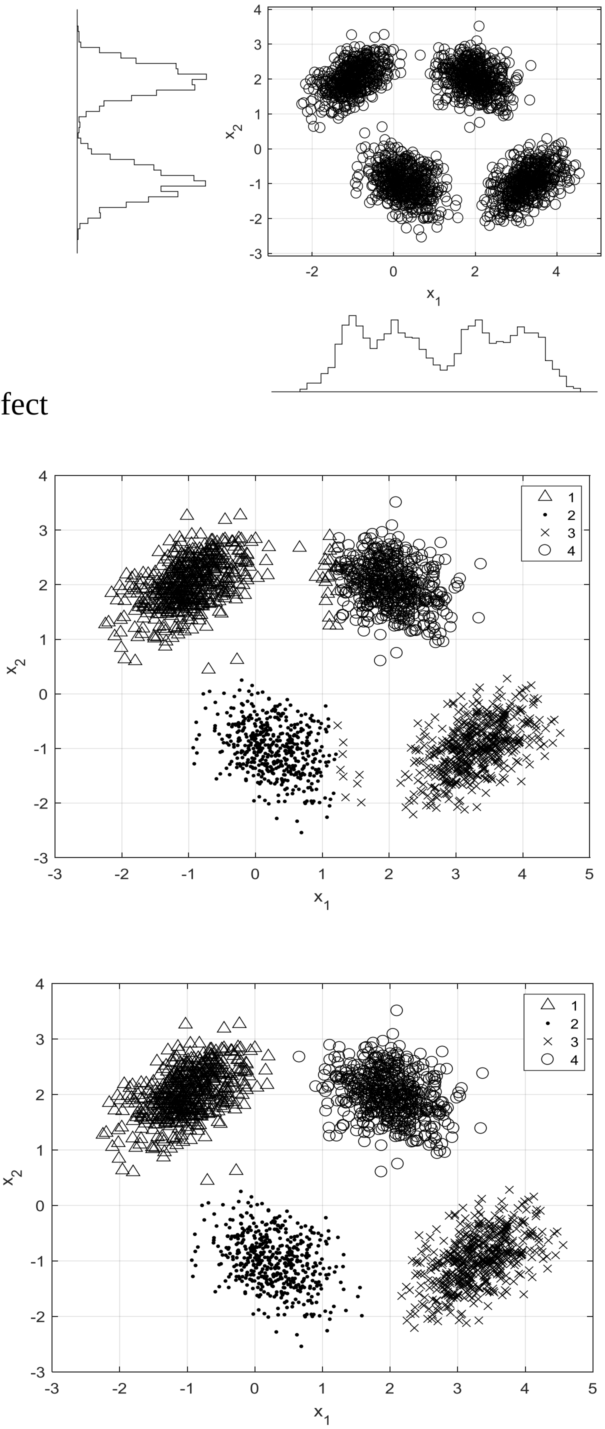


Figure 6.1 (Top) The raw observation space and marginal histograms, (Middle) clustering using candidate splits from the original raw observation space, and (Bottom) clusters using the adaptive splitting procedure.

clustering since the candidate splits are reexamined after each split, so the low-density region for the top two clusters is now $x_1 = 0.3835$, and for the bottom two $x_1 = 1.7105$.

The cut-points are calculated to be the midpoints of the low-density regions, which are the midpoints of the marginal histogram bins having smaller count. The minima in the marginal histograms are identified by a peak-picking procedure, which is a search for the points that have higher values than those next to it, or in other words, the local maxima, from the negative of the histogram. The midpoints are calculated as follows:

$$CP_{iF_j} = \frac{h(i) + h(i+1)}{2}, \qquad i \subset M, \qquad 1 \le j \le N_F \qquad 35$$

Where CP_{iF_j} is the i^{th} candidate cut-point for the j^{th} feature F_j, while i is the index of the left edge of a bin in the histogram h, $h(i)$ is the value of the i^{th} bin of the histogram, M is the set of local minima identified, and N_F is the number of features in the data.

The output from this step is a binarization of each observation, where each attribute indicates whether the corresponding feature exceeds a certain value or not.

$$b_{iF_j} = \begin{cases} 1 \ \ if \ \ F_j \ge CP_{iF_j} \\ 0 \ \ if \ \ F_j < CP_{iF_j} \end{cases} \qquad 36$$

Where b_{iF_j} is the i^{th} binary attribute of the j^{th} feature F.

6.2.2 Selecting the Best Split

The proposed algorithm follows a beam search procedure. It is a compromise between a greedy algorithm following only the best split at each step, which has the drawback that it can deviate from near optimal solution, and a full search of all possible combinations of the candidate splits, which can be very time consuming and computationally exhaustive.

At this point in the proposed technique, the data is in a binary form as a result of identifying the candidate cut-points of each feature dimension. The next step is to choose the n-best splits at each level in the hierarchy. As n increases, the number of possibilities being explored increase, and better clusters can be found if any exists. The choice of n is based on the expected complexity of the boundaries between clusters. Now binarized data minimizes the search cost to simple logical

AND operations between the clusters identified at the previous phase and the new ones at the current phase to identify the observations within a certain subspace. Hence, there is no need to do comparison operations for each candidate split nor there is a need to consider each combination of cut-points explicitly.

The ordering of the candidate splits to identify the n-best is based on a quality measure. This quality measure identifies how compact the current clusters are relative to their separation from each other. There are many measures that can be used for such a task. Cluster validity indices are chosen to be the quality measure for the proposed technique since they do not assume a certain distribution or geometric form on the extracted clusters. These are presented in the next sub-section.

6.2.2.1 Cluster Validity Indices

The candidate clusters at any level of the clustering procedure will get a better validity score when the similarity between the members of a single cluster is much more than that of between members of other clusters. These cluster validity indices can also work as split quality measures as suggested by (Mazzeo, Masciari, and Zaniolo 2017).

The holistic measure of inter-cluster and intra-cluster similarities is called scatter. There is no clear definition of the scatter within and between clusters. According to (Arbelaitz et al. 2013; Everitt et al. 2011) there is no single reliable validation index, hence we use the top three performing indices as suggested by (Arbelaitz et al. 2013).

6.2.2.1.1 Modified Davies–Bouldin (DB)*

This measure calculates the ratio of the average distance between observations in a cluster and its centroid which is the scatter within a given cluster, to the distance between cluster centroids which represents the scatter between clusters (Kim and Ramakrishna 2005). For a given distance measure $d(x, y)$ between any two observations, $C = \{c_1, \dots, c_k\}$ a set of k clusters for a total of N observations, and each cluster has members x_i with a count of N_{c_i} and mean $\bar{c_i}$ then the DB* is defined as follows:

$$DB_{mod}(C) = \frac{1}{N_{c_i}} \sum_{c_i \in C} \frac{\max\limits_{c_j \in C \setminus c_i} \{\sigma(c_i) + \sigma(c_j)\}}{\min\limits_{c_j \in C \setminus c_i} \{d(\bar{c_i}, \bar{c_j})\}} \qquad 37$$

Where $\sigma(c_i) = \frac{1}{N_{c_i}} \sum_{x_i \in c_i} d(x_i, \bar{c}_i)$. A small value of this index indicates a more precise clustering.

6.2.2.1.2 Calinski–Harabasz (CH)

This index calculates the ratio of the average distance between cluster centroids around the global observations' mean which represents the scatter between clusters, to the average distance between observations of each cluster and their centroid which is the scatter within each cluster (Caliński and Harabasz 1974). The formulation is as follows:

$$CH(C) = \frac{N-k}{k-1} \frac{\sum_{c_i \in C} N_{c_i} d(\bar{c}_i, \mu)}{\sum_{c_i \in C} \sum_{x_i \in c_i} d(x_i, \bar{c}_i)} \qquad 38$$

Where μ is the global mean of all the observations. A large value of this index indicates a more precise clustering.

Another merit for using the cluster validity indices is that the proposed technique will have a built-in automatic stopping criterion. Any clustering procedure can either be stopped after reaching a predefined number of clusters, by adding a certain threshold on maximum separation between members of the same cluster, by adding a threshold on the minimum separation between clusters, or by evaluation using a validity index through trial and error. Using a predefined number of clusters, or adding thresholds requires prior knowledge about the data at hand, which is rarely available.

6.2.3 Rule Set Identification

This is the main goal of the proposed clustering, which is to produce a set of logical rules to identify different groups in the data given by the user.

After either the required number of clusters by the user is reached or by finding the best clustering measure during the search procedure explained in the previous section. The value of the found cluster validity index will be output to the user.

Each cluster is identified by a combination of binary attributes formed at each step of the splitting procedure. Each element of theses combinations can either be any of the binary attributes or their negations. These combinations are the patterns that identify each cluster, and they can be used to classify any new future observations to their corresponding cluster.

6.3 Benchmark Dataset Examples

In this section, the performance of the proposed clustering technique is compared with other conventional clustering techniques, namely the k-means and agglomerative clustering. The experiments are conducted on several benchmark datasets from the UCI repository (Lichman 2013). The dataset description is shown in Table 6.1. The classes of the glass dataset are reduced to only two which are window glass and non-window glass.

Table 6.1 Benchmark UCI datasets' discribtion.

Dataset	Number of Attributes	Number of Classes
Iris	4	3
Wine	13	3
Seeds	7	3
Glass	9	2
Banknote Authentication	4	2
Mice Treatment	69	2
Mice Shock	69	2

The proposed technique will be termed as IC-READ-CH or IC-READ-DB* when using the CH index or the DB* index respectively.

6.3.1 Experimental Setup

Since the ground truth labels are available for these datasets, data partition comparison indices can be used. The indices used for the comparison are the purity index, the adjusted Rand index, and the F-measure, which are defined in equations 39 through 41.

If the set of cluster labels is $C = \{c_1, \dots, c_k\}$, and the set of class labels is $L = \{l_1, \dots, l_k\}$, where k is the number of clusters, then the partition comparison indicies can be defined as follows:

$$Purity = \frac{1}{n}\sum_i \max_j \left|l_i \cap c_j\right| \qquad 39$$

Where n is the number of observations, and $1 \leq i, j \leq k$ are the indices of the class labels, and cluster labels respectively. The overlap between every pair of ground truth labels, and predicted clusters is calculated at first, then the pairs are selected according to the number of overlapping examples without replacement.

The Adjusted Rand Index (ARI) is a modification to the original Rand index, which is simply the purity index, to produce more stable results by assuming a hyper-geometric distribution on the labels (Hubert and Arabie 1985). The ARI is formulated is as follows

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}} \qquad 40$$

Where $n_{ij} = \left|l_i \cap c_j\right|$, which accounts for all possible intersections between the cluster labels and the ground truth labels, and their marginal sums $a_i = \sum_j n_{ij}$, and $b_i = \sum_i n_{ij}$, which account for the total number of actual members of each class, and the number of predicted members of each class respectively. The $\binom{n}{r}$ notation is equivalent to n combination r. Higher values of this index indicate better results.

Another famous partition comparison index is the F-measure, which is a weighted average of the precision and recall (Bishop 2006), and can be defined as

$$Fmeasure = (1 + \alpha)\frac{precision * recall}{\alpha * precision + recall} \qquad 41$$

Where the $precision = \frac{TP}{TP+FP}$, the $recall = \frac{TP}{TP+FN}$, such that TP is the number of true positives, FP is the number of false positives, FN is the number of false negatives, and α is the factor which emphasizes the importance of recall with respect to precision. α is chosen to be 1 for this comparison, which is most commonly used and indicates equal importance.

ResultsTable 6.2 shows the comparison results between the proposed technique, k-means, and agglomerative clustering using Euclidean distance (Everitt et al. 2011). In these experiments, the number of clusters was set to be the same as the actual ground truth to be able to make a fair comparison with the other techniques that cannot estimate the number of clusters automatically. For the k-means clustering technique, the clustering was rerun with three different initializations, and the one with the least sum of squared errors between the evaluated means and the observations was selected. A measure indicated by 0 means that the clustering algorithm resulted in an empty cluster. The proposed technique shows superiority to the other two techniques.

Without guidance, the IC-READ-CH correctly predicts the number of classes for all datasets except for the case of the wine dataset which is predicted as 6 instead of 3, and the seeds dataset which is predicted as 2 instead of 3.

The set of rules induced by the IC-READ-CH are compared to that induced by decision trees in Table 6.3. The rule coverage identifies the percentage of observations covered by that rule. This can help qualify the effectiveness of the rules induced by the IC-READ-CH versus those extracted by a supervised learning procedure such as the decision tree. It is important to note that the more the classes are overlapped and non-linearly separable, the decision tree will branch into many rules to cover all of the observations of a given class, while the IC-READ-CH will not be able to guess the correct rules without labels. This fact is clear from the results shown in Table 6.3 Comparison of the induced rules by the IC-READ-CH and the supervised decision trees.

In the case of the Iris dataset, the classification rule for class 1 is almost the same for both. Also, the rule for class 2 is very similar except for the condition on x_4 which only separates 4.67% of the data to class 3 from class 2. Hence, the IC-READ-CH provides a good generalization for the classification rules of the Iris dataset.

A similar argument goes for the glass dataset, where only 6.54% of the data are separated using extra conditions on x_1 and x_9, while a single condition on x_3 correctly separates the rest of the data. Again, the IC-READ-CH provides a good generalization for the classification rules.

Table 6.2 Performance comparison between different clustering techniques on benchmark UCI datasets.

Dataset / Partition index	Technique	Purity	ARI	F-measure
Iris	k-means	0.893	0.73	0.818
	AHC	0.68	0.564	0.077
	IC-READ-CH	0.893	0.732	0.8095
	IC-READ-DB*	0.927	0.8	0.8913
Glass	k-means	0.897	0.6	0.784
	AHC	0.771	0.0407	0.075
	IC-READ-CH	0.907	0.635	0.808
	IC-READ-DB*	0.762	0	0
Wine	k-means	0.702	0.371	0.5272
	AHC	0.433	0.006	0
	IC-READ-CH	0.708	0.371	0.5487
	IC-READ-DB*	0.691	0.365	0.5
Banknote Authentication	k-means	0.612	0.049	0.504
	AHC	0.555	0	0
	IC-READ-CH	0.641	0.078	0.592
	IC-READ-DB*	0.641	0.078	0.592
Seeds	k-means	0.895	0.71	0.845
	AHC	0.371	0.002	0.056
	IC-READ-CH	0.862	0.651	0.76
	IC-READ-DB*	0.667	0.454	0
Mice Treatment	k-means	0.558	0.013	0.538
	AHC	0.531	0.00067	0.012
	IC-READ-CH	0.622	0.059	0.62
	IC-READ-DB*	0.531	0.00067	0.012
Mice Shock	k-means	0.575	0.022	0.568
	AHC	0.514	0	0
	IC-READ-CH	0.706	0.1698	0.7
	IC-READ-DB*	0.514	0	0

Table 6.3 Comparison of the induced rules by the IC-READ-CH and the supervised decision trees.

Iris dataset				
Decision tree rules	Class label	Rule coverage	IC-READ-CH rules	Cluster Label
x3 < 2.45	1	33.33	x3 < 2.11	1
x3 > 2.45 & x4 > 1.75	3	28.67	x3 > 2.11 & x3 < 5.1975	2
x3 > 2.45 & x4 < 1.75 & x3 > 4.95	3	4	x3 > 5.1975	3
x3 > 2.45 & x4 < 1.75 & x3 < 4.95 & x4 > 1.65	3	0.67		
x3 > 2.45 & x4 < 1.75 & x3 < 4.95 & x4 < 1.65	2	33.33		
Glass dataset				
Decision tree rules	Class label	Rule coverage	IC-READ-CH rules	Cluster Label
x3 < 2.695 & x1 > 1.5422	2	3.27	x3 < 1.9775	1
x3 < 2.695 & x1 < 1.5422 & x9 > 0.13	2	2.34	x3 > 1.9775	2
x3 < 2.695 & x1 < 1.5422 & x9 < 0.13	1	22.9		
x3 > 2.695 & x6 > 1.28	1	0.93		
x3 > 2.695 & x6 < 1.28	2	70.56		

6.4 IC-READ Beyond Clustering

Another merit of IC-READ is to identify modes, or states within a given class. An illustrative example is shown in Figure 6.2. Any rule-induction technique such as LAD can easily identify a classification rule, such as $x_2 > 0.3$ to separate both classes shown in Figure 6.2, but it cannot identify different modes within each class since they have the same label. This property of identifying rules can be very useful in applications such as machine diagnosis, where a working machine can have multiple normal operating conditions. This is depicted in the illustrative example of Figure 6.2, where class 1 has two sub-clusters, which can indicate 2 operating conditions for class 1.

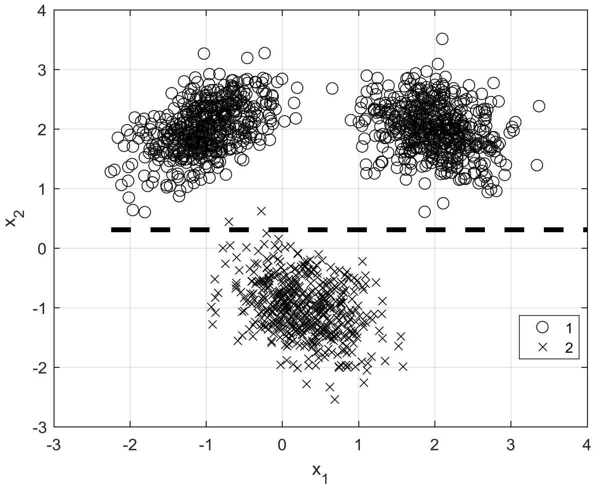


Figure 6.2 Modes of operation within a single class.

There are two applications for IC-READ along with supervised rule-induction:

1) Operation mode Identification for each class, or more generally sub-classes.
2) One-Class Classification (OCC) or anomaly detection where there is a lot of data from a certain class, for example like normal operation of a machine, and very few or no fault data.

6.5 IC-READ for Anomaly Detection

Since the proposed clustering algorithm can identify subspaces where there are groups of similar observations, then anomalies can be considered as observations apart from the similar groups. Instead of specifying a threshold on the distance between a group center and the tested observation, a cluster tightening procedure is adopted.

For each of the identified clusters, a bounding hyper-rectangle is bound to it from each feature dimension leaving out low density regions that have density lower than a certain percentage of the maximum density in that dimension. Figure 6.3 shows such hyper-rectangles for a 2D case using anomalous fraction of 0.1. This percentage is called the anomalous fraction, which is similar to the ν parameter in the popular One-Class Classification Support Vector Machine (OCCSVM) (A.G., Abdulla, and Asharaf 2017; Schölkopf et al. 2000), which is a rough estimation of the anticipated anomalies in the data.

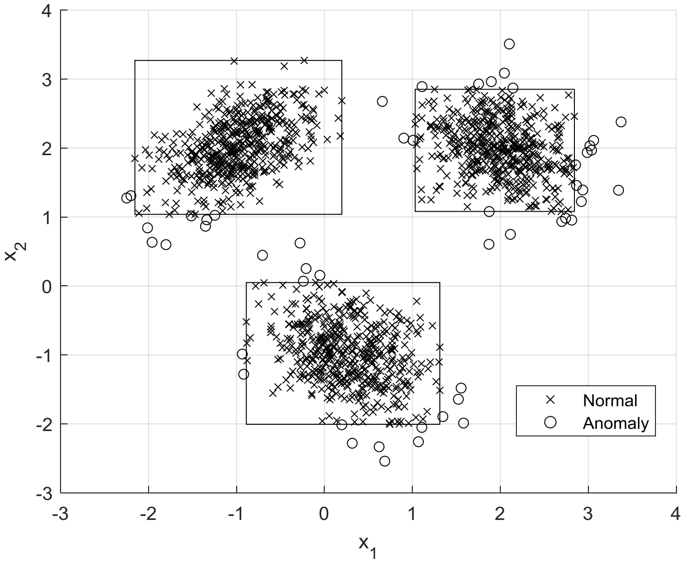


Figure 6.3 Example of 2D anomaly detection using IC-READ-CH using hyper-rectangle tightening with anomalous fraction 0.1.

6.6 Experimental Setup

A benchmark shuttle dataset from the NASA StatLog database found in the UCI repository (Lichman 2013), where 80% of the data are from one class, and the other classes are considered to be anomalies. This dataset has 9 feature dimensions and 46464 observations.

Another important industrial application for anomaly detection is bearing vibration analysis, where the data collected throughout the lifetime of a bearing is in normal operation, and very few faulty operation data is available, or not even yet collected. The experiments in this section are conducted on the vibrational data from the bearing dataset provided by the Center on Intelligent Maintenance Systems (IMS), University of Cincinnati (Lee et al. 2007). The setup consists of four bearings installed on a shaft with two accelerometers in each. Each accelerometer collects vibration signals in a different axe than the other. An AC motor drives the shaft at 2000 rpm along with a 6000 lb radial load applied the shaft and bearings by a spring mechanism. The sensors' data are 1 second snapshots that were collected every 10 min for 164 h with a sampling rate of 20 kHz, such that each snapshot is written in a separate file. This is a test-to-failure experiment, and this section focuses on identifying the outer race defect of bearing 1.

For this experiment, five representative features were extracted from the vibration signals to decrease the redundancy of the available data. Each file contains 20480 samples, features were extracted from windows of 200 samples with a 50% overlap as shown in Figure 6.4. Data files considered were one out of every five files in a sequence, which means that the analysis was done once every 50 minutes. All readings during the last 120 minutes were considered as failure. The features are extracted from the Power Spectral Density (PSD) of the analyzed window. The features are the maximum and median PSD values and their corresponding frequency, along with the average frequency weighted by the PSD.

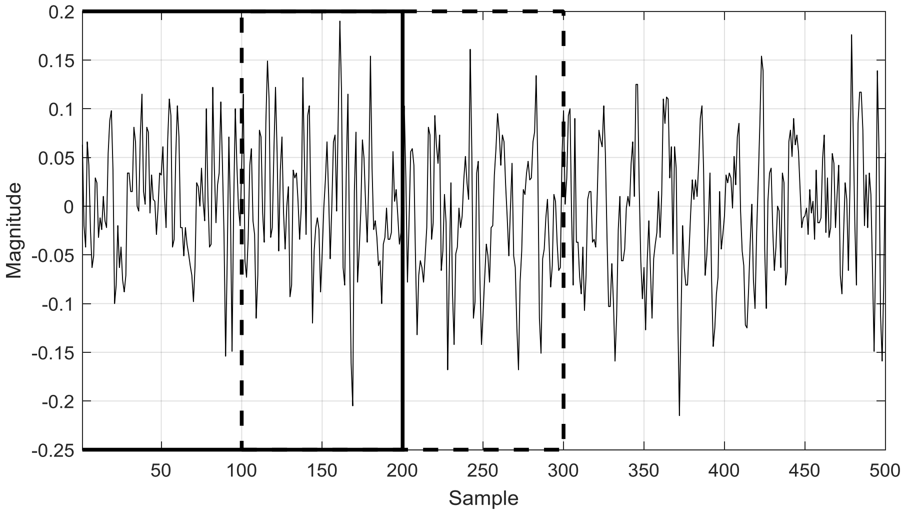


Figure 6.4 Widowing procedure used for feature extraction.

6.7 Results

A comparison between the results of the OCCSVM using a radial basis function kernel, and the IC-READ-CH for the shuttle dataset is shown in Table 6.4, and the for the bearing dataset in Table 6.5 for different anomalous fractions. Two performance measures are used, the detection rate, which is the fraction of correctly identified anomalies, and the false alarm rate which is the fraction of normal observations identified as anomalies. Of course, the best performance is achieved when the detection rate is maximum, and the false alarm rate is minimum.

It is identifiable that the IC-READ-CH achieves higher detection rates using very low anomalous fractions for both datasets, which is a great merit since it will require lesser tuning compared with the OCCSVM where the anomalous fraction is not of the same range in both applications, as well as the other parameters that require tuning such as the kernel type and its internal factors.

Table 6.4 Shuttle dataset performance comparison for different anomalous fraction values.

Shuttle Dataset	IC-READ-CH		OCCSVM	
Anomalous Fraction	Detection rate	False alarm rate	Detection rate	False alarm rate
0.005	0.916	0.012	0.369	0.001
0.008	1	0.021	0.542	0.0025
0.011	1	0.034	0.749	0.003
0.014	1	0.042	0.901	0.005
0.017	1	0.047	1	0.007
0.02	1	0.056	1	0.01

Table 6.5 Bearing dataset performance comparison for different anomalous fraction values.

Bearing Dataset	IC-READ-CH		OCCSVM	
Anomalous Fraction	Detection rate	False alarm rate	Detection rate	False alarm rate
0.005	0.992	0.018	0.069	0.0038
0.008	0.992	0.026	0.204	0.0042
0.011	0.992	0.029	0.342	0.0046
0.014	0.992	0.042	0.494	0.0047
0.017	0.992	0.046	0.646	0.0048
0.02	0.992	0.054	0.76	0.006

In real scenarios of this type of application, there is usually insufficient data of anomalous operation. Moreover, the anomalies usually do not have a certain form, they can be unexpectedly of any different nature than the normal case. Hence, the ability of the anomaly detection technique to have very few tuning required is beneficial. It is noticeable that the false alarm rate of the IC-READ is higher than OCCSVM, but it is always safer to have a higher detection rate than having lower false alarm rate. Depending on the application the slightly higher false alarm rate is well tolerable.

Given the data provided by the user and a rough estimation of the anomalous fraction, the output of the proposed technique is the ranges of normal operation identified for each feature dimension similar to that shown in Table 6.6. Any future observation that falls outside the limits of any of these normal operating regions is considered to be anomalous.

Table 6.6 Ranges of normal operation on each feature identified by the IC-READ-CH anomaly detection procedure.

Shuttle dataset	
Minimum	Maximum
[37 , -1 , 77 , -10 , 28 , 6339 , 4 , -353 , -356]	[79 , 3 , 118 , 1751 , 436 , 15164 , 73 , 48 , 8]
[37 , -28 , 74 , -20 , -4 , -30 , 22 , 22 , 0]	[59 , 33 , 111 , 17 , 60 , 47 , 72 , 90 , 42]
[37 , -1 , 77 , -1 , -188 , -12809 , 4 , 240 , 196]	[78 , 0 , 81 , 0 , -160 , -8392 , 43 , 270 , 266]
[37 , -386 , 75 , -3 , -40 , 1327 , 1 , -258 , -298]	[101 , 0 , 102 , 0 , 336 , 4910 , 42 , 123 , 120]
Bearing dataset	
Minimum	Maximum
[55.55 , 3.26e-06 , 7.29e-08 , 4560 , 0]	[561.38 , 5.50e-05 , 6.01e-07 , 7280 , 1920]
[52.06 , 2.87e-06 , 8.30e-08 , 0 , 0]	[584.27 , 3.75e-05 , 5.81e-07 , 4480 , 1920]
[98.46 , 2.75e-06 , 9.48e-08 , 4560 , 3360]	[619.14 , 0.00 , 1.18e-06 , 7280 , 4960]
[96.18 , 2.24e-06 , 8.57e-08 , 0 , 3280]	[677.39 , 0.00 , 1.36e-06 , 4480 , 4960]
[59.79 , 3.01e-06 , 8.14e-08 , 7360 , 0]	[608.18 , 4.74e-05 , 4.79e-07 , 10240 , 1920]
[94.22 , 3.11e-06 , 8.88e-08 , 7360 , 3280]	[557.13 , 0.00 , 1.16e-06 , 10240 , 4960]

A byproduct of using IC-READ-CH for anomaly detection is the identification of subclasses or modes of operation of normal operation, which is estimated to be 4 in the case of the shuttle dataset, and 6 in the case of the bearing dataset.

6.8 Conclusion

In this paper, a new unsupervised divisive Interpretable Clustering for Rule Extraction and Anomaly Detection (IC-READ) technique is proposed. The proposed technique targets revealing interesting information in an unlabeled dataset, where this information can be represented in simple Disjunctive Normal Form (DNF) logic.

This is achieved through an unsupervised data binarization procedure using histograms, where each binary attribute represents a candidate cut-point for a certain feature. Then a beam search procedure which preserves the n-best candidates at each step for exploration, instead of using a greedy algorithm that cannot undo any selected split. The proposed IC-READ technique also has a built-in estimation for the number of clusters, since the used quality of split measures are cluster validity indices, which means that the number of clusters is not a required input. This reliefs the analyst from a trial and error phase to choose the best number of clusters

The proposed technique is shown to outperform, or achieve comparable results with other well-known clustering techniques such as k-means, and agglomerative clustering on several UCI benchmark datasets. As shown in Table 6.2.

Another application for the proposed technique is anomaly detection. A proposed modification allows the IC-READ technique to be used for anomaly detection through cluster tightening. The performance of the proposed approach is compared with the popular One-Class Classification Support Vector Machine (OCCSVM) on two datasets. One of them is the shuttle dataset from the NASA StatLog project, and the other is the bearing vibration analysis dataset provided by the Center of Intelligent Maintenance Systems (IMS), University of Cincinnati.

The proposed approach achieves high detection rate using very low anomalous fractions, unlike OCCSVM which requires a lot of tuning to identify the correct anomalous fraction, as well as its other parameters such as the kernel and its internal factors. This tuning procedure can be in some cases infeasible due to the lack of enough labelled anomalous data. A pseudo code for the proposed algorithm is given in the appendix.

So far, the implementation of this algorithm runs sequentially on a CPU, yet it can be easily parallelized to speed up the computations. The proposed algorithm can be considered for quantization, or lossy compression applications. Also, it can be considered as an observation space divider as a preparation step for training a group of experts.

6.9 Appendix

6.9.1 Pseudo Code for Main Program

```
• Initialization:
    ○ $[CP,\ B] = GetCandidateCutpoints(X)$
    ○ $LB = GetCandidateLabels(CP, B)$
    ○ $EV = EvaluateEachCandidate(LB, X)$
    ○ $BC = KeepNBestCandidates(EV, CP, LB)$
• Iterate until the maximum combination length is reached (maximum pattern dimension):
    ○ For each of the previous NBest splits:
        ▪ $[CPs, Bs] = GetCandidateCutpoints(Xs)$
        ▪ $LBs = GetCandidateLabels(CPs, Bs)$
        ▪ $EVs = EvaluateEachCandidate(LBs, Xs)$
        ▪ $BCs = KeepNBestCandidates(EVs, CPs, LBs)$
• Find the best combination of splits which has the maximum evaluation value.
```

6.9.2 Pseudo Code for $\boldsymbol{GetCandidateCutpoints(X)}$

Input: a continuous observation matrix with each row representing an observation, and each column a continuous feature.

Output: the candidate cut-points, and the binarized observation matrix.

For each feature F_j

```
1. $binSize = \frac{\max(X(:,j)) - \min(X(:,j))}{n_b}$, where $n_b$ is the number of bins, and $X(:,j)$ means all the rows for feature $j$.
2. For each bin $i$ in the histogram
    2.1. $h_{F_j}(i) = \sum_{k=1}^{n}\left[ \left(X(k,j) \geq \{\min(X(:,j)) + (i-1) * binSize\}\right) \, and \, \left(X(k,j) \leq \{\min(X(:,j)) + (i) * binSize\}\right) \right]$, where $n$ is the number of observations in $X$.
3. $\Delta h_{F_j}(i) = h_{F_j}(i+1) - h_{F_j}(i)$.
```

```
4. $minima = \text{index}\left( \Delta\left(\text{sign}\left(-\Delta h_{F_j}\right)\right) == 2 \right)$, where sign() is a function that returns the sign of its argument, and index() returns the indices of the none zero elements in its argument.
5. Return:
    5.1. $CP_{F_j}(i_m) = \frac{h(i)+h(i+1)}{2}$, is the cut-points for the $j^{th}$ feature, where $i \in minima$, and $1 \le i_m \le |minima|$ the cardinality of the set of minima.
    5.2. For each observation $k$
        5.2.1. $B_{F_j}(k,i) = \begin{cases} 1 \ \ if \ \ F_j \ge CP_{F_j}(i_m) \\ 0 \ \ if \ \ F_j < CP_{F_j}(i_m) \end{cases}$, the binary representation of the $j^{th}$ feature of the $k^{th}$ observation.
```

6.9.3 Pseudo Code for $\boldsymbol{GetCandidateLabels}(\boldsymbol{CP}, \boldsymbol{B})$

Input: the candidate cut-points CP and the binarized feature matrix B.

Output: the candidate labeling LB^t of observations for each candidate cut-point, for the iteration number t.

```
6. If initial CP, which is the first time the function is called in the code
    6.1. $LB^1 = B$, the candidate labels for the initialization step
7. Else
    7.1. For each set of observations having label $L^{t-1}$ in $LB^{t-1}$
        7.1.1. For each binary feature $B(L^{t-1}, j)$
            7.1.1.1. $L^t = L^{t-1} + B(L^{t-1}, j) * [B(L^{t-1}, j) + n_c^{t-1}]$, where $n_c^{t-1}$ is the number of clusters in the step $t-1$.
```

6.9.4 Pseudo Code for $\boldsymbol{EvaluateEachCandidate}(\boldsymbol{LB}, \boldsymbol{X})$

Input: the candidate labels LB, and the observation matrix X

Output: the validity index, which is used as the quality measure for this candidate set of clusters.

```
7. $EV(LB) = \frac{N-k}{k-1} \frac{\sum_{c_i \, \epsilon \, C} N_{c_i} d(\bar{c}_i, \mu)}{\sum_{c_i \, \epsilon \, C} \sum_{x_i \, \epsilon \, c_i} d(x_i, \bar{c}_i)}$, illustrated in section 2.2.1.2 in the paper.
```

6.9.5 Pseudo Code for $\boldsymbol{KeepNBestCandidates}(\boldsymbol{EV}, \boldsymbol{CP}, \boldsymbol{LB})$

Input: the new values of the validity index for the candidate cut-points and their corresponding labels.

Output: update the data structures.

Update and keep track of the NBest cut-points, their corresponding feature index, sequence of splits identified until this point, and their corresponding evaluation values. This is done by storing all of these values in suitable data structures to be retrieved when needed.

Table 6.7 Summary of Potentials and Limitations of IC-READ

Proposed Technique	Interpretable Clustering for Rule Extraction and Anomaly Detection
Rational, Need	Provide an unsupervised rule-extraction technique that can be used also for anomaly detection
Potential in industry	Identifying modes of operation and anomalous behavior for subsequences of observations
Benchmarks used	11 UCI datasets and NASA bearing outer-race failure dataset
Average performance improvement against benchmark	20% improvement in cluster purity
Scalability of the algorithm	It can handle any amount of data
Computational Complexity	The testing phase is of complexity $O(d)$ where d is the number of sensor readings
Extendibility to other potential applications	Any application that requires clustering of subsequences of data, or anomaly detection.
Inputs to the algorithm	Multivariate subsequences of data that have no labels, or belong to a single known class which is typically normal operation
Output from the algorithm	Clusters and ranges of sensor readings that identify different normal modes of operation.
Parameters to set / decided upon	Defining the maximum depth of cluster splitting or defining the number of clusters.

CHAPTER 7 GENERAL DISCUSSION

This thesis provides data analysts working in the industrial field with four tools that serve different problems which have a sequential nature. Most of the data collected from the industrial field is constituted of sequences of observations while monitoring certain working assets. The machine learning tools available for an analyst are numerous. Theoretically, none of them is always better performing than the others, which is known as the “no free lunch theorem” (Wolpert 1996). Nevertheless, practically, some perform better than others when solving specific problems. In other words, some techniques are more suitable for analyzing certain types of data more than others. Moreover, some techniques are easier to use and produce more comprehensible results than others. The above reasons were the motivation for this thesis, tackling a challenging problem concerning time-series analysis and producing comprehensible results with easy-to-use techniques.

With the above motivation in mind, four new techniques are developed in this thesis to serve the analysts working in the industrial field. There are two main types of machine learning problems from the nature of data perspective; supervised and unsupervised, where the data either include certain labels as a guide for the analysis task or not respectively.

The first two techniques provide solutions to the unsupervised tasks. One of them is named Interpretable Clustering for Rule Extraction and Anomaly Detection (IC-READ) and it is a solution to unsupervised rule induction. This technique is itself a complete unsupervised technique, which means it is not a tandem composition of an unsupervised technique to initialize another supervised one. Moreover, it can perform clustering without requiring any inputs from the analyst, and accepts a guide about the number of clusters if available. Also, it can be used to perform anomaly detection by making use of the extracted clusters. Anomaly detection requires an input from the analyst specifying a coarse estimation of the fraction of anomalous observations in the data. The results are produced in the form of simple logic. This makes the results comprehensible for decision makers and practitioners, as well as making it easy for hardware implementation in the form of a simple logic circuit. This technique is applied to machine diagnosis by inspecting chunks of observations of the working asset and deciding whether it is normal or anomalous. Results show superior and comparable performance of this proposed technique to other commonly used techniques in literature.

This technique has a limitation which is that it cannot handle variable length sequences of observations. Some of the problems that arise in the industry require that clustering sequences of observations according to their overall trends. This cannot be achieved efficiently using only small chunks of observations. This was the trigger for the second proposed technique which is the profile favoring dynamic time warping distance measure. This measure achieves a compromise between the magnitudes of the monitored signals, and their trend of change without using a weighting factor. This is because a weighting factor would require tuning by the analyst which is impractical in the case of unsupervised learning because there are no known targets to validate the choice of the factor against. This compromise was achieved by realigning the matches sequences to match each other in length using their rate of change only, and then matching the realigned signals by their magnitude. Hence, the final distance measure takes into account both the trend and magnitude of the matched signals. This distance measure was experimented using road maintenance planning data and showed promising results. The proposed distance measure is suggested for use with hierarchal clustering for grouping and visualization using a dendogram, or with kernel principal components analysis for visualizing multivariate time-series of variable lengths on simple 2D scatter plots.

These two unsupervised techniques cannot make use of labeled data when available to achieve a specific task. This was the motive for the other two techniques which focus on the supervised type of problems, specifically Remaining Useful Life (RUL) estimation, which is a very important problem in prognostics. Inspired by the idea of grouping, the first of those two is a new technique named logical analysis of survival curves. This technique modifies the classical non-parametric Kaplan-Meier survival curves by adaptively grouping survivor assets, and short-life life one and estimating the survival curves for each. This mitigates the averaging effects when making estimations using the whole dataset and gives more accurate estimations. This technique provides simple logical rules at a set of predefined inspection times separating long-surviving assets from others according to a set of predefined thresholds on the deterioration measure such as wear. The results are very simple which makes its implementation easy on any system. This technique was tested for RUL estimation on a set of 28 cutting tools working on Titanium composites. This experiment was conducted in the labs of Ecole Polytechnique de Montreal. The results show comparable performance to other well-known regression techniques, yet in much simpler format, and allows online RUL estimation, which means it does not require to buffer some observations to make reliable estimations.

The logical analysis of survival curves interprets its results in a simple manner, yet constraints on the predictions cannot be imposed explicitly, for example if late prediction endanger the system's performance. This was the motivation to implement the last technique which called bidirectional handshaking long short-term memory networks with safety oriented objective function. Another motive for this technique is provide analysts with simpler yet effective approaches to train neural networks to solve complex RUL estimation problems.

The proposed network architecture specifically serves the purpose of RUL forecasting using small sequences of observations with random initial working condition. Moreover, the proposed objective function provides restrictions on making late predictions. Also, this technique proposed a methodology to estimate the target RUL to train the network without knowing the actual health state of the working machine which usually difficult to obtain explicitly.

The proposed approach estimates the target RUL used for training using the sensor readings of the working asset, and hence requiring lesser assumptions to be made by the analyst about the target RUL. The performance of the proposed network is tested against conventional network architectures, objective functions, and target RUL generation.

The results show improvement when using each of the proposed components together as well as separately, which suggests that the proposed components can be used by the analyst with other techniques. In other words, the objective function can be used with other architectures, the proposed structure can be used with other objective functions, or the target generation can be used to train other types of networks.

CHAPTER 8 CONCLUSIONS AND RECOMMENDATIONS

This thesis presented new original techniques for different time-series analysis problems that arise in the industrial engineering field. The main objective of this thesis is to provide analysts working in the field with a set tools that require the least possible number of choices to decide on when performing their analyses. Moreover, this limitation does not affect the efficiency of the results and its comprehensibility.

The diagram in Figure 8.1 summarizes the proposed techniques and their recommended area of usage.

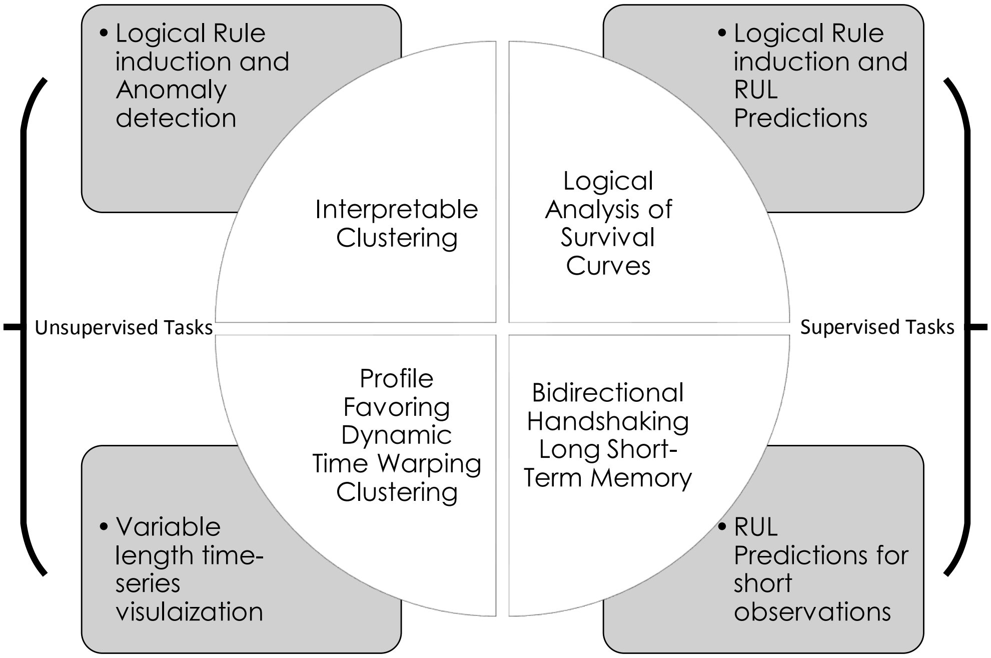

• Logical Rule induction and Anomaly detection
• Logical Rule induction and RUL Predictions
Interpretable Clustering
Logical Analysis of Survival Curves
Unsupervised Tasks
Supervised Tasks
Profile Favoring Dynamic Time Warping Clustering
Bidirectional Handshaking Long Short-Term Memory
• Variable length time-series visulaization
• RUL Predictions for short observations

Figure 8.1 Summary of the proposed techniques

8.1 Summary of Objectives' Satisfaction

The different techniques developed in this thesis satisfy its objectives according to the following highlights:

1- A new adaptive Kaplan-Meier (KM) survival curve calculation technique for RUL estimation named the Logical Analysis of Survival Curves (LASC). At every observation instant, the working asset is classified using Logical Analysis of Data (LAD) as a long life or short life tool, and the corresponding *survival curve* is chosen for *RUL estimation*. This approach is *completely non-parametric* and provides *simple rules* for classifying long surviving assets from short-life ones. The improvement in performance is due to the breakage of the augmented estimation of the survival curve using all types of working assets together.

2- A new LSTM *network architecture* to make RUL predictions using short sequences of observation with random initial wear state, along with a new *safety-oriented objective function* which favors early estimates more than late ones, and a *target RUL generation approach for the network which requires only one assumption* about the actual health state of the working asset.

3- A new time-series similarity measure using a *non-parametric* combination of Differential Dynamic Time Warping (DDTW), and the Euclidean Distance (ED). This measure gives more focus on the profile of the time-series being matched instead of their magnitude only, or trend only. This new measure is used within a hierarchal clustering technique to group sequential observations collected offline in an *unsupervised* manner. This approach can be either used for *clustering or visualization of variable length multivariate time-series*.

4- A new *Interpretable Clustering for Rule Extraction and Anomaly Detection (IC-READ)* technique that can estimate the *number of clusters automatically, or accept it as input*. Moreover, it can be used as an *anomaly detector* requiring only one parameter to choose.

Table 8.1 summarizes the proposed techniques, their purpose, their potential in industry, and their fulfillment of the objectives of the thesis.

8.2 Recommended Applications for the Proposed Techniques

The proposed Interpretable Clustering for Rule Extraction and Anomaly Detection (IC-READ) technique is an unsupervised clustering technique that produces its clustering results in terms of disjunctive normal form logic, which is a simple set of rules of conditions on the features available in the dataset. It can also be used for anomaly detection.

Table 8.1 Summary of proposed techniques and achievements of the thesis objectives

Logical Analysis of Survival Curves	
Rational, Need	Cluster sequences of variable length observations based on trends, each of which can have multiple factors
Potential in industry	Scheduling for assets' maintenance based on the type of degradation
Objectives met	*Requires no tunable parameters that affect the learning phase, it just requires defining the analysis times at which predictions will be required. *Makes predictions based on stratification with simple Boolean logic rules and a non-parametric survival curve.
Bidirectional Handshaking Long Short-Term Memory	
Rational, Need	Predicting when a partially observed time-series will reach a certain threshold based on historical data
Potential in industry	Better scheduling for maintenance for assets that undergo partial maintenance, or enter the system in a used state
Objectives met	*Proposes a new architecture to solve the problem of making predictions for the random initial-condition of assets without requiring external knowledge or a different model for initial condition estimation. *Provides a new way for automated training target generation that requires lesser assumptions than previous work.
Profile Favoring Dynamic Time Warping	
Rational, Need	Cluster and visualize sequences of variable length observations based on trends, each of which can have multiple factors
Potential in industry	Scheduling for assets' maintenance based on the type of degradation
Objectives met	*Provides a similarity measure for time-series trends that takes into account both the magnitudes of the series and their nature of change without requiring a weighting factor between both aspects. *Can be used to visualize multivariate time-series with variable lengths using wither dendrograms or 2D Kernel Principal Components.
Interpretable Clustering for Rule Extraction and Anomaly Detection	
Rational, Need	Provide an unsupervised rule-extraction technique that can be used also for anomaly detection
Potential in industry	Identifying modes of operation and anomalous behavior for subsequences of observations
Objectives met	*Performs clustering without requiring any input from the user, and can also accept the number of clusters if known *Provides results in simple Boolean Logic format *Performs anomaly detection using the extracted clusters

This technique can estimate the number of clusters on its own and does not require any input from the analyst. The results show comparable and superior results with respect to commonly used clustering techniques. As for the anomaly detection task, the analyst is required to provide a rough estimate of the fraction of anomalous data. The experiments are conducted on a bearing dataset to diagnose the observed signals as anomalous or normal. The results show that the technique is much more tolerant to estimation errors of the anomalous fraction provided by the analyst than the commonly used One-Class Classification Support Vector Machine OCCSVM technique.

This technique is recommended for analysts who have datasets that are collected without being labelled and they need to formulate their findings in a simple formulation to be provided to decision makers and other non-specialists. This technique is not advisable for use with datasets having observations with high dimensionality.

The second tool termed profile favoring Dynamic Time Warping (DTW) which is a distance measure that can deal with variable length time-series. It focuses on matching the time-series' trends and magnitudes together without neglecting either aspect, unlike conventional DTW and its variant Differential DTW. This distance measure can be used with either hierarchal clustering to group time-series in an unsupervised manner, or to visualize them using Kernel Principal Components Analysis (KPCA).

The analysist is required to provide either the number of clusters if grouping is required, or a threshold on the ratio between the sum of distances between clusters, and that within a cluster. Alternatively, the analyst can inspect the clustering dendogram and decide where to separate clusters or use the KPCA to visualize how the time-series are distributed. This technique is applied to road maintenance planning using a small dataset from Transport Quebec (TQ) and the results are compared against the conventional DTW, and it shows superior performance.

This technique is recommended when the analyst has a database of multi-variate time-series that are unlabeled for a specific task, and grouping is required. This technique groups time-series based on their trend and magnitude simultaneously, which is a useful property in the case of planning maintenance schedules or categorizing consumer products based on their performance overtime for example. The results can be clearly visualized which eases the analysts' task of post-processing the results.

The third technique named Logical Analysis of Survival Curves (LASC) provides analysts with a Remaining Useful Life (RUL) prediction technique. This technique uses a supervised rule induction classifier named Logical Analysis of Data (LAD) to adaptively stratify observations into two groups of long-life and short-life. This allows the estimation of the Survival Curves (SC) which are estimated using the nonparametric Kaplan-Meier (KM) technique to be more accurate. This is because it avoids calculating aggregated SC which average out the lifetimes of all observations in the dataset. This technique requires as input the time instants at which the model should be updates, which can be simply chosen to satisfy the Nyquist rate of the observations. It also requires defining thresholds at which the observations that identify the modes of operation of the working asset according to the measured deterioration factor such as wear. This technique has been tested on a set of 28 cutting tools working on Titanium composites. This experiment was conducted in the labs of Ecole Polytechnic de Montreal. The results were compared against other well-known techniques, such as support vector regression, and decision tree regression. The result of LASC are superior in the case of online RUL estimation, and comparable when using windows of observations with the other techniques.

This technique is particularly useful for analysts who want to make online RUL estimations. It is also useful if feedback indicators are required to be added in the system to identify the tool condition based on the monitored signals. The analyst does not have to make any assumption about the distribution of the failure times as the technique uses a non-parametric technique, which is also beneficial in the case of having small amounts of observations that would not be sufficient to estimate a parametric distribution. This technique is not recommended for use with high dimensional data which is rarely the case for monitoring signals.

The last technique is used also for RUL estimation. It uses a new bidirectional Long Short-Term Memory (LSTM) network architecture named Bidirectional Handshaking LSTM (BHLSTM). The network is trained using a new safety oriented objective function that trains the network to favor early safe predictions rather than late ones. The main purpose of this technique is to make accurate predictions when given only small sequences of observations from assets having a random initial physical state. This BHLSTM process the observed sequences in both directions consecutively using the final state of the forward LSTM to initialize the backward one. Since in most cases the actual health index of a machine is not known, then it must be assumed by the analyst. The proposed technique suggests a new target generation process for the BHLSTM network is proposed which requires no assumption from the analyst and estimates the target value from the given data. The BHLSTM architecture is compared to the conventional LSTM architectures and assumptions using the NASA turbo engine data. The results show that the proposed BHLSTM has improved performance resulting from its architecture, objective function, and it generated targets.

This technique is useful for analysts dealing with assets that are partially maintained, or enter operation is a used state. This technique can be used as a whole, or its components; namely the handshaking technique, the objective function, or the target generation, can be used with other neural networks. The proposed architecture might be slow if used with long sequences of observations. The proposed target generation process might not be very useful if the monitored signals do not well reflect the asset's condition.

The unsupervised techniques can help, in some cases where there is a lot of variation within certain classes, to ease the supervised learning task by breaking it into smaller problems having the same nature.

8.3 Future Work

There are many areas of open research for developing tools for the industrial engineering field. One of the most critical areas is the semi-supervised learning. This type of learning makes use of substantial amounts of unlabeled data to build improved models using few amounts of labeled data. In most industrial applications, data became available in substantial amounts, yet most of it is unlabeled for a specific task. That is because labeling is a tedious and costly procedure. Hence, semi-supervised learning provides a compromise for ambitious researchers and growing industries.

Another very promising area of research is transfer learning. This type of learning makes use of previously trained models and try to use the knowledge encoded to improve the performance of this model on another related task. Accumulation of knowledge from one application to another is certainly interesting for many applications.

Last but not least, exploring new Neural Network (NN) architectures to serve the industrial field is a currently growing trend. By observing different NN architectures that have become numerous in literature lately, it can be seen that the interconnections within a NN can highly impact its performance on different applications. For example, one of the recent architectures that has not been explored yet in the industrial engineering field it the NN with attention layer. This layer can help the NN focus on specific readings in the input data to produce the output. Not only can it improve performance of NNs on various tasks but also provide insight about certain observations that require attention during monitoring different assets for example. Another exemplar is the multi-objective NN, which can for example do diagnosis and prognosis at the same time.

REFERENCES OF BIBLIOGRAPHY

A.G., Rekha, Mohammed Shahid Abdulla, and A. S. Asharaf. 2017. “Lightly Trained Support Vector Data Description for Novelty Detection.” *Expert Systems with Applications* 85: 25–32.

Abadi, Martin et al. 2015. “TensorFlow : Large-Scale Machine Learning on Heterogeneous Distributed Systems.”

Abdulla, Waleed H, David Chow, Gary Sin, and New Zealand. 2003. “Cross-Words Reference Tempiate for DTW-Based Speech Recognition Systems.” In *TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region,* IEEE, 1576–79.

Abellan-Nebot, J V, and F R Subrión. 2009. “A Review of Machining Monitoring Systems Based on Artificial Intelligence Process Models.” *The International Journal of Advanced Manufacturing Technology* 47(1–4): 1–21. http://link.springer.com/10.1007/s00170-009-2191-8.

Aghabozorgi, Saeed, Ali Seyed Shirkhorshidi, and Teh Ying Wah. 2015. “Time-Series Clustering - A Decade Review.” *Information Systems* 53: 16–38. http://dx.doi.org/10.1016/j.is.2015.04.007.

Ahmad, Rosmaini, and Shahrul Kamaruddin. 2012. “An Overview of Time-Based and Condition-Based Maintenance in Industrial Application.” *Computers and Industrial Engineering* 63(1): 135–49. http://dx.doi.org/10.1016/j.cie.2012.02.002.

Alexe, Sorin, Eugene Blackstone, and Peter L. Hammer. 2003. “Coronary Risk Prediction by Logical Analysis of Data.” *Annals of Operations Research* 119: 15–42.

An, Dawn, Nam H. Kim, and Joo Ho Choi. 2015. “Practical Options for Selecting Data-Driven or Physics-Based Prognostics Algorithms with Reviews.” *Reliability Engineering and System Safety* 133: 223–36. http://dx.doi.org/10.1016/j.ress.2014.09.014.

Andrew, Dan Pelleg, Andrew Moore, and D Pelleg. 2001. “Mixtures of Rectangles: Interpretable Soft Clustering.” In *International Conference on Machine Learning (ICML-01),* , 401--408.

Aramesh, M. et al. 2014. “Survival Life Analysis of the Cutting Tools during Turning Titanium Metal Matrix Composites (Ti-MMCs).” *Procedia CIRP* 14: 605–9.

http://dx.doi.org/10.1016/j.procir.2014.03.047.

Arbelaitz, Olatz et al. 2013. “An Extensive Comparative Study of Cluster Validity Indices.” *Pattern Recognition* 46(1): 243–56.

Aye, S. A., and P. S. Heyns. 2017. “An Integrated Gaussian Process Regression for Prediction of Remaining Useful Life of Slow Speed Bearings Based on Acoustic Emission.” *Mechanical Systems and Signal Processing* 84: 485–98. http://dx.doi.org/10.1016/j.ymssp.2016.07.039.

Baccar, D., and D. Söffker. 2017. “Identification and Classification of Failure Modes in Laminated Composites by Using a Multivariate Statistical Analysis of Wavelet Coefficients.” *Mechanical Systems and Signal Processing* 96: 77–87. http://linkinghub.elsevier.com/retrieve/pii/S0888327017301826.

Bahlmann, Claus, Bernard Haasdonk, and Hans Burkhardt. 2002. “Online Handwriting Recognition with Support Vector Machines - A Kernel Approach.” *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*: 49–54.

Bailey, Michael et al. 2007. “Automated Classification and Analysis of Internet Malware.” *10th International Symposium on Recent Advances in Intrusion Detection (RAID)*: 178–97. http://link.springer.com/chapter/10.1007/978-3-540-74320-0_10.

Banjevic, D, A.K.S. Jardine, V Makis, and M Ennis. 2001. “A Control-Limit Policy and Software for Condition-Based Maintenance Optimization.” *INFOR* 39(1): 32–50. http://www.omdec.com/papers/control-limitPolicyCbmOptimization.pdf.

Bankó, Zoltán, and János Abonyi. 2012. “Correlation Based Dynamic Time Warping of Multivariate Time Series.” *Expert Systems with Applications* 39(17): 12814–23.

Barakat, Nahla, and Andrew P. Bradley. 2010. “Rule Extraction from Support Vector Machines: A Review.” *Neurocomputing* 74(1–3): 178–90. http://dx.doi.org/10.1016/j.neucom.2010.02.016.

Barragan, João Francisco, Cristiano Hora Fontes, and Marcelo Embiruçu. 2016. “A Wavelet-Based Clustering of Multivariate Time Series Using a Multiscale SPCA Approach.” *Computers and Industrial Engineering* 95: 144–55.

Ben-Akiva, Moshe, and Rohit Ramaswamy. 1993. “An Approach for Predicting Laten

Infrastructure Facility Deterioration.” *Transportation Science* 27(2): 174–93.

Benkedjouh, T., K. Medjaher, N. Zerhouni, and S. Rechak. 2015. “Health Assessment and Life Prediction of Cutting Tools Based on Support Vector Regression.” *Journal of Intelligent Manufacturing* 26(2): 213–23.

Bernal De Lázaro, José Manuel, Alberto Prieto Moreno, Orestes Llanes Santiago, and A. J. Da Silva Neto. 2015. “Optimizing Kernel Methods to Reduce Dimensionality in Fault Diagnosis of Industrial Systems.” *Computers and Industrial Engineering* 87: 140–49.

Bishop, Christopher M. 2006. 1 Statewide Agricultural Land Use Baseline 2015 *Pattern Recognition and Machine Learning*. 4th ed. New York: Springer.

Boley, Daniel. 1998. “Principal Direction Divisive Partitioning.” *Data Mining and Knowledge Discovery* 2(4): 325–44. http://dx.doi.org/10.1023/A:1009740529316%5Cnhttp://www.springerlink.com/content/w15313n737603612/.

Boros, E, Pl Hammer, and T Ibaraki. 2000. “An Implementation of Logical Analysis of Data.” *IEEE Transactions on Knowledge and Data Engineering* 12(2): 292–306. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=842268.

Boros, Endre et al. 2011. “Logical Analysis of Data: Classification with Justification.” *Annals of Operations Research* 188(1): 33–61.

Caliński, T, and J Harabasz. 1974. “A Dendrite Method for Cluster Analysis.” *Communications in Statistics-theory and Methods* 3(1): 1–27. http://www.tandfonline.com/doi/abs/10.1080/03610927408827101.

Canfield, R V. 1986. “Cost Optimization of Periodic Preventive Maintenance.” *IEEE Transactions on Reliability* R-35(1): 78–81.

Chavent, Marie, Yves Lechevallier, and Olivier Briant. 2007. “DIVCLUS-T: A Monothetic Divisive Hierarchical Clustering Method.” *Computational Statistics and Data Analysis* 52(2): 687–701.

Chen, Tao, Ruifeng Xu, Yulan He, and Xuan Wang. 2016. “Improving Sentiment Analysis via Sentence Type Classification Using BiLSTM-CRF and CNN.” *Expert Systems With*

Applications 72: 1–10. http://dx.doi.org/10.1016/j.eswa.2016.10.065.

Chen, Tiejian et al. 2016. “A Machine Vision Apparatus and Method for Can-End Inspection.” *IEEE Trans. Instrum. Meas.* 65(9): 2055–66.

Chen, Yanping et al. 2015. “The UCR Time Series Classification Archive.” www.cs.ucr.edu/~eamonn/time_series_data/.

Cho, Kyunghyun, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. “On the Properties of Neural Machine Translation: Encoder-Decoder Approaches.” http://arxiv.org/abs/1409.1259.

Chollet, F. 2015. “Keras.” *GitHub repository*.

Chung, Wingyan, and Tzu Liang Tseng. 2012. “Discovering Business Intelligence from Online Product Reviews: A Rule-Induction Framework.” *Expert Systems with Applications* 39(15): 11870–79. http://dx.doi.org/10.1016/j.eswa.2012.02.059.

Corduas, Marcella, and Domenico Piccolo. 2008. “Time Series Clustering and Classification by the Autoregressive Metric.” *Computational Statistics and Data Analysis* 52(4): 1860–72.

Crama, Yves, Peter L. Hammer, and Toshihide Ibaraki. 1988. “Cause-Effect Realtionships and Partially Defined Boolean Functions.” *Annals of Operations Research* 16: 299–325.

Crammer, Koby, and Yoram Singer. 2001. “On The Algorithmic Implementation of Multiclass Kernel-Based Vector Machines.” *Journal of Machine Learning Research (JMLR)* 2: 265–92.

Datong, Liu, Pang Jingyue, Zhou Jianbao, and Peng Yu. 2012. “Data-Driven Prognostics for Lithium-Ion Battery Based on Gaussian Process Regression.” In *Prognostics and System Health Management (PHM), 2012 IEEE Conference On*, , 1–5.

Duda, Richard O., Peter E. Hart, and David G. Stork. 2000. *Pattern Classification*. 2nd ed. John Wiley & Sons.

Durso, Pierpaolo, and Elizabeth Ann Maharaj. 2012. “Wavelets-Based Clustering of Multivariate Time Series.” *Fuzzy Sets and Systems* 193: 33–61. http://dx.doi.org/10.1016/j.fss.2011.10.002.

Efrat, Alon, Quanfu Fan, and Suresh Venkatasubramanian. 2007. “Curve Matching , Time Warping , and Light Fields : New Algorithms for Computing Similarity between Curves.” *J.*

Math Imaging Vis. 27(3): 203–16.

Everitt, Brian S, Sabine Landau, Morven Leese, and Daniel Stahl. 2011. *Cluster Analysis*. 5th ed. John Wiley & Sons.

Feldman, Kiri, Peter Sandborn, and Taoufik Jazouli. 2008. “The Analysis of Return on Investment For PHM Applied to Electronic Systems.” In *2008 International Conference on Prognostics and Health Management, PHM 2008,*.

Fu, Tak Chung. 2011. “A Review on Time Series Data Mining.” *Engineering Applications of Artificial Intelligence* 24(1): 164–81. http://dx.doi.org/10.1016/j.engappai.2010.09.007.

Gers, Felix. 2001. 2366 Lausanne, EPFL “Long Short-Term Memory in Recurrent Neural Networks.” http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.6677&rep=rep1&type=pdf%5Cnhttp://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.6677.

Gers, Felix a, Nicol N Schraudolph, and Jurgen Schmidhuber. 2002. “Learning Precise Timing with LSTM Recurrent Networks.” *Journal of Machine Learning Research* 3(1): 115–43. http://www.crossref.org/jmlr_DOI.html.

Ghasemi, Alireza, Soumaya Yacout, and M-salah Ouali. 2010. “Evaluating the Reliability Function and the Mean Residual Life for Equipment With Unobservable States.” *IEEE Transactions on Reliability* 59(1): 45–54.

Gilabert, Eduardo, Santiago Fernandez, Aitor Arnaiz, and Egoitz Konde. 2017. “Simulation of Predictive Maintenance Strategies for Cost-Effectiveness Analysis.” *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 231(13): 2242–50.

Górecki, Tomasz, and Maciej Łuczak. 2015. “Multivariate Time Series Classification with Parametric Derivative Dynamic Time Warping.” *Expert Systems with Applications* 42(5): 2305–12.

Górecki, Tomasz, and MacIej Łuczak. 2013. “Using Derivatives in Time Series Classification.” *Data Mining and Knowledge Discovery* 26(2): 310–31.

Graves, Alex, and Navdeep Jaitly. 2014. “Towards End-To-End Speech Recognition with

Recurrent Neural Networks." *JMLR Workshop and Conference Proceedings* 32(1): 1764–1772. http://jmlr.org/proceedings/papers/v32/graves14.pdf.

Greff, Klaus et al. 2017. "LSTM: A Search Space Odyssey." *IEEE Transactions on Neural Networks and Learning Systems* 28(10): 2222–32.

Guillén, A.J., A. Crespo, M. Macchi, and J. Gómez. 2016. "On the Role of Prognostics and Health Management in Advanced Maintenance Systems." *Production Planning & Control* 27(12): 991–1004. http://www.tandfonline.com/doi/full/10.1080/09537287.2016.1171920.

Guo, Liang et al. 2017. "A Recurrent Neural Network Based Health Indicator for Remaining Useful Life Prediction of Bearings." *Neurocomputing* 240: 98–109. http://dx.doi.org/10.1016/j.neucom.2017.02.045.

Haddad, G, P a Sandborn, and M G Pecht. 2012. "An Options Approach for Decision Support of Systems with Prognostic Capabilities." *IEEE Transactions on Reliability* 61(4): 872–83. http://www.scopus.com/inward/record.url?eid=2-s2.0-84870495152&partnerID=40&md5=bc138d1d3c5df77499ec91a86cd4b976.

Hailesilassie, Tameru. 2016. "Rule Extraction Algorithm for Deep Neural Networks: A Review." *International Journal of Computer Science and Information Security* 14(7): 376–81.

Hammer, Peter L, Alexander Kogan, Bruno Simeone, and Sándor Szedmák. 2004. "Pareto-Optimal Patterns in Logical Analysis of Data." *Discrete Applied Mathematics* 144: 79–102.

Hasperué, Waldo, Laura Cristina Lanzarini, and Armando De Giusti. 2012. "Rule Extraction on Numeric Datasets Using Hyper-Rectangles." *Computer and Information Science* 5(4): 116–31. http://www.ccsenet.org/journal/index.php/cis/article/view/15766.

Haykin, Simon. 2001. 40 Angewandte Chemie International Edition *Neural Networks and Learning Machines*. Third. http://doi.wiley.com/10.1002/1521-3773(20010316)40:6%3C9823::AID-ANIE9823%3E3.3.CO;2-C.

Van der Heijden, Maarten, Marina Velikova, and Peter J F Lucas. 2014. "Learning Bayesian Networks for Clinical Time Series Analysis." *Journal of Biomedical Informatics* 48: 94–105. http://dx.doi.org/10.1016/j.jbi.2013.12.007.

Heimes, F.O. 2008. "Recurrent Neural Networks for Remaining Useful Life Estimation." In

Prognostics and Health Management, 2008. PHM 2008. International Conference On, , 1–6. http://ieeexplore.ieee.org.etechconricyt.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=4711422.

Heng, Aiwina, Andy C C Tan, et al. 2009. "Intelligent Condition-Based Prediction of Machinery Reliability." *Mechanical Systems and Signal Processing* 23(5): 1600–1614.

Heng, Aiwina, Sheng Zhang, Andy C C Tan, and Joseph Mathew. 2009. "Rotating Machinery Prognostics: State of the Art, Challenges and Opportunities." *Mechanical Systems and Signal Processing* 23(3): 724–39.

Hess, Andrew, Giulio Calvello, and Peter Frith. 2005. "Challenges, Issues, and Lessons Learned Chasing the 'Big P': Real Predictive Prognostics Part 1." In *IEEE Aerospace Conference Proceedings*, , 3610–19.

Hochreiter, Sepp, and J Urgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9(8): 1735–80. http://www7.informatik.tu-muenchen.de/~hochreit%5Cnhttp://www.idsia.ch/~juergen.

Huang, Runqing et al. 2007. "Residual Life Predictions for Ball Bearings Based on Self-Organizing Map and Back Propagation Neural Network Methods." *Mechanical Systems and Signal Processing* 21(1): 193–207.

Hubert, Lawrence, and Phipps Arabie. 1985. "Comparing Partitions." *Journal of Classification* 2(1): 193–218.

Hyndman, Rob J., and Anne B. Koehler. 2006. "Another Look at Measures of Forecast Accuracy." *International Journal of Forecasting* 22(4): 679–88.

Jardine, A. K S et al. 2015. "Current Status of Machine Prognostics in Condition-Based Maintenance: A Review." *Mechanical Systems and Signal Processing* 29(1): 135–49. http://dx.doi.org/10.1016/j.cirp.2015.05.011.

Jardine, A. K S, Daming Lin, and Dragan Banjevic. 2006. "A Review on Machinery Diagnostics and Prognostics Implementing Condition-Based Maintenance." *Mechanical Systems and Signal Processing* 20(7): 1483–1510.

Javed, Kamran, Rafael Gouriveau, and Noureddine Zerhouni. 2017. "State of the Art and

Taxonomy of Prognostics Approaches , Trends of Prognostics Applications and Open Issues towards Maturity at Different Technology Readiness Levels." *Mechanical Systems and Signal Processing* 94: 214–36. http://dx.doi.org/10.1016/j.ymssp.2017.01.050.

Johannes, Furnkranz, Gamberger Dragan, and Lavrac Nada. 2013. The Deductive Spreadsheet *Foundations of Rule Learning*. http://dx.doi.org/10.1007/978-3-642-37747-1_8.

Kantardzic, Mehmed. 2011. *Data Mining: Concepts , Models , Methods , and Algorithms*. Wiley-IEEE Press.

Kent, Renee M, and Dennis A Murphy. 2000. *Health Monitoring System Technology Assessments: Cost Benefits Analysis*.

Keogh, Eamonn J., and and Michael J. Pazzani. 2001. "Dynamic Time Warping with Higher Order Features." In *First SIAM International Conference on Data Mining (SDM'2001)*, Chicago.

Kim, Minho, and R. S. Ramakrishna. 2005. "New Indices for Cluster Validity Assessment." *Pattern Recognition Letters* 26(15): 2353–63.

Korson, S.L., B.G. Schings, P.P. Velleca, and M.B. Golding. 2003. "Method and System for Variable Flight Data Collection." https://www.google.com/patents/US6628995.

Kronek, Louis Philippe, and Anupama Reddy. 2008. "Logical Analysis of Survival Data: Prognostic Survival Models by Detecting High-Degree Interactions in Right-Censored Data." *Bioinformatics* 24(16): 248–53.

Kruskall, J.B., and M. Liberman. 1983. "The Symmetric Time-Warping Problem: From Continuous to Discrete." In *The Symmetric Time Warping Algorithm: From Continuous to Discrete, Time Warps, String Edits and Macromoleculesfrom Continuous to Discrete, Time Warps, String Edits and Macromolecules,* eds. David Sankoff and J.B. Kruskall. Addison-Wesley, 125–61.

Kuzmanić, Ana, and Vlasta Zanchi. 2007. "Hand Shape Classification Using DTW and LCSS as Similarity Measures for Vision-Based Gesture Recognition System." In *EUROCON 2007 - The International Conference on Computer as a Tool*, , 264–69.

Le, Quoc V., Navdeep Jaitly, and Geoffrey E. Hinton. 2015. "A Simple Way to Initialize Recurrent Networks of Rectified Linear Units." http://arxiv.org/abs/1504.00941.

LeCun, Yann, and Yoshua Bengio. 1995. “Convolution Networks for Images, Speech, and Time-Series.” *The handbook of brain theory and neural networks* 3361(10): 1–5.

Lee, J., M. Ghaffari, and S. Elmeligy. 2011. “Self-Maintenance and Engineering Immune Systems: Towards Smarter Machines and Manufacturing Systems.” *Annual Reviews in Control* 35(1): 111–22. http://dx.doi.org/10.1016/j.arcontrol.2011.03.007.

Lee, J., H. Qiu, G. Yu, and J. Lin. 2007. “Bearing Data Set.”

Li, Ning, Yongjie Chen, Dongdong Kong, and Shenglin Tan. 2017. “Force-Based Tool Condition Monitoring for Turning Process Using v-Support Vector Regression.” *The International Journal of Advanced Manufacturing Technology* 91(1–4): 351–61. http://link.springer.com/10.1007/s00170-016-9735-5.

Lichman, M. 2013. “UCI Machine Learning Repository.” *University of California, Irvine, School of Information and Computer Sciences*. http://archive.ics.uci.edu/ml.

Maletič, Damjan;, Matjaž; Maletič, Basim; Al-Najjar, and Boštjan Gomišček. 2014. “The Role of Maintenance in Improving Company’s Competitiveness and Profitability: A Case Study in a Textile Company.” *Journal of Manufacturing Technology Management* 25(4): 441–56.

Mazzeo, Giuseppe M., Elio Masciari, and Carlo Zaniolo. 2017. “A Fast and Accurate Algorithm for Unsupervised Clustering around Centroids.” *Information Sciences* 400–401: 63–90. http://linkinghub.elsevier.com/retrieve/pii/S0020025517305765.

Mohamad-ali, Mortada, Soumaya Yacout, and Aouni Lakis. 2014. “Fault Diagnosis in Power Transformers Using Multi-Class Logical Analysis of Data.” *Journal of Intelligent Manufacturing* 25(6): 1429–39.

Mukkamala, Mahesh Chandra, and Matthias Hein. 2017. “Variants of RMSProp and Adagrad with Logarithmic Regret Bounds.” http://arxiv.org/abs/1706.05507.

Murphy, Kevin P. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Myers, C., L. Rabiner, and A. Rosenberg. 1980. “Performance Tradeoffs in Dynamic Time Warping Algorithms.” *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28(6): 623–35.

Ng, Andrew Y, and M.I. Jordan. 2001. “On Discriminative versus Generative Classifiers: A

Comparison of Logistic Regression and Naive Bayes." *Adv. Neural Inform. Process. Syst.* 14: 605–10.

Pacella, Massimo, and Quirico Semeraro. 2007. "Using Recurrent Neural Networks to Detect Changes in Autocorrelated Processes for Quality Monitoring." *Computers and Industrial Engineering* 52(4): 502–20.

Petitjean, François, Alain Ketterlin, and Pierre Gançarski. 2011. "A Global Averaging Method for Dynamic Time Warping, with Applications to Clustering." *Pattern Recognition* 44(3): 678–93.

Qi, Min, and G. Peter Zhang. 2008. "Trend Time-Series Modeling and Forecasting with Neural Networks." *IEEE Transactions on Neural Networks* 19(5): 808–16.

Quinlan, J. R. 1986. "Induction of Decision Trees." *Machine Learning* 1(1): 81–106.

Rabiner, L R. 1989. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." *Poceedings of theIEEE* 77(2): 257–86. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=18626.

Rafiei, Davood, and Alberto O Mendelzon. 2000. "Querying Time Series Data Based on Similarity." *IEEE Transactions on Knowledge and Data Engineering* 12(5): 675–93.

Ragab, Ahmed, Mohamed-Salah Ouali, Soumaya Yacout, and Hany Osman. 2016. "Remaining Useful Life Prediction Using Prognostic Methodology Based on Logical Analysis of Data and Kaplan–Meier Estimation." *Journal of Intelligent Manufacturing* 27(5): 943–58. http://link.springer.com/10.1007/s10845-014-0926-3.

Reddy, Anupama Rajasekhara. "Combinatorial Pattern-Based Survival Analysis Combinatorial Pattern-Based Survival Analysis." Rutgers University.

Riad, A. M., Hamdy K. Elminir, and Hatem M. Elattar. 2010. "Evaluation of Neural Networks in the Subject of Prognostics as Compared to Linear Regression Model." *International Journal of Engineering & Technology* 10(06): 52–58.

Rigamonti, Marco et al. 2017. "Ensemble of Optimized Echo State Networks for Remaining Useful Life Prediction." *Neurocomputing* 281: 121–38. https://doi.org/10.1016/j.neucom.2017.11.062.

Rokach, Lior, and Oded Maimon. 2008. *Data Mining with Decision Trees: Theory and Applications*. World Scientific.

Rousseeuw, Peter J. 1987. "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." *Journal of Computational and Applied Mathematics* 20: 53–65.

Ryoo, Hong Seo, and In Yong Jang. 2009. "MILP Approach to Pattern Generation in Logical Analysis of Data." *Discrete Applied Mathematics* 157(4): 749–61. http://dx.doi.org/10.1016/j.dam.2008.07.005.

Sakoe, Hiroaki, and Seibi Chiba. 1978. "Dynamic Programming Algorithm Optimization for Spoken Word Recognition." *Ieee Transactions on Acoustics, Speech, and Signal Processing* 26(1): 43--49. http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.114.3782.

Saxena, Abhinav, Member Ieee, et al. 2008. "Damage Propagation Modeling for Aircraft Engine Prognostics." In *Proceedings of IEEE International Conference on Prognostics and Health Management*, , 1–9.

Saxena, Abhinav, Jose Celaya, et al. 2008. "Metrics for Evaluating Performance of Prognostic Techniques." In *International Conference on Prognostics and Health Management*, , 1–7.

Saxena, Abhinav et al. 2010. "Metrics for Offline Evaluation of Prognostic Performance." *International Journal of Prognostics and Health Management* (1): 1–20. http://72.27.231.73/sites/phmsociety.org/files/phm_submission/2010/ijPHM_10_001.pdf.

Sayers, Michael W., and Steven M. Karamihas. 1998. *The Little Book of Profiling*. University of Michigan. http://www.amazon.com/dp/1905828063.

Scanff, E. et al. 2007. "Life Cycle Cost Impact of Using Prognostic Health Management (PHM) for Helicopter Avionics." *Microelectronics Reliability* 47(12): 1857–64.

Scholkopf, B. 2001. "The Kernel Trick for Distances." *Advances in Neural Information Processing Systems 13* 13: 301–7.

Schölkopf, B, A Smola, R Williamson, and PL Bartlett. 2000. "New Support Vector Algorithms." *Neural Computation* 12: 1207–1245. http://dx.doi.org/10.1162/089976600300015565.

Schuster, M., and K. K Paliwal. 1997. "Bidirectional Recurrent Neural Networks." *IEEE Transactions on Signal Processing* 45(11): 2673–81.

http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=650093.

Shaban, Yasser, Soumaya Yacout, and Marek Balazinski. 2015. “Tool Replacement Based on Pattern Recognition with LAD.” In *Proceedings - Annual Reliability and Maintainability Symposium*, Palm Harbor, FL, 1–6.

Shaban, Yasser, Soumaya Yacout, Marek Balazinski, and Krzysztof Jemielniak. 2017. “Cutting Tool Wear Detection Using Multiclass Logical Analysis of Data.” *Machining Science and Technology* 21(4): 526–41. https://doi.org/10.1080/10910344.2017.1336177.

Shao, Haidong, Hongkai Jiang, Huiwei Zhao, and Fuan Wang. 2017. “A Novel Deep Autoencoder Feature Learning Method for Rotating Machinery Fault Diagnosis.” *Mechanical Systems and Signal Processing* 95: 187–204. http://linkinghub.elsevier.com/retrieve/pii/S0888327017301607.

Si, Xiao Sheng, Wenbin Wang, Chang Hua Hu, and Dong Hua Zhou. 2011. “Remaining Useful Life Estimation - A Review on the Statistical Data Driven Approaches.” *European Journal of Operational Research* 213(1): 1–14. http://dx.doi.org/10.1016/j.ejor.2010.11.018.

Sikorska, J. Z., M. Hodkiewicz, and L. Ma. 2011. “Prognostic Modelling Options for Remaining Useful Life Estimation by Industry.” *Mechanical Systems and Signal Processing* 25(5): 1803–36.

Steinbach, M, G Karypis, and V Kumar. 2000. “A Comparison of Document Clustering Techniques.” In *KDD Workshop on Text Mining*, , 1–2. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4721382.

Thabtah, Fadi, Issa Qabajeh, and Francisco Chiclana. 2016. “Constrained Dynamic Rule Induction Learning.” *Expert Systems with Applications* 63: 74–85.

Venkatasubramanian, Venkat. 2005. “Prognostic and Diagnostic Monitoring of Complex Systems for Product Lifecycle Management: Challenges and Opportunities.” *Computers and Chemical Engineering* 29(6 SPEC. ISS.): 1253–63.

Vogl, Gregory W., Brian A. Weiss, and Moneer Helu. 2016. “A Review of Diagnostic and Prognostic Capabilities and Best Practices for Manufacturing.” *Journal of Intelligent Manufacturing*. http://link.springer.com/10.1007/s10845-016-1228-8.

Wang, Tianyi, Jianbo Yu, David Siegel, and Jay Lee. 2008. “A Similarity-Based Prognostics Approach for Engineered Systems.” In *International Conference on Prognostics and Health Management*, , 1–6.

Warren Liao, T. 2005. “Clustering of Time Series Data - A Survey.” *Pattern Recognition* 38(11): 1857–74.

Webb, Andrew R. 2003. *Statistical Pattern Recognition*. John Wiley & Sons.

Wolpert, David H. 1996. “The Lack of A Priori Distinctions Between Learning Algorithms.” *Neural Computation* 8(7): 1341–1390.

Wu, Yuting et al. 2017. “Remaining Useful Life Estimation of Engineered Systems Using Vanilla LSTM Neural Networks.” *Neurocomputing* 0: 1–13. http://dx.doi.org/10.1016/j.neucom.2017.05.063.

Xiong, Yimin, and Dit Yan Yeung. 2004. “Time Series Clustering with ARMA Mixtures.” *Pattern Recognition* 37(8): 1675–89.

Ye, Lexiang, and Eamonn Keogh. 2009. “Time Series Shapelets.” *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’09*: 947. http://portal.acm.org/citation.cfm?doid=1557019.1557122.

Zemouri, Ryad, Daniel Racoceanu, and Noureddine Zerhouni. 2003. “Recurrent Radial Basis Function Network for Time-Series Prediction.” *Engineering Applications of Artificial Intelligence* 16(5–6): 453–63.