Titre: Title:	Applicability and Interpretability of Logical Analysis of Data in Condition Based Maintenance
Auteur: Author:	Mohamad-Ali Mortada
Date:	2010
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Mortada, M.-A. (2010). Applicability and Interpretability of Logical Analysis of Data in Condition Based Maintenance [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/428/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/428/
Directeurs de recherche: Advisors:	Soumaya Yacout, & Aouni A. Lakis
Programme: Program:	Génie Industriel

UNIVERSITÉ DE MONTRÉAL

APPLICABILITY AND INTERPRETABILITY OF LOGICAL ANALYSIS OF DATA IN CONDITION BASED MAINTENANCE

MOHAMAD-ALI MORTADA
DÉPARTEMENT DE MATHÉMATIQUES ET DE GENIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIAE DOCTOR (Ph.D.)
(GÉNIE INDUSTRIEL)
DECEMBRE 2010

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée:

APPLICABILITY AND INTERPRETABILITY OF LOGICAL ANALYSIS OF DATA IN CONDITION BASED MAINTENANCE

Présentée par : MORTADA Mohamad-Ali

en vue de l'obtention du diplôme de : Philosophiae Doctor (Ph.D.)

a été dûment accepté par le jury d'examen constitué de :

M. ADJENGUE Luc, Ph.D., président

Mme. YACOUT Soumaya, D.Sc., directrice de recherche

M. LAKIS Aouni A., Ph.D., codirecteur de recherche

M. PELLERIN Robert, Ph.D., membre

M. TIAN Zhigang, Ph.D., membre externe

Acknowledgements

This dissertation could not have been written without Professor Soumaya Yacout, whom I am heavily indebted to. As my director of research she helped and encouraged me throughout the course of my studies. With her dedication and her insightful comments, she challenged me to never accept anything but my best efforts. I would also like to express my gratitude to my co-director of research, Professor Aouni Lakis, whose guidance and valuable advice helped me discover and navigate successfully the academic path I have chosen for myself. I thank him for giving me the opportunity to prove myself. I would also like to thank the president of the jury, Dr. Luc Adjengue, as well as the members, Dr. Robert Pellerin, and Dr. Zhigang Tian, and Dr. Jean Dansereau for their contribution and advice. Finally I would like to thank my family, without whose support I never could have made it this far.

Résumé

Cette thèse étudie l'applicabilité et l'adaptabilité d'une approche d'exploration de données basée sur l'intelligence artificielle proposée dans [Hammer, 1986] et appelée analyse logique de données (LAD) aux applications diagnostiques dans le domaine de la maintenance conditionnelle (CBM). La plupart des technologies utilisées à ce jour pour la prise de décision dans la maintenance conditionnelle ont tendance à automatiser le processus de diagnostic, sans offrir aucune connaissance ajoutée qui pourrait être utile à l'opération de maintenance et au personnel de maintenance. Par comparaison à d'autres techniques de prise de décision dans le domaine de la CBM, la LAD possède deux avantages majeurs : (1) il s'agit d'une approche non statistique, donc les données n'ont pas à satisfaire des suppositions statistiques et (2) elle génère des formes interprétables qui pourraient aider à résoudre les problèmes de maintenance. Une étude sur l'application de la LAD dans la maintenance conditionnelle est présentée dans cette recherche dont l'objectif est (1) d'étudier l'applicabilité de la LAD dans des situations différentes qui nécessitent des considérations particulières concernant les types de données d'entrée et les décisions de maintenance, (2) d'adapter la méthode LAD aux exigences particulières qui se posent à partir de ces applications et (3) d'améliorer la méthodologie LAD afin d'augmenter l'exactitude de diagnostic et d'interprétation de résultats.

Les aspects innovants de la recherche présentés dans cette thèse sont (1) l'application de la LAD dans la CBM pour la première fois dans des applications qui bénéficient des propriétés uniques de cette technologie et (2) les modifications innovatrices de la méthodologie de la LAD, en particulier dans le domaine de la génération des formes, afin d'améliorer ses performances dans le cadre de la CBM et dans le domaine de classification multiclasses.

La recherche menée dans cette thèse a suivi une approche évolutive afin d'atteindre les objectifs énoncés ci-dessus. La LAD a été utilisée et adaptée à trois applications : (1) la détection des composants malveillants (Rogue) dans l'inventaire de pièces de rechange réparables d'une compagnie aérienne commerciale, (2) la détection et l'identification des défauts dans les transformateurs de puissance en utilisant la DGA et (3) la détection des défauts dans les rotors en utilisant des signaux de vibration. Cette recherche conclut que la LAD est une approche de prise de décision prometteuse qui ajoute d'importants avantages à la mise en œuvre de la CBM dans l'industrie.

Abstract

This thesis studies the applicability and adaptability of a data mining artificial intelligence approach called Logical Analysis of Data (LAD) to diagnostic applications in Condition Based Maintenance (CBM). Most of the technologies used so far for decision support in CBM tend to automate the diagnostic process without offering any added knowledge that could be helpful to the maintenance operation and maintenance personnel. LAD possesses two key advantages over other decision making technologies used in CBM: (1) it is a non-statistical approach; as such no statistical assumptions are required for the input data, and (2) it generates interpretable patterns that could help solve maintenance problems. A study on the implementation of LAD in CBM is presented in this research whose objective are to study the applicability of LAD in different CBM situations requiring special considerations regarding the types of input data and maintenance decisions, adapt the LAD methodology to the particular requirements that arise from these applications, and improve the LAD methodology in line with the above two objectives in order to increase diagnosis accuracy and result interpretability.

The novelty of the research presented in this thesis is (1) the application of LAD to CBM for the first time in applications that stand to benefit from the advantages that this technology provides; and (2) the innovative modifications to LAD methodology, particularly in the area of pattern generation, in order to improve its performance within the context of CBM.

The research conducted in this thesis followed an evolutionary approach in order to achieve the objectives stated in the Introduction. The research applied LAD in three applications: (1) the detection of Rogue components within the spare part inventory of reparable components in a commercial airline company, (2) the detection and identification of faults in power transformers using DGA, and (3) the detection of faults in rotor bearings using vibration signals. This research concludes that LAD is a promising decision making approach that adds important benefits to the implementation of CBM in the industry.

Condensé en Français

Introduction

Cette thèse étudie l'applicabilité et l'adaptabilité d'une approche d'exploration de données basée sur l'intelligence artificielle proposée dans [Hammer, 1986] et appelée analyse logique de données (LAD) aux applications diagnostiques dans le domaine de la maintenance conditionnelle (CBM). La maintenance conditionnelle (CBM) est définie comme une procédure qui surveille la santé d'un bien physique en utilisant des procédures non intrusives et recommande des mesures d'entretien lorsque le comportement anormal du système l'exige. Les applications diagnostiques détectent et isolent les défauts dans un bien physique et aident à la prise de décision quant aux actions à prendre. Un programme de maintenance conditionnelle comprend les trois étapes suivantes : acquisition de données, traitement de données et prise de décision [Jardine et al. 2006]. La première étape consiste en la collecte de données relatives à la santé de l'équipement ou le bien. La deuxième étape analyse et traite les données obtenues à partir de la première étape et extrait toutes informations pertinentes. La troisième étape consiste à utiliser un système de prise de décision pour recommander des stratégies de maintenance appropriées sur la base des données analysées.

L'acquisition de données peut être divisée en 2 grandes catégories : les données d'événement et les données de surveillance d'état. Les données d'événement donnent des informations historiques sur ce qui est arrivé à l'équipement durant sa vie. Quelques exemples de ce type de données sont les nombres de défaillance, le mécanisme de défaillance et les raisons de défaillance. Les données de surveillance d'état révèlent de l'information sur l'état du système et sa dégradation ou amélioration. Elles peuvent être divisées en 3 types : les données en valeur, les données en forme d'onde et les données multidimensionnelles. Les données en valeur sont des mesures simples prises à un moment ou un instant. Les données en forme d'onde sont généralement des séries chronologiques telles que des signaux de vibration ou des signaux acoustiques. Les données multidimensionnelles sont des données observées sur 3 dimensions ou plus.

Après l'acquisition, la prochaine étape d'une stratégie de maintenance conditionnelle est le traitement des données dans le but d'y extraire de l'information pertinente sous forme d'indicateurs de l'état du système. Le type d'outil utilisé pour le traitement de données dépend du type de données acquises. Les données en forme d'onde, comme les signaux de vibration, sont les données les plus rencontrées dans les applications diagnostiques. Pour cette raison, de nombreuses techniques d'analyse du signal de vibration ont été conçues pour extraire des données pertinentes de ces signaux [Hamdy 2008; Mandal & Asif 2007].

La troisième étape d'une stratégie de maintenance conditionnelle nécessite la mise en œuvre d'un système de prise de décision concernant les actions à prendre en se basant sur l'information obtenue à partir des données traitées. Des technologies différentes ont été utilisées pour automatiser cette tâche et remplacer le diagnostic manuel. Les modèles de décision automatique peuvent être divisés en deux catégories en fonction de leur architecture : les approches statistiques et les approches d'intelligence artificielle.

La LAD est une approche d'exploration de données (Data-Mining) avec apprentissage supervisé qui extrait des formes à partir des données binarisées et formule des règles de décision afin de classifier des données nouvelles dans des classes séparées. L'approche est mise en œuvre en trois étapes : la binarisation des données, la génération des formes et la formation de la fonction discriminante de décision.

L'étape de binarisation consiste à traduire les données d'apprentissage en un ensemble de données binaires en utilisant une technique de binarisation qui traduit chaque indicateur de l'état du système par un ensemble d'attributs binaires. Chaque indicateur u se traduit en au moins un attribut binaire $b_i(u)$. Le nombre d'attributs binaires est régi par une règle établie selon les valeurs que u prend.

L'étape de génération de formes génère un ensemble de règles booléennes appelées *formes* qui sont vraies pour des observations d'une classe et pas l'autre. Une forme $\boldsymbol{p}$ est composée de *littéraux;* un littéral est une variable binaire x ou sa négation $\bar{x}$ [Boros et al. 2000]. Chaque attribut binaire b_i dans l'ensemble d'apprentissage peut être représenté dans une forme $\boldsymbol{p}$ par un littéral x_i ou sa négation $\overline{x_i}$. Dans son sens le plus strict, une forme $\boldsymbol{p}$ de degré d est une conjonction de d littéraux de telle sorte que c'est vrai pour au moins une observation d'une classe et faux pour toutes les observations de l'autre classe. Une forme qui est vraie pour une certaine observation est dite à *couvrir* cette observation. Pour une forme $\boldsymbol{p}$ couvrant une observation appartenant à une certaine classe, un littéral x_i peut faire partie de cette forme si l'observation couverte par $\boldsymbol{p}$ a la valeur 1 dans b_i. De même, le littéral $\overline{x_i}$ peut faire partie de la forme si l'observation couverte par $\boldsymbol{p}$ a la valeur 0 dans b_i. Plusieurs techniques de génération de formes ont été discutées dans la littérature; elles peuvent être classées en techniques de recensement, heuristiques et programmation linéaire. Cette thèse a adopté une technique de cette dernière classe.

La dernière étape dans la formation d'un système de décision LAD est l'étape de formation de fonction discriminante où les formes générées à l'étape précédente sont utilisées pour créer une fonction de décision appelée le discriminant. Pour un système de décision LAD à deux classes, où une classe est marquée positive et l'autre négative, une fonction discriminante prend la forme suivante :

$$\Delta(\mathbf{a}_i) = \sum_{n=1}^{N} w_n^+ \boldsymbol{p}_n^+(\mathbf{a}_i) - \sum_{m=1}^{M} w_m^- \boldsymbol{p}_m^-(\mathbf{a}_i)$$

où chaque $\boldsymbol{p}_n^+$ ou $\boldsymbol{p}_m^-$ est l'une des N formes positives ou M formes négatives dans l'étape précédente, respectivement. $\boldsymbol{p}_n^+(\mathbf{a}_i) = 1$ si une observation binarisée $\mathbf{a}_i$ est couverte par la forme $\boldsymbol{p}_n^+$ et $\boldsymbol{p}_n^+(\mathbf{a}_i) = 0$ si c'est le contraire. Idem pour les formes négatives. Les valeurs $w \geq 0$ sont des poids, attribués à chaque forme pour majorer son importance. Si la sortie de la fonction discriminante ci-dessus est positive pour une certaine observation, alors l'observation appartient à la classe positive. Dans le cas contraire, l'observation appartient à la classe négative. Si la sortie est égale à zéro, aucune décision ne peut être prise.

La plupart des technologies utilisées à ce jour comme système de décision dans la maintenance conditionnelle ont tendance à faire des décisions diagnostiques, sans offrir aucune connaissance ajoutée qui pourrait être utile à l'opération de maintenance et au personnel de maintenance. La LAD possède deux avantages majeurs sur les autres technologies de prise de décision utilisées dans la CBM :

1 - La LAD est une approche non statistique -- Avec la LAD, il n'est pas nécessaire de faire des hypothèses statistiques concernant les données d'entrée. Cela signifie qu'on peut

utiliser des indicateurs qui ne sont pas indépendants et identiquement distribués comme données d'entrée dans la LAD, par exemple les indicateurs sous forme de statistiques descriptives extraits du même signal de vibration. Il n'est également pas nécessaire de respecter des restrictions sur le type de données à utiliser. Par conséquent, les données d'événement ainsi que les données de surveillance d'état peuvent être utilisées.

2 - La LAD génère des formes interprétables -- Les modèles de décision générés par la LAD se composent de formes, qui sont des caractérisations significatives des attributs qui groupent des observations dans une classe commune. Chaque attribut représente une valeur ou un intervalle d'une caractéristique physique d'un équipement ou d'un bien.

Le pouvoir de générer des formes peut fournir une mine de ressources pour les spécialistes de la maintenance qui pourrait être utilisée pour résoudre les causes d'une défaillance et améliorer le programme de maintenance conditionnelle en place.

L'application de la LAD dans le domaine de la maintenance a été récemment évoquée à l'École Polytechnique de Montréal [Salamanca & Yacout, 2007] et [Yacout, 2010]. Une étude complète sur la mise en œuvre de la LAD dans le domaine de la CBM n'existait pas avant la recherche effectuée dans le cadre de cette thèse. En tant que tel, les principaux objectifs de cette thèse sont :

1 - étudier l'applicabilité de la LAD dans des situations différentes de la CBM qui nécessitent des considérations particulières concernant les types de données d'entrée et les décisions de maintenance;

2 - adapter la méthodologie LAD aux exigences particulières des différentes applications de la CBM;

3 - améliorer la méthodologie de la LAD afin d'augmenter l'exactitude de diagnostic et l'interprétation des résultats.

Les aspects innovants de la recherche présentés dans cette thèse sont (1) l'application de la LAD dans la CBM pour la première fois dans des applications qui bénéficient des propriétés uniques de cette technologie et (2) les modifications innovatrices de la méthodologie de la LAD, en particulier dans le domaine de la génération de formes, afin d'améliorer ses performances dans le cadre de la CBM et dans le domaine de classification multiclasses.

Les recherches menées dans cette thèse ont suivi une approche évolutive afin d'atteindre les objectifs énoncés ci-dessus. La recherche a donc commencé par la mise en œuvre d'une méthodologie éprouvée de la LAD dans une application industrielle innovante : la détection des composants malveillants (Rogue) dans l'inventaire de pièces de rechange réparables d'une compagnie aérienne commerciale. Avec le progrès de la recherche, des modifications novatrices ont été introduites à la méthodologie de la LAD qui a permis sa mise en œuvre dans les applications industrielles plus compliquées, mais plus couramment rencontrées : 1- la détection et l'identification des défauts dans les transformateurs de puissance en utilisant l'analyse des gaz dissous (DGA) à l'aide d'un modèle multicouches de la LAD, 2- la détection des défauts dans les roulements en utilisant des signaux de vibration, 3- le diagnostic des défauts dans les transformateurs de puissance à l'aide d'un modèle multiclasses de la LAD.

Détection des composants Rogue en utilisant l'analyse logique des données

Dans un inventaire de pièces de service réparables, les composants *Rogue* sont les pièces dont le mode de défaillance est hors du champ d'application des procédures de détection standard et de remise en état. Ceci fait en sorte que les composants Rogue sont difficiles à identifier et à isoler. Les composants Rogue ont affligé l'industrie du transport aérien et semé le chaos dans leurs programmes de gestion du patrimoine. Les équipes de maintenance s'appuient sur leurs expertises pour identifier ces composants Rogue et les isoler de l'inventaire afin d'éviter les perturbations opérationnelles.

Il n'y a pas eu de recherche sur la mise en œuvre des techniques de diagnostic pour la détection automatique des composants Rogue. La LAD a été utilisée ici pour automatiser le système de décision élaboré par des experts tout en fournissant de la rétroaction aux experts en forme de formes interprétables. Des données d'événements historiques ont été utilisées comme données d'apprentissage pour la méthode LAD. Les indicateurs ont été obtenus à partir des codes de motifs de la dépose (Reason for Removal Codes - RRC) et les codes de délais de renvoi (Time to Removal - TTR). Ces indicateurs ont été traditionnellement utilisés par le personnel de maintenance pour détecter et isoler les composants Rogue. Un modèle de décision LAD a été construit en utilisant un algorithme classique ascendant de génération de formes. L'exactitude des résultats obtenus en utilisant le modèle de décision LAD révèle une mesure de qualité maximale de 99,65 %. Le taux de vrais positifs indiquant le nombre de composants Rogue qui ont été correctement détectés lors de la classification est de 100 %. Aucun composant Rogue n'a été classé à tort comme non-Rogue et vice versa. Plus important encore, une évaluation des formes générées par le modèle de décision LAD a révélé qu'elles sont semblables à celles trouvées par l'analyse des données de composants Rogue par des experts. Les résultats des tests ont donc montré que la LAD est capable de détecter automatiquement les composants Rogue en utilisant les données historiques de performance des composants. Cela a démontré le potentiel de la LAD, non seulement pour automatiser le processus de décision en économisant ainsi beaucoup de ressources, mais aussi de potentiellement générer de nouvelles formes si les données de surveillance d'état sont mises à sa disposition. Les nouvelles formes qui pourraient être générées à partir des données de surveillance d'état peuvent aider à comprendre les causes racines des composants Rogue, ce qui aide à mettre à jour les mesures de maintenance préventive (PM) mises en place pour éviter leur apparition.

Diagnostic des défauts de transformateurs de puissance en utilisant la LAD en configuration 2-classes

Les transformateurs de puissance sont des équipements à prix élevé qui nécessitent une surveillance continue afin de détecter tout défaut dans leur fonctionnement avant que des risques sur la sécurité ne se produisent. Les indicateurs les plus utilisés pour le diagnostic des transformateurs de puissance sont obtenus en utilisant l'analyse des gaz dissous (DGA). Pour détecter les défauts dans le fonctionnement d'un transformateur, l'analyse se fonde sur le phénomène de la décomposition chimique de l'huile en gaz d'hydrocarbures à certaines conditions environnementales. La composition des gaz produits peut être liée au type de défaut qui a eu lieu même si de nombreux facteurs non liés au défaut ont une influence considérable sur l'arrivée du défaut. Plusieurs interprétations diagnostiques des formes faites par des experts et reliant la composition des gaz à des défauts spécifiques ont été publiées dans les dernières décennies. Les interprétations les plus courantes sont les rapports de diagnostic Burton & Davis (1972), les ratios de Rogers (1974), le Triangle Duval (1970), les ratios Dornenburg et plus récemment, la version révisée de la Commission électrotechnique internationale IEC 60599 (1999).

Nous utilisons la LAD pour automatiser la détection et l'isolement des défauts dans les transformateurs de puissance tout en générant des formes à partir des données traitées. Afin de répondre à la demande particulière de cette application, nous introduisons une technique de génération de formes modifiée pour la LAD, basée sur la programmation linéaire en nombres entiers (Mixed 0-1 Integer and Linear Programming - MILP). La motivation derrière la modification est la nécessité de générer plus de formes pour en accroître la « différentiabilité » de la fonction discriminante du modèle de décision LAD et d'augmenter le volume de connaissances utiles générées par le modèle de décision. Ceci a été réalisé par l'introduction d'une nouvelle série de conditions à la méthodologie de génération de formes basée sur MILP et en modifiant le mécanisme qui boucle l'algorithme de génération de formes :

$$\sum_{j=1}^{2q} r_{k,j} w_j \leq d_k - 1 \quad \forall \mathbf{r}_k \in \mathbf{R}$$

$$r_{k,j} = \begin{cases} 1 & if \quad v_{k,j} = 1 \\ -1 & if \quad v_{k,j} = 0 \end{cases} \quad j = \{1,2,\ldots,2q\}$$

où $\mathbf{r}_k$ est le vecteur booléen correspondant à chaque forme trouvée $\mathbf{v}_k \in \mathbf{V}$ de degré d_k et $\mathbf{w}$ est le vecteur de la nouvelle forme candidate. La nouvelle série de conditions et le mécanisme de bouclage permettent la génération de plusieurs formes pour chaque observation unique, tel que chaque nouvelle forme générée a la meilleure couverture et porte la mention « strong ».

Afin de différencier entre plusieurs types de défauts, plusieurs modèles de décision 2-classes de LAD ont été placés dans une configuration multicouches. Les résultats de la classification ont été comparés avec les résultats des autres modèles de décision. Dans la première expérience, la LAD a bien fonctionné contre des approches basées sur les réseaux de neurones et la logique floue et a été dépassée seulement par une approche neurale floue intégrée jumelée avec une nouvelle méthode d'extraction d'indicateurs. Dans une seconde expérience, l'exactitude obtenue était similaire aux approches de logique floue, ANN, et systèmes experts (ES), mais était inférieure à celle atteinte par SVM. Le véritable avantage de la LAD, tout comme dans l'application précédente, a été la génération automatique de formes similaires aux règles standardisées pour la détection de défauts basée sur la connaissance des experts. Une analyse des formes générées par la LAD révèle qu'ils sont similaires aux règles utilisées par la méthode de ratios de Rogers [Duval and DePablo 2001].

Détection des défauts dans les roulements en utilisant l'analyse logique des données

L'objectif ici est de tester l'applicabilité et la performance de la LAD dans la détection automatique de défauts dans les machines tournantes à l'aide des signaux de vibration. Le défi ici est notamment le traitement des données de vibration avant son utilisation par la LAD pour le diagnostic. Nous nous appuyons sur les techniques de traitement de signaux par temps et par temps-fréquence pour évaluer les signaux dans la base de données des signaux. Les données d'apprentissage utilisées pour former le modèle de décision LAD sont donc basées sur l'inspection visuelle et l'analyse des signaux traités.

Une technique MILP est utilisée pour générer les formes pour le modèle de décision LAD. Des indicateurs sont extraits des techniques de traitement de signaux dans le domaine temporel et temps-fréquence appliqués sur les signaux de vibration de roulement. Ensuite, les données des indicateurs sont binarisées pour les utiliser dans la LAD.

La technique MILP mise en œuvre dans l'application antérieure a abouti à la génération des formes qui sont des sous-ensembles de celles déjà générées. Cela a créé une redondance dans l'explication des formes générées qui a affecté le processus d'interprétation des résultats. Pour remédier à ce problème, la série des conditions suivantes a été ajoutée :

$$\sum_{j=1}^{2q} v_{k,j} w_j \le d_k - 1 \quad \forall \mathbf{v}_k \in \mathbf{V}$$

$$v_{k,j} = \begin{cases} 1 & if \quad w_{k,j} = 1 \\ 0 & if \quad w_{k,j} = 0 \end{cases} \quad j = \{1,2,\ldots,2q\}$$

où $\mathbf{v}_k \in \mathbf{V}$ est le vecteur booléen d'une forme de degré d_k, déjà trouvée, et $\mathbf{w}$ est le vecteur de la nouvelle forme candidate. Ces conditions empêchent l'algorithme de générer des formes qui sont des sous-ensembles des formes déjà obtenues. Cela augmente par conséquent le volume de nouvelles connaissances qui peuvent être créées. En raison de l'introduction des modifications ci-dessus, l'exactitude moyenne des modèles de décision a augmenté par rapport aux niveaux d'exactitude obtenue sans les modifications ci-dessus.

Le modèle de décision LAD résultant a été testé dans deux essais (sur deux roulements qui manifestent des défauts différents) pour vérifier si elle est capable d'automatiser le processus de décision. Les résultats des modèles de décision montrent une exactitude maximale variant entre 95,2 % et 97,5 % pour le premier essai et entre 97,1 % et 98,9 % pour le deuxième essai. Les formes générées par le modèle de décision LAD apportent une nouvelle perspective sur la façon de classer les signaux de vibration. Certaines formes obtenues avaient un pouvoir de différencier entre les observations normales et les observations défaillantes de plus de 80 %. Les résultats montrent donc une bonne exactitude de la classification à la fois dans le cas des indicateurs temporels et le cas des indicateurs temps-fréquences et temporels combinés. L'outil de diagnostic mis en œuvre sous la forme d'un logiciel dans un environnement de production ou des opérations de maintenance peut être très utile aux spécialistes de l'entretien car elle révèle des formes interprétables qui conduisent au diagnostic et qui facilitent les efforts visant à comprendre les raisons derrière la défaillance des composants.

Diagnostic des défauts dans les transformateurs de puissance à l'aide d'une configuration multiclasses de la LAD

Le problème du diagnostic de pannes dans les transformateurs de puissance en utilisant la DGA est revisité ici dans le but d'introduire une nouvelle approche multiclasses de la LAD. La motivation derrière la conception d'une nouvelle approche multiclasses est la complexité et le long temps d'apprentissage du modèle multicouches 2-classes de la LAD qui a été utilisé dans le deuxième article. La nouvelle approche identifie des défauts dans les transformateurs de puissance en utilisant une seule mise en œuvre du modèle de décision LAD.

En comparaison avec les modèles de décision multiclasses présentés dans la littérature, la nouvelle approche multiclasses de la LAD a donné des résultats comparables au meilleur classificateur et a été supérieure à l'approche LAD multiclasses présentée dans [Moreira 2000] en termes d'exactitude de la classification. Cependant, cela s'est fait au détriment du temps d'apprentissage qui a augmenté. Lors d'un essai sur des données de transformateur de puissance, les résultats avaient une exactitude moyenne en comparaison à d'autres méthodes de classification. Cependant, les formes obtenues ont été plus intuitives et plus significatives car elles relient plus que 2 classes en même temps. Cela peut s'expliquer par le fait que les frontières entre les différentes classes ne sont souvent pas des lignes claires qui peuvent être séparées par des règles à deux classes. Les formes multiclasses sont donc plus intuitives et s'ajustent mieux aux frontières de séparation entres les classes.

Conclusion

Cette thèse étudie la mise en œuvre d'une nouvelle approche de prise de décision, appelée Analyse Logique des Données, dans la maintenance conditionnelle. L'approche présente des avantages sur d'autres approches plus conventionnelles qui ont fait leurs preuves dans le domaine de la maintenance. La LAD a été testée sur trois applications pour la première fois : la détection des composants Rogue dans un inventaire de pièces réparables, le diagnostic de défauts de transformateur de puissance et la détection de défauts dans les roulements. Les aspects novateurs introduits à la LAD dans le cadre de la CBM sont :

1- l'utilisation d'un processus de sollicitation d'experts (Expert Elicitation) pour obtenir des connaissances qui aident à automatiser, pour la première fois, en utilisant la LAD, le processus de détection des composants Rogue;

2- l'amélioration de la méthodologie LAD pour mieux répondre à l'exigence de générer plus de formes et l'application de la LAD sur le diagnostic de transformateurs de puissance;

3- l'adaptation de la méthodologie de la LAD à l'application de diagnostic à l'aide des signaux de vibrations et l'amélioration du processus de génération de formes en termes d'exactitude et interprétabilité;

4- le développement d'une nouvelle approche multiclasses pour la LAD et l'analyse comparative entre cette approche et d'autres méthodes.

Les objectifs réalisés dans cette thèse démontrent que la LAD est une approche prometteuse dans le domaine de la maintenance conditionnelle en raison de ses bonnes performances, sa capacité d'adaptation et l'interprétabilité des formes qu'elle génère.

Les modifications à l'algorithme MILP mises au point dans cette thèse ont démontré une amélioration de la performance générale de l'approche LAD dans le diagnostic des défauts. Ces algorithmes peuvent être facilement adaptés aux différentes applications de diagnostic dans la CBM.

En plus de l'automatisation du processus de décision, la LAD crée des formes interprétables comme démontré au cours des trois applications étudiées tout au long de cette recherche. Ces modèles peuvent jouer un grand rôle dans la résolution des problèmes qui affligent les programmes de maintenance préventive (PM) dans l'industrie. Les formes générées peuvent :

1 - Aider le personnel de maintenance à confirmer les avis d'experts et les règles utilisées dans l'industrie. Cela a été démontré dans les problèmes de la détection des composants Rogue et l'identification des défauts dans les transformateurs de puissances où certaines des formes générées étaient identiques à des règles conçues en utilisant les connaissances d'experts.

2 - Aider le personnel de maintenance à mieux comprendre le comportement de l'équipement qu'il surveille par les nouvelles informations fournies. Comme le montre l'application des transformateurs de puissance, la LAD génère également de nouvelles formes qui apportent de nouvelles informations au personnel d'entretien et qui ont le potentiel pour les aider à mieux comprendre les raisons qui conduisent à la défaillance de certains équipements et les conditions dans lesquelles ils échouent.

3 - Aider dans l'amélioration des actions de maintenance préventive PM et la mise à jour des stratégies de maintenance. Le personnel de maintenance peut utiliser les formes et les nouvelles connaissances produites pour analyser leurs stratégies de maintenance et actualiser leurs stratégies PM afin d'éviter des pannes inutiles de système.

4 - Aider à évaluer les indicateurs surveillés par l'utilisation des technologies de surveillance coûteuse. Les formes générées par la LAD sont des fonctions de données caractéristiques utilisées pour surveiller l'état de l'équipement ou du système. Les formes générées donnent un aperçu sur les indicateurs qui sont les plus efficaces dans la surveillance de l'équipement pour un certain type de défaut. Sur la base de cette information, le personnel d'entretien peut mieux gérer leurs stratégies d'acquisition de données pour obtenir de l'information plus pertinente.

L'utilisation de la LAD dans la maintenance conditionnelle est donc justifiée par :

1 - sa bonne performance en termes d'exactitude et de temps de formation;

2 - l'adaptabilité de l'approche aux différentes exigences en termes de type d'application étant adressée, la nature des données d'entrée et les types de décisions nécessaires;

3 - l'avantage de la propriété d'interprétabilité des formes que la LAD possède comme avantage sur d'autres approches décisionnelles et le rôle que joue l'interprétabilité des formes en aidant le personnel de maintenance à résoudre les problèmes.

Finalement, nous avons identifié deux domaines de développement qui pourraient élargir l'applicabilité de la LAD dans la CBM. Le premier domaine est l'utilisation de la LAD pour la prise de décisions pronostiques. Cette recherche a démontré l'applicabilité de la LAD dans des applications diagnostiques. Les tests initiaux sur les applications pronostiques ont donné des résultats mitigés. Le potentiel de l'utilisation de la LAD dans les pronostics est élevé avec le développement de nouvelles procédures d'apprentissage théorique et de nouvelles façons d'interpréter les formes générées. Le deuxième domaine du développement est l'adaptation de la LAD aux applications qui exigent l'apprentissage non supervisé pour contourner le processus de « Expert Elicitation ». Cette étape est particulièrement difficile, mais les conséquences possibles de sa réalisation sont prometteuses.

Table of Contents

ACKNOWLEDGEMENTS III

RÉSUMÉ IV

ABSTRACT V

CONDENSÉ EN FRANÇAIS VI

TABLE OF CONTENTS XVIII

LIST OF TABLES XXII

LIST OF FIGURES XXV

LIST OF ACRONYMS XXVII

INTRODUCTION 1

Types of Maintenance 2

Condition Based Maintenance Methodology 4

Data Acquisition 6

Data Processing 6

Decision Making 11

Logical Analysis of Data 12

Problem Statement 13

Objectives 15

CHAPTER 1 LITERATURE REVIEW 17

1.1 Diagnostic Decision Models 17

1.1.1 Statistical Approaches 17

1.1.2 Artificial Intelligence Approaches 21

1.1.3 Other Approaches 24

1.2 LAD Literature Review 25

1.2.1 Data Binarization 25

1.2.2 Pattern Generation 28

1.2.3 Theory Formation 32

1.2.4 Alternative Implementations of LAD *33*

1.2.5 Multi-class LAD *34*

CHAPTER 2 THESIS ORGANIZATION 38

CHAPTER 3 ROGUE COMPONENTS - THEIR EFFECT AND CONTROL USING LOGICAL ANALYSIS OF DATA 41

3.1 Abstract 42

3.2 Introduction 42

3.3 Rogue Component Definition 45

3.4 The Effect of Rogue Components 47

3.4.1 Maintenance Effectiveness *47*

3.4.2 Mechanical support *49*

3.4.3 Operational Reliability *49*

3.4.4 Asset Management *50*

3.4.5 Preventive Maintenance Programs *52*

3.4.6 Maintenance Training Programs *53*

3.4.7 Component Repair Facility *53*

3.4.8 Operator / OEM Engineering *54*

3.4.9 Mechanical System Hardware *56*

3.5 Control of Rogue Components 57

3.5.1 LAD Methodology *58*

3.6 Implementation 63

3.7 Results 67

3.8 Conclusion 71

3.9 References 72

CHAPTER 4 FAULT DIAGNOSIS OF POWER TRANSFORMERS USING LOGICAL ANALYSIS OF DATA 74

4.1 Abstract 75

4.2 Introduction 75

4.3 Problem Description: Power Transformer Fault Diagnosis 76

4.3.1 Data Processing *77*

4.3.2 Automated Classification Approaches *78*

4.4 LAD Classification Approach 79

4.4.1 Data Binarization *79*

4.4.2 Pattern Generation *80*

4.4.3 Theory Formation *87*

4.5 Implementation 88

4.5.1 Databases *88*

4.5.2 Data Processing Techniques *89*

4.5.3 Multilayer LAD configurations *90*

4.6 Experimental Results 91

4.7 Pattern Interpretability 96

4.8 Conclusion 98

4.9 References 99

CHAPTER 5 DIAGNOSIS OF ROTOR BEARING DEFECTS USING LOGICAL ANALYSIS OF DATA 103

5.1 Abstract 104

5.2 Introduction 104

5.3 Analysis of Signal Features for fault detection 107

5.3.1 Time-Domain Features *108*

5.3.2 Time-Frequency Domain Features *109*

5.4 LAD Decision Model 111

5.5 Application 117

5.6 Implementation 122

5.7 Results 124

5.8 Conclusion 131

5.9 References 132

CHAPTER 6 MULTI-CLASS FAULT DIAGNOSIS IN POWER TRANSFORMERS USING LOGICAL ANALYSIS OF DATA 135

6.1 Abstract 136

6.2 Introduction 136

6.3 Multi-Class LAD Approach 138

6.3.1 Data Binarization..139

6.3.2 Pattern Generation 140

6.3.3 Theory Formation 149

6.4 Experiments...150

6.5 Implementation in CBM 153

6.6 Conclusion...155

6.7 References 156

CHAPTER 7   GENERAL DISCUSSION 160

CONCLUSION 165

REFERENCES...168

List of Tables

Table i-1: Commonly used higher-order statistics. $\boldsymbol{x}$ is a signal of length $\boldsymbol{T}$, whose mean is $\boldsymbol{\mu}$, standard deviation is $\boldsymbol{\sigma}$ and expected value $\boldsymbol{E}$. 8

Table 3-1: Non-Binarized Training Data 67

Table 3-2: Pattern Numbers found for each Decision Model 68

Table 3-3: The value of Discriminant function Δ for all 3 Decision Models 68

Table 3-4: Calculating the Quality of Classification....69

Table 3-5: Performance Measures of the 3 Decision Models 70

Table 4-1: The numerical attributes of the four data sets obtained using the four different processing methods. 89

Table 4-2: The Diagnostic Accuracy and No Decision Rate (NDR) of the 16 classification results obtained in experiment 1. 92

Table 4-3: Number of correctly classified observations for each fault type in experiment 1 in the 4 data sets trained at $l = 20$. 93

Table 4-4: The Diagnostic Accuracy and No Decision Rate (NDR) of the 16 classification results obtained in experiment 2. 95

Table 4-5: Best training time for LAD classifiers compared to the times reported for other approaches in Lv et al. [6]. 95

Table 4-6: Two patterns for D1/D2 type defects obtained from experiment 2 at $l = 1$ using absolute content data. 96

Table 4-7: The Rogers Ratio fault diagnosis rules used to identify different defect types [37]. ...96

Table 4-8: Patterns deduced from experiment 1 using the Combine ratios data set at $l = 5$ listed according to the percentage coverage of a pattern with respect to the training data set. 97

Table 5-1: 5 data sets are extracted from bearing 3 signal database. Each data set is composed of a training set and a testing set. Above is the size of each set and its composition......................124

Table 5-2: 5 data sets are extracted from bearing 4 signal database. Each data set is composed of a training set and a testing set. Above is the size of each set and its composition......................125

Table 5-3: Best Classification results using the training sets of Table 1 using all features. The table shows training times, the number of patterns created by cbmLAD and some performance statistics. 126

Table 5-4: Best Classification results using the training sets of Table 1 using wavelet features. The table shows training times, the number of patterns created by cbmLAD and some performance statistics. 127

Table 5-5: Best Classification results using the training sets of Table 2 using all features. The table shows training times, the number of patterns created by cbmLAD and some performance statistics. 128

Table 5-6: Best Classification results using the training sets of Table 2 using wavelet features. The table shows training times, the number of patterns created by cbmLAD and some performance statistics. 128

Table 5-7: The list of positive (defective) patterns generated by the LAD decision model using bearing 4`s data set 5 at k=10 with wavelet features only. 130

Table 5-8: The list of positive (defective) patterns generated by the LAD decision model using bearing 4`s data set 5 at k=10 with all 18 features. 130

Table 6-1: Data Sets obtained from the machine learning repository [35]. 150

Table 6-2: Number of Attributes and Number of Patterns Generated. 151

Table 6-3: Training Time in seconds of the Decision Models using an ordinary computer. 151

Table 6-4: Accuracy Levels Achieved using the Multi-Class LAD decision models. 151

Table 6-5: Comparison of Classification Accuracy with the results stated in Moreira [26]........152

Table 6-6: Power Transformer Data Set Feature^{12} [41] 153

Table 6-7: Rules Generated By LAD patterns. 154

List of Figures

Figure i-1: Types of Maintenance 3

Figure i-2: Time-Frequency resolution map generated by a dyadic wavelet transform 10

Figure 2-1: Progress of LAD methodology throughout the articles and CBM applications..........38

Figure 3-1: Fault Confirmation Codes as presented in [Leung et al. 2007] describe the nature of the removal 64

Figure 4-1: LAD multilayer classifiers: configuration 1. In order to classify observations into 5 fault types, 4 LAD classifiers must be trained and applied in 3 layers. 90

Figure 4-2: LAD multilayer classifiers: configuration 2. In order to identify faulty observations and classify them into 3 fault types, 3 LAD classifiers must be trained and applied in cascade. ..91

Figure 4-3: Classification Accuracy Comparison: Experiment 1. The best LAD based classifier accuracy obtained in experiment 1 using the Combined ratios data set at $l = 20$ compared to the accuracy of other classifiers obtained from Naresh et al. [20]. LAD achieves the second highest result. 94

Figure 4-4: Classification Accuracy Comparison: Experiment 2. The best LAD based classifier accuracy obtained in experiment 2 using the Absolute Content data set at $l = 10$ compared to the classification accuracy of other classifiers obtained from Lv et al. [6]..95

Figure 5-1: The kurtosis plot for signals obtained from bearing 3 over 35 days of testing suggest that the defect started on day 32 of operation, with the kurtosis levels beginning to surge in signals acquired during that time. The kurtosis values calculated in MATLAB were plotted using Excel...119

Figure 5-2: The kurtosis plot for signals obtained from bearing 4 over 35 days of testing reveals that a defect happens during day 20 of the test as shown in the first spike of kurtosis level. The graph later becomes flat and smooth, revealing the healing phenomenon before another incident causes another surge (Excel). 120

Figure 5-3: A plot of the STFT using MATLAB of a signal collected from bearing 3 on day 32 reveals a peak of high energy at the defective frequencies at each rotational cycle 0.03 seconds of the bearing. 121

Figure 5-4: (a) STFT of the signal that generated the first peak in the Kurtosis graph of Figure 2. The STFT plot reveals a shock occurring at a specific point in time (left). (b) Further examination of the plot reveals that the shock does not span the entire frequency range with the same amplitude (right). This rules out the interpretation that the shock is irrelevant to the state of the bearing itself (MATLAB). 122

Figure 5-5: The 12 scale DWT of signal 2120 of bearing 3 as obtained using MATLAB. The 12 scales span a frequency range of 20 KHz whereas the horizontal axis is the time scale which spans 16384 data points representing 0.8 seconds. 124

List of Acronyms

ANN	-	Artificial neural networks
AR	-	Auto Regressive
BP	-	Back Propagation
CCNN	-	Cascade Correlation Neural Networks
CBM	-	Condition Based Maintenance
CM	-	Corrective maintenance
ES	-	Expert Systems
FFT	-	Fast Fourier Transform
FFNN	-	Feed Forward Neural Networks
GA	-	Genetic Algorithms
HMM	-	Hidden Markov Models
IDEAL	-	Iterative Discriminant Elimination Algorithm
ICA	-	Independent component analysis
LDA	-	Linear Discriminant Analysis
LAD	-	Logical Analysis of Data
MILP	-	Mixed 0-1 Integer and Linear Programming
MLP	-	Multilayer perceptrons
O&M	-	Operation and Maintenance
OEM	-	Original Equipment Manufacture
PM	-	Preventive Maintenance
PCA	-	Principle Component Analysis
RBF	-	Radial Based Functions
RRC	-	Reason for Removal Code
RNN	-	Recurring Neural Networks
RCM	-	Reliability Centered Maintenance
SOM	-	Self-Organizing Maps
STFT	-	Short Time Fourier Transform
SG	-	Simple Greedy
SPC	-	Statistical Process Control

SVM	-	Support vector Machines
TSA	-	Time Synchronous Averaging
TTR	-	Time to Removal
TPM	-	Total Productive Maintenance
TQM	-	Total Quality Management
WG	-	Weighted-Greedy
WVD	-	Wigner-Ville Distribution

INTRODUCTION

For years after World War II, the advancement of industry had been tied to two factors: "Technological Advancement", which was responsible for the development of innovative products that existed only in people's dreams before the war, and "Volume Production", which put these products in the hands of consumers of every class and social status thanks to economies of scale. Operation and Maintenance (O&M) was relegated to the status of a "necessary evil" needed to keep the wheels of production turning [Smith & Hinchcliffe 2004]. This however changed after the '80s, partly due to the emergence of concepts such as product safety, warranty, and environmental concerns which became prerequisites rather than a competitive advantage. However, the main reason for this change is the realization by management that O&M costs were putting a serious dent in their bottom-line. Depending on the type of industry, O&M costs can make up between 15% and 60% of the total production cost of consumer goods [Mobley 2002]. The highest percentages are represented in iron and steel, pulp and paper, and other heavy industries [Mobley 2002].

Attention to maintenance is still increasing due to the ever rising costs of operating and supporting systems and equipment [Dhillon 2006]. The U.S. department of defence, for example, allocated $11.8 billion of its fiscal year (FY) 2009 budget for equipment maintenance [Garamone 2008]. In 1981, domestic plants in the US alone spent $600 billion in maintenance costs, an amount which has doubled in 20 years [Heng et al. 2009]. Although field maintenance has evolved significantly over the years, equipment maintenance still poses a challenge due to complexity, cost, and competition [Dhillon 2006]. Of the many problems that plague maintenance operations, we identify three that are frequently encountered in the industry:

1- Reliance on Reactive Maintenance: The largest amount of maintenance resources, in terms of costs and plant personnel, is dedicated to corrective maintenance [Smith & Hinchcliffe 2004]. In other words, maintenance is still conducted on a reactive basis. Plants that follow such maintenance strategies incur a lot of production downtime and as such suffer from the highest unit costs.

2- Recurrent Problem Repetition: This is a direct consequence of the above problem. As most maintenance resources go in to fixing problems that have already occurred, there is no time or personnel left to assess the situation and analyse the information so that the deficiency can be permanently corrected [Smith & Hinchcliffe 2004].

3- Human Error: Evidence shows that human error that occurs during intrusive maintenance intervention causes 50% of plant forced outages [Smith & Hinchcliffe 2004]. In other words, human error may occur in one of every two maintenance tasks being performed. This problem can also be tied in with the first point. The reliance on reactive maintenance inevitably leads to maintenance tasks of intrusive nature that create room for human error.

Over the past two decades a lot of research has been conducted on maintenance management strategies in an attempt to solve maintenance problems such as the ones described above; so much so to the point that there has been a proliferation of programs and acronyms that aim to provide solutions. Terms such as Reliability Centered Maintenance (RCM), Total Productive Maintenance (TPM), and Total Quality Management (TQM) have been attributed to programs that provide insight on how to best manage maintenance. One common denominator between these programs is their advocacy of proactive (preventive) maintenance and, particularly, Condition Based Maintenance (CBM). This thesis focuses on the implementation of a relatively new technology called Logical Analysis of Data (LAD) in CBM. However, before discussing the objective of this work in detail, a definition of maintenance terminology is presented.

Types of Maintenance

Maintenance is defined as any sort of action or activity performed for the purpose of retaining an item/part/equipment or restoring it to a serviceable state [Dhillion 2002]. Traditionally, maintenance operations have been divided into two categories: "Corrective Maintenance" and Preventive Maintenance".

Corrective maintenance (CM), also known in the literature as "Reactive Maintenance" or "Breakdown Maintenance" is defined as "the performance of unplanned maintenance tasks to restore the functional capabilities of failed or malfunctioning equipment or systems" [Smith & Hinchcliffe 2004].

Preventive maintenance (PM), also known as proactive maintenance, is defined as the "performance of inspection and/or servicing tasks that have been pre-planned for accomplishment at specific points in time to retain the functional capabilities of operating equipment or systems"

[Smith & Hinchcliffe 2004]. A more detailed definition states that PM is “a series of tasks performed at a frequency dictated by the passage of time, the amount of production, machine hours mileage or condition that either: (1) extend the life of an asset or (2) Detect that an asset has had critical wear and is about to fail or break down” [Levitt 2003].

The key in differentiating between the two types of maintenance is that CM actions are unscheduled whereas PM actions are premeditated. However, as obvious as the difference is, some confusion still exists between the two terms. This is due to the fact that corrective maintenance has been deeply rooted in O&M strategies in the industry for so long. In one example, a plant was rewarding its employees for their good practice in fixing outages rapidly thus limiting the forced downtime. This is a pure reactive operating philosophy; however plant personnel perceived it as preventive maintenance as they were able to prevent long outages because of effective reactive measures [Smith & Hinchcliffe 2004].

Preventive Maintenance can be further subdivided in different ways, however it is most commonly broken down to two categories: Time-based and condition-based.

Time based preventive maintenance aims at failure prevention by performing maintenance actions according to a time interval that could either be fixed (hard time) or flexible depending on the nature of the item/equipment and the technology used for maintenance time calculation. Maintenance actions here are intrusive in nature most of the time, which creates room for human error.

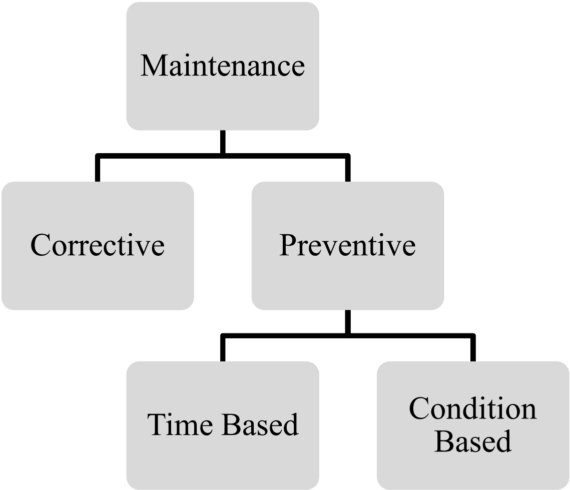

Figure i-1: Types of Maintenance

Condition based maintenance (CBM) is defined as a procedure that monitors a physical asset`s health (condition) using non-intrusive procedures and recommends maintenance action when evidence of the system's abnormal behaviour arises. CBM is also defined as "a set of maintenance actions based on real-time or near real-time assessment of equipment condition obtained from embedded sensors and/or external tests and measurements using portable equipment" [Butcher 2000].

CBM directly addresses the three problems of maintenance in the industry discussed above. Contrary to reactive maintenance, CBM actions are proactive measures that focus on resolving issues prior to the onset of failure. It is reported that 99% of mechanical failures are preceded by signs or indicators that are noticeable [Bloch & Geitner 1997, Heng et al. 2009]. CBM takes advantage of this fact to monitor the condition of an asset without interrupting its normal operation. This requires an informed analysis of performance parameters and their correlation with failure onset. With this knowledge, CBM constantly monitors the system's state and takes the necessary corrective measures to prevent system failure, thus minimizing the chance of problem repetition. These actions lead to minimized downtime, which consequently minimizes any losses that result from low productivity and long maintenance labour hours. More importantly, and contrary to all other typed of maintenance mentioned above, CBM tasks are nonintrusive and, as such, reduce the risk of human error which is a major cause of system downtime. The next section discusses the different types of CBM and its methodology before addressing the main objective of this thesis; the implementation of LAD within CBM.

Condition Based Maintenance Methodology

Before we discuss the different steps that go into the implementation of CBM, we must point out that CBM can be applied for two different purposes: diagnostics and prognostics. In order to understand the difference between these two terms, it is important that we distinguish between a "fault" and a "failure" within the context of maintenance.

A Failure is defined as the transition of a system from a state (Up) where it is able to perform a required function(s) within specified performance requirements to another state (Down) where it is unable to perform this (these) required function(s) [Marquez 2007]. Failure is therefore an event that leads to the transition of a system from the Up state to the Down state. A Fault is not an event, but a state in itself [Marquez 2007]. When a system is in its Down state, it is fundamentally also in a state of Fault. However, a system could remain in its Up state even though it is in a state of Fault. In such a case, the Fault is described as latent [Marquez 2007]. A fault can therefore be defined as an abnormal state or condition that may cause a reduction in, or loss of, the capability of a functional system to perform the required function(s). It is therefore a deviation from the system's desired or intended state.

Based on the above definitions, we identify diagnostic applications as those which deal with the detection and isolation of faults. Fault detection is defined as the identification of fault appearance, whereas isolation consists of determining the kind, time or place of appearance of the fault [Korbicz et al. 2004]. Prognostic applications, on the other hand, refer to the prediction of how soon and how likely a failure will occur [Jardine et al. 2006]. The above definitions of diagnostics and prognostics may vary between publications and within different contexts. Their tasks are sometimes interchanged, and the distinctions between them blurred. For example, some definitions of diagnostics attribute to it the task of fault identification, defined as the determination of fault size and its changeability in time [Korbicz et al. 2004] even though it overlaps the role of prognostics. In this text, the meaning for diagnostics and prognostics will follow the definitions for them stated above.

Although Prognostics promises more in terms of reducing downtime, spares inventory, and labour costs [Heng et al. 2009], the decision to apply each is dependent on the nature of the physical asset, the predictability of the failure, and the availability of input data. Ideally, Diagnostics plays a complementary role to Prognostics as it contributes to the improvement of the latter [Jardine et al. 2006]. The aim of this thesis is to apply LAD within a diagnostic application of CBM. As such further description of CBM methodology will be discussed within the context of fault diagnostics.

A CBM program consists of the following three steps: Data Acquisition, Data Processing and Decision Making [Jardine et al. 2006]. The first step involves the collection of data relative to the health of the asset. The second step analyses the data obtained from step one and processes it, extracting meaningful information from it. The third step involves using decision making and classification technologies to recommend the appropriate maintenance actions based on the analyzed data.

Data Acquisition

At the Data Acquisition stage information about the state of the system is collected. Any decision made through CBM is based on information obtained at this stage. As the definition implies, CBM relies on the monitoring of two types of data: that which depicts the system's performance and that which carries information about the system's degradation. As such, acquisition Data can be divided into 2 broad categories: Event Data and Condition Monitoring Data.

Event data express information about what happened. Examples of such types of data are the number of times the system has failed, the failure mechanism or how the failure occurred (mechanical failure, electrical failure, chemical, etc...), the cause of the failure (contamination, a wrong selection, dirt, etc...) and the action taken after the failure occurred (repairs done, replacements made, etc...). Despite the advancement in communication technologies, such data is often still recorded manually, at least at the first stage of the acquisition process.

Condition monitoring data reveal information about the system's current state and its degradation or improvement. It can be divided into 3 types: Value data, Waveform data, and Multidimensional data. Value data are single measurements taken at a specific time or instant. Examples of such values are measures of temperature, pressure, or the number of iron particles in oil. Waveform data are generally time series measurements such as vibrations signals or acoustic signals that are collected over a certain interval. Multidimensional data is data observed over 3 or more dimensions. An example of such data in a maintenance context is infrared thermographic images that measure temperature variations on a body surface. The automation of collecting such types of data has been successful due to advancement in sensor technologies.

Data Processing

The next step after acquisition is the processing of data for the purpose of extracting relevant information from it, i.e. feature extraction. Naturally, the type of data processing tool used is dependent on the type of data available. In some cases processing is not needed due to the ability to read the information directly from the data; however, often processing tools must be used to extract the important features. In the case of multidimensional data such as thermographic images, different image processing tools are used to extract the relevant information. The most relevant types of data processing to this research are event data, value data, and waveform data (vibration data) processing.

Event Data Processing

Event data analysis is often called reliability analysis [Jardine et al. 2006]. Failure times are an example of event data that can be used to determine or predict the failure of a system. The conventional method to analyse such data is to fit them with a known probability distribution. The Weibull distribution is by far the most common distribution to be used for characterizing failure events in CBM. The appeal of Weibull lies in its ability to take the shape of three common stages that most mechanical components go through in their lives: infant mortality (decreasing failure rate), normal useful life (constant failure rate) and Wear-out stage (increasing failure rate). However, some electronic components' lifetimes may exhibit an exponential distribution, which is a special case of the Weibull distribution, whereas others may exhibit gamma distributions. The disadvantage of fitting data with lifetime models such as the Weibull distribution is that the independence of the data sets is a precondition to their validity. In many cases the test for dependencies is ignored due to lack of sufficient data. Another disadvantage is that this type of data is historical, meaning that any information extracted from it regarding the current or the future condition of a system is speculative.

Value Data Processing

Value type data can be obtained via direct data acquisition or from processing waveform data. Examples of the first type of value data are measures of temperature, pressure, humidity, etc.... Such data are commonly analysed by trend analysis techniques such as autoregressive (AR) models [Jardine et al. 2006]. Another example of value data is dissolved gas readings in power transformers. These readings measure the amount of chemical breakdown of oil into hydrocarbon gases at certain environmental conditions in power transformers. Such data is commonly processed by creating standardized ratios created through expert knowledge. Value data extracted from waveform data are often correlated, which makes them inconvenient to use in statistical applications. Methods such as Principle Component Analysis (PCA) and Independent Component Analysis have been used to analyse data with high correlation and extract non-correlated data [Jardine et al. 2006]. A key disadvantage of the use of ICA and PCA techniques is the loss of transparency as the resulting data is not representative or meaningful.

Vibration Data Processing

The most common form of input data encountered in machinery diagnostics is waveform data. Monitoring vibrations using sensors (usually accelerometers) is a cost-effective condition monitoring technique in mechanical equipment. Data processing techniques are used on vibration signals to extract useful information to use in monitoring the condition of the system. Vibration signal analysis techniques are in general divided to three categories: time, frequency, and time-frequency domain analysis.

Time-domain analysis techniques extract information directly from the sensor signal in its raw form as a time series. Sometimes, this can be done by visually estimating the period of the wave signal. In most cases, however, the signal is too distorted by different frequencies, thus more sophisticated feature extraction means are necessary. The most traditional time-domain analysis techniques calculate descriptive statistics or high-order statistics from the time signal. Some common statistics are shown in Table 1.

Table i-1: Commonly used higher-order statistics. $\boldsymbol{x}$ is a signal of length $\boldsymbol{T}$, whose mean is $\boldsymbol{\mu}$, standard deviation is $\boldsymbol{\sigma}$ and expected value $\boldsymbol{E}$.

Statistic	Formula	Statistic	Formula
Mean	$\bar{x} = \frac{1}{T}\int_0^T x(t)dt$	Kurtosis	$k = \frac{E(x-\mu^4)}{\sigma^4}$
Peak Value	$\max(x(t))$	Crest Factor	$\frac{\text{Peak}}{\text{RMS}}$
Peak-to-Peak	$\frac{1}{2}\left(\max(x(t)) - \min(x(t))\right)$	Skewness	$k = \frac{E(x-\mu^3)}{\sigma^3}$
Standard Deviation	$\frac{1}{T}\int_0^T (x(t)-\bar{x})^2 dt$	RMS	$\sqrt{\frac{1}{T}\int_0^T x(t)dt}$

Each of the above statistics has advantages and disadvantages depending on the type of vibration signal and the type of fault being diagnosed. Other time-domain analysis techniques are Time Synchronous Averaging (TSA), the shock-pulse method, and the Orbits method.

Frequency domain analysis techniques transform the vibration signal to the frequency domain for a better identification of the frequency components that are of interest. The most widely used frequency analysis technique is Spectrum Analysis. Spectrum analysis derives a frequency spectrum from the time signal using the Fast Fourier Transform (FFT). The peaks found in the spectrum indicate the type of fault whereas the amplitude of the peaks indicates its severity. Several feature extraction methods may be applied to the resulting spectrum to detect the faults. One of the commonly used ones is power spectrum which is the average of the squared amplitude of the Fourier transform of the signal. Other tools are RMS levels, envelope analysis, and side band structure analysis [Jardine et al 2006]. The disadvantage of using FFT appears when the defect frequency overlaps that of the components in the machine (for example as the speed of the rotating component changes). An improved version of the spectrum called the synthesized spectrum addresses this issue; however the fault frequencies remain hard to identify [Safizadeh et al. 1999]. Other techniques include waterfall analysis, Cepstrum, and Holospectrum.

The most common disadvantage of both time and frequency analysis techniques is their inability to handle non-stationary input signals. Time-frequency domain analysis techniques were developed to overcome this omnipresent problem, as most vibration signals are non-stationary. Three major time-frequency analysis techniques are used in the industry for the analysis of vibration signals that contain large numbers of components: Short Time Fourier Transform (STFT), Wigner-Ville distribution (WVD), and wavelet transforms.

STFT solves the non-stationarity problem by dividing the signal into small segments that could be considered locally stationary. Conventional FFT is then applied to every segment. The advantage of this technique is that it reveals the transient events occurring due to impacts; however it cannot accurately display the duration of the events as it is affected by window length [Howard 1994]. The Wigner-Ville distribution is calculated as the Fourier transform of the instantaneous autocorrelation function of the input signal. The main advantage of WVD is its non-reliance on segmentation and its time-frequency resolution limitations. As such it can be used to analyze non-stationary signals without any preconditions. The disadvantage of WVD is the fact that it can produce large unwanted frequency components which can interfere with the terms of the input signal thus hindering analysis [Jardine et al. 2006]. However, many variations on the Wigner-Ville distribution have been devised to circumvent this problem. The wavelet transform is the most recent addition to time-frequency data analysis techniques.

As with the STFT and WVD, the output of the wavelet transform is a time-frequency map of the input vibration signal. The difference however, lies in the fact that the frequency and time resolution properties do not remain uniform throughout the map. The continuous wavelet transform is given by the following formula:

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\psi^*\left(\frac{t-b}{a}\right) dt$$

where $x(t)$ represents the vibration signal and the function ψ represents the mother wavelet function. The mother wavelet function can take any possible form provided it satisfies the conditions of admissibility, i.e. it must be a finite zero average oscillatory function centered at zero with a finite energy. The parameter a in the above equation represents the scale parameter which controls the size of the mother wavelet while b represents the translation (time) parameter which is responsible for the position of the wavelet as it is convoluted with the input signal.

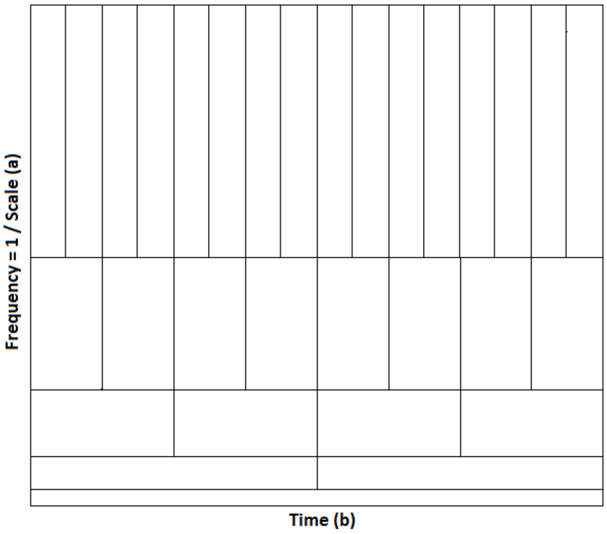

Figure i-2: Time-Frequency resolution map generated by a dyadic wavelet transform

At each scale $a = 1/f$, where f is the frequency, the mother wavelet is passed over the input signal, translated each time by a distance b. This is repeated for a variety of a's. The result is a two dimensional transform of varying time and frequency resolutions. As seen in the time-frequency resolution map shown in figure 1.2, when the mother wavelet is at a large scale a it enjoys excellent frequency resolution and poor time resolution. As the scale decreases, frequency resolution suffers but time resolution improves. This is ideal for the detection of high frequency transient responses generated by faults in machinery as these events carry limited frequency information but are extremely short in duration. On the other hand, most information is generally found at the lower frequencies which are detectable by this wavelet resolution configuration.

The resulting transform is very data heavy, especially if the continuous wavelet transform is used. In most cases, however, the discrete wavelet transform is applied. The difference is the discretized shifts in the scale and translation parameters; the signal remains continuous. Interpretation of the transform is done by plotting the Scalogram (obtained by squaring the transform output $|W(a,b)|^2$) and the phase spectrum (obtained by plotting the phase angles of the complex value $W(a,b)$.

Decision Making

The third step in implementing a CBM strategy, and the most pertinent step to this research, is the Decision Making step where an analytical model is used to assess the condition of an asset based on the information obtained from the data analysis techniques discussed above. In practical terms, this process diagnoses the health of the asset by mapping the information obtained from the physical space to the fault space (where a machine or component is classified, for example, as either faulty or functioning normally) [Jardine et al. 2006].

Different technologies have been used for decision support in CBM in an effort to replace manual diagnosis, whereby maintenance experts decide whether a machine or component requires maintenance based on visual inspection of the data processed in the previous step. The disadvantage of manual diagnosis is its reliance on experts and its vulnerability to human error. Automatic decision models offer a better alternative [Jardine et al. 2006].

Automatic decision models can be divided to two categories based on their mathematical architecture: Statistical approaches and artificial intelligence approaches [Jardine et al. 2006]. Each of those categories includes decision model technologies that operate using supervised and unsupervised learning. Models that operate by deducing a function from training data are considered supervised learning models. Training data consist of input observations, typically vectors made up of the values at specific instances of the features extracted in the data processing step, whose outputs are already known. The choice of what kind of features, or how many, to use depends on the judgement and expertise of the CBM program modeller as much as on the availability of data.

A literature review of the technologies used in decision support will be presented in Chapter 1 of this thesis.

Logical Analysis of Data

LAD is a supervised learning data mining approach that was first proposed in 1986 by PL Hammer in an article titled "Partially defined Boolean functions and cause-effect relationships" [P.L. Hammer 1986]. In this paper, a Boolean technique was introduced to identify the causes of a certain event through investigating a set of factors representing all the possible causes of that event. In a follow-up paper, [Crama et al. 1988] illustrate the technique using a medical example. Later on, LAD evolved as an approach that extracts patterns from binarized training data and formulates decision rules in order to classify new data into separate classes. The approach is implemented in three steps: Data Binarization, Pattern Generation, and Theory Formation. In some cases, a feature extraction step is also added after binarization. The binarization step involves translating the training data to a binary data set using a binarization technique that translates each numerical feature to a set of binary attributes. Each numerical feature u translates to at least one binary attribute $b_i(u)$, whose outcome is governed by a certain rule established by the values that u takes in the training data set.

The pattern generation step generates a set of Boolean rules called patterns that are distinct to the observations of one class and not the other. A pattern $\boldsymbol{p}$ is composed of *literals*; a literal is a Boolean variable x or its negation $\bar{x}$ [Boros et al. 2000]. Each binary attribute b_i in the training set can be represented in a pattern by a literal x_i or its negation $\bar{x}_i$. In its strictest sense, a pattern $\boldsymbol{p}$ of degree d is a conjunction of d literals such that it is true for at least one observation of one class and not true for all the observations of the other class. A pattern that is true for a certain observation is said to "*cover*" that particular observation. For a pattern $\boldsymbol{p}$ covering an observation belonging to a certain class, a literal x_i may be included in the pattern if the observation being covered by $\boldsymbol{p}$ has the value 1 at b_i. Similarly, literal $\overline{x}_i$ may be included in the pattern if the observation being covered by $\boldsymbol{p}$ has the value 0 at b_i. Several pattern generation techniques have been discussed in the literature; they can be categorized into enumeration, heuristic, and linear programming techniques. At a later stage, this thesis adopts a technique based on linear programming, which will be discussed in details.

The final step in the formation of the LAD decision model is the theory formation step where the patterns generated in the previous step are used to create a decision function called the discriminant. For a two-class LAD decision model, where one class is labelled positive and the other negative, a discriminant function takes the following shape:

$$\Delta(\mathbf{a}_i) = \sum_{n=1}^{N} w_n^+ \boldsymbol{p}_{\boldsymbol{n}}^+(\mathbf{a}_i) - \sum_{m=1}^{M} w_m^- \boldsymbol{p}_{\boldsymbol{m}}^-(\mathbf{a}_i)$$

Where each $\boldsymbol{p}_{\boldsymbol{n}}^+$ or $\boldsymbol{p}_{\boldsymbol{m}}^-$ is one of N positive or M negative pattern found in the previous step respectively. $\boldsymbol{p}_{\boldsymbol{n}}^+(\mathbf{a}_i) = 1$ if a binarized observation $\mathbf{a}_i$ is covered by pattern $\boldsymbol{p}_{\boldsymbol{n}}^+$ and $\boldsymbol{p}_{\boldsymbol{n}}^+(\mathbf{a}_i) = 0$ otherwise. The same property applies to negative patterns. The values $w \geq 0$ are weights, assigned to each pattern to enhance its importance. If the output of the above discriminant function is positive for a certain observation, then that observation belongs to the positive class. If the output is negative, then the observation belongs to the negative class. If the output is zero, then no decision can be made.

The above is an explanation of the steps involved in the generation of a two class LAD based decision model. Within a CBM context, the above model can be used for fault diagnosis and the identification between two faults. For the identification of multiple faults, a multi-class configuration is needed. A literature review of what has been accomplished so far with LAD is presented in Chapter 1 of this thesis.

Problem Statement

We started this introduction by discussing how maintenance has become essential to the bottom line of companies and the competitiveness of their products in all kinds of industries. We mentioned 3 common problems that these companies face and explained how CBM can help resolve them. However, two issues that affect maintenance in the industry do not receive enough attention as they should:

1- Improper implementation of PM actions – Although this seems at first to contradict with the first problem discussed in the thesis, a parallel needs to be drawn with the question: are the current PM actions right for the situation at hand? The extent of this problem is emphasized by the statistic that between 5 to 20 % of plants' existing PM tasks have no effect on the outcome [Smith & Hinchcliffe 2004]. In some cases, the maintenance actions are outdated, while in others the PM tasks are too conservative in the sense that they are performed more often than needed. However, it is also common to find plants focusing their efforts on the implementation of sophisticated condition monitoring technology without getting enough analysis or feedback on the relevance of the data collected, and sometimes, at the expense on more useful simpler solutions.

2- Rationale for PM actions is unknown in most tasks – This problem ties in closely with the previous point. It is safe to say that a major portion of preventive maintenance strategies applied in the industry is based on expert knowledge and past experiences of maintenance personnel. For example, decisions on whether a certain part or equipment requires a certain type of maintenance is based on certain threshold values applied to processed data features by in-house technicians or by standardized procedures. In many cases the reasons behind these values are unknown to the technicians applying them and the rationale that led to their adoption is not readily available to them. In fact, the aviation industry and nuclear industry are the only industries that require all PM tasks to be documented and approved [Smith & Hinchcliffe 2004].

Most of the technologies used so far for decision support in CBM do not properly address these two problems. Although existing CBM strategies allow for the replacement of Time Based Maintenance procedures which would alleviate the conservative maintenance problem, most CBM decision models tend to automate the maintenance decision process without providing any added knowledge to the maintenance operation. There is, therefore, a need for a decision model that can offer insight that helps guide and update the maintenance procedure and provide rationale for the maintenance technicians on the ground.

The main purpose of this thesis is to address the two problems discussed above by proposing the implementation of LAD as a new decision making technology in CBM. LAD possesses two key advantages over other decision making technologies used in CBM:

1- LAD is a non-statistical approach – Most commonly used decision making approaches assume that the input data belong to a certain statistical distribution in order to proceed. However, such assumptions are not always true, and finding the correct statistical distribution for the input data is not easy either. Incorrect assumptions result in an unusable decision model that could compromise a CBM program. Additionally, the statistical nature of these approaches imposes conditions on the nature of input data used: mainly that they be independent and identically distributed (i.i.d.). The fact that LAD is a non-statistical approach removes the need for any statistical assumptions regarding the input data or any restrictions on the type of data used.

2- LAD generates interpretable patterns – Decision models generated by LAD consist of patterns, which are meaningful characterizations of attributes that group observations into one common class. Each attribute is a representational form of a value or interval of a feature. As such, the output of such decision models can be traced back to the feature values that resulted in the categorization of a specific observation into a certain class. Users of LAD can thus take advantage of the "interpretability" power of this decision making approach.

The above two properties of LAD suggest that it is a suitable technology for implementation in CBM as it helps address problems that until now have been difficult to solve through CBM in the industry. LAD imposes no restrictions on the type of information that can be used towards forming a diagnostic decision model. More importantly, diagnostic decision models created by LAD consist of patterns that are interpretable by maintenance personnel. This could provide a wealth of resources to maintenance experts that could be devoted towards solving the two issues discussed above and improving the CBM program.

Objectives

LAD has achieved considerable success in diagnostics in medical applications as noted in [Alexe G. et al. 2004], [Alexe G. et al. 2005], [Alexe G. et al. 2006], and [Abramson et al. 2005].

However, its application in the field of maintenance has only recently been tested at École Polytechnique de Montreal in [Salamanca & Yacout 2007] and [Yacout 2010]. A full study on the implementation of LAD in CBM does not exist prior to the research done within the context of this thesis. As such, the main objectives of this research can be summarized in the following 3 points:

1- Study the applicability of LAD in different CBM situations requiring special considerations regarding the types of input data and maintenance decisions.

2- Adapt the LAD methodology to the particular requirements that arise from different CBM applications.

3- Improve the LAD methodology in line with the above two objectives in order to increase diagnosis accuracy and result interpretability.

The novelty of the research presented in this thesis is (1) the application of LAD to CBM for the first time in applications that stand to benefit from the advantages that this technology provides; (2) the novel modifications to LAD methodology, particularly in the areas of pattern generation and adaptation to multi-class applications in order to improve its performance within the context of CBM. Chapter 1 of this thesis presents a literature review of the decision making technologies used in CBM and a review of the current progress of LAD as a decision model. A detailed description of the methodology followed to achieve the above goals and a plan depicting the organization of this thesis are presented in Chapter 2.

Chapter 1 LITERATURE REVIEW

As the aim of this thesis is to apply LAD as a new diagnostic decision making technology and to alter its design in order to improve its performance within CBM, it is appropriate to provide an overview of the diagnostic decision making technologies used so far and to review the state of the art in LAD. This chapter is therefore divided into two parts: The first part presents a review of the literature regarding diagnostic decision making technologies used in CBM. The second part provides a review of the progress of LAD as a classification and diagnostic approach throughout the literature.

1.1 Diagnostic Decision Models

As mentioned in the Introduction, diagnostic decision models can be divided into two categories based on their inherent mathematical architecture. Hence we differentiate between Statistical Approaches and Artificial Intelligence based approaches.

1.1.1 Statistical Approaches

A known statistical decision tool is statistical process control (SPC), or trend analysis, which uses statistical and graphical measurements to detect abnormalities in operation [Evans and Lindsay 2004]. This simple tool has been used extensively in the industry especially since the rise of the six sigma process improvement concept. Tools such as control maps and histograms facilitate the detection of faults. An example of the use of this method in CBM applications is found in [Fugate et al. 2001], where x-bar and s control charts, obtained from the difference between vibration signals obtained from an accelerometer and the signal estimated by an AR model, are calculated and monitored to detect faults.

Hypothesis testing is one of the most extensively used statistical approaches in pattern recognition. Traditionally, hypothesis tests separate input observations in to two classes: H_0, representing the existence of a fault, and H_1 representing normal operating conditions. Multi-class hypothesis testing is also possible. The prerequisite to the use of hypothesis tests as a decision model is knowledge of the conditional density function of each class. As such, this method requires supervised learning in order to determine those density functions. Several types of test statistics may be used to test for the H_0 hypothesis. An example is found in [Ma et al.

1995] who developed a bearing fault detection algorithm using, as input, vibration signals that were assumed stationary and as a decision model, a Hypothesis test with the Neyman-Pearson test statistic. The amplitude of the vibration signals was modeled as a normal distribution of zero mean and a variance σ^2. The variance of the normal (non defective) observations was determined experimentally using vibration observations that were assumed independent and identically-distributed (i.i.d).

In cases where a normal distribution for both classes cannot be assumed, the boundary function resulting from tests such as Neyman-Pearson or Bayes become complicated functions that are hard to calculate. This problem could be alleviated by fixing the shape of the boundary function to be either linear or quadratic, which makes the calculation of the coefficients of that function much easier. However, the disadvantage here lies in the fact that in most cases the boundary lines separating classes may be too complicated to model using simple linear or quadratic equations. Piecewise boundary (discriminant) functions is a solution to this problem, whereby a series of hypothesis tests result in a number of linear boundary functions that model bits and pieces of the real boundary between the two classes [Fukanaga 1990].

Another statistical approach is cluster analysis, which is an unsupervised learning multivariate approach that classifies a set of signals into clusters having the same characteristics. Clustering algorithms differ with respect to their architecture and in terms of the distance measure used to determine the similarity of one observation to the other. One type of clustering architecture is called hierarchical clustering. An example is the top-bottom architecture which starts by assigning a cluster to each observation in the input observation set and then gradually eliminating unneeded clusters until only k clusters are left, where k is the number of classes. The bottom-up architecture starts with the assumption that all observations belong to one cluster and then splits that cluster until k clusters are obtained. A common hierarchical clustering algorithm is the nearest neighbour algorithm which fuses similar groups with each other based on a distance measure. [Isermann & Balle 1997] reviews a number of papers using nearest neighbour algorithm for fault detection. [Staszewski et al. 1997] used clustering analysis with the Mahalanobis distance on compressed Wigner-Ville transforms of the vibration signal to detect faults in gearboxes.

Partition clustering algorithms determine the clusters representing the k classes all at once. One partition clustering algorithm that has been used in fault detection is the $k-means$ algorithm. [Jamaludin et al. 2001] used $k-means$ clustering data obtained from stress waves to detect component defects in slow speed bearings. Other common distance measures are Euclidian distance, Kullback-Leibler distance and Bayesian distance [Jardine et al. 2006].

Another application of cluster analysis in machinery fault diagnosis was presented in [Sun et al. 2004] where features obtained from rotor bearing vibration signals where fused in to a two dimensional space using neural networks. Then, clustering was used to obtain a piecewise linear boundary function separating the 6 classes representing the 5 fault types and the normal state. Such a classification method is called Linear Discriminant Analysis (LDA) using Euclidian distance; however, traditionally, the Fisher criterion is used in LDA.

The disadvantage of clustering is that it is not always guaranteed that a global minimum is reached among elements of the same group, which may sometimes lead to the wrong clustering of elements. In addition, when the selection of cluster centers is done randomly, the outcome of the classification may differ for the same data set from one trial to the other.

Support Vector Machines (SVM) is a supervised learning statistical decision making approach which finds the optimum hyper-plane that forms the boundary between two classes by maximizing the distance between the hyper-plane and the closest point to the boundary. This is achieved through convex quadratic programming optimization algorithms. SVM has been discussed thoroughly in the literature in [Steinwart & Christmann 2008], [Kecman 2005] and [Diederich 2008]. SVMs are known to be universal approximators of any multivariate function to any degree of accuracy [Kecman 2005]. SVM became popular in the early 1990s with advancements in computer computational power [Kecman 2005]. SVM was devised to overcome the advantages of traditional statistical approaches which rely on modeling the probability distributions of data sets as Gaussian functions and using the maximum likelihood methodology to estimate the parameters of the function that estimates the boundaries of the classification. SVM is nonparametric in the sense that the number of parameters composing the resulting function is dependent solely on the training data and is not predefined [Kecman 2005].

SVM essentially is a linear classifier as it creates a hyperplane that can separate two linearly separable classes at a maximum separating distance. However, nonlinear cases can also be classified through the transformation of the original input space into a higher dimension feature space [Kecman 2005]. The resulting function approximating the boundary minimizes a certain risk function $R(w)$ where w is the support vector of weights in the SV machine (Structural Risk Minimization).

The use of SVM in CBM was discussed in [Christian et al. 2007], where after the collection of a set of indicators from vibration signals using ICA (independent component analysis), an SVM with two classes was used to detect the existence of bearing faults. In [Abbasion et al. 2007] SVMs were used for detecting and classifying bearing fault into 7 types. The inputs in this case where 2 time domain wavelet de-noised vibration signals modeled using negative log Weibull probability density functions. Similarly, [Widodo et al. 2007] used SVMs for fault detection and classification, applying ICA to extract features from frequency and time domain indicators in induction motors.

Another commonly used statistical approach in CBM is Hidden Markov Models (HMM). [Ocak & Loparo 2001] used HMM for the classification of defects in ball bearings; where input observations to the HMM model were indicators obtained from processing vibration signals. A linear autoregressive (AR) function was used to model the envelopes of filtered vibration signals which were cut in equal windows. For each of the four possible states of the ball bearing (normal state, inner race fault, outer race fault and ball fault), an HMM was built, where a weighted sum of 2 Gaussian distributions was used to model the observation probability distribution of each state.

In [Li et al. 2005], HMM was used for the detection and classification of 4 fault types in rotating machinery using, as input, features obtained from different waveform data processing tools. Three feature vectors from each feature type were used to model three different classification schemes. The Baum-Welch algorithm was used to build the HMM models representing each of the five classes. [Xu & Ge 2004] used a similar approach for fault diagnosis using features obtained from the wavelet transform.

1.1.2 Artificial Intelligence Approaches

Methods based on artificial intelligence (AI) are reported to have an "improved performance over conventional approaches [Jardine et al. 2006]." AI approaches attempt to mimic the architecture of human reasoning which is based on both generalization and memorization. By far, the most popular AI method is artificial neural networks (ANN), an approach which has been used extensively in CBM over the past decade.

Neural networks copy the anatomical structure of the nerve cells (neurons) which constitute the human brain. The mathematical equivalent of the neuron, the perceptron, is a mapping with several inputs and outputs. The mapping function, called an activation function, can take several forms. Traditionally, a step function, called the Heaviside function, is used for activation; this transforms the perceptron to the linear classification hypothesis test discussed previously in statistical approaches. Another activation function is the sigmoid function which introduces nonlinearity to the mapping. If the *a priori* probabilities are input into the perceptron, then it is transformed to the Bayesian classifier. If the *a priori* probabilities are normal distributions, then the perceptron becomes essentially the linear classifier again.

As discussed earlier, the linearity of the perceptron can be restrictive. The feasibility of using linear classifiers depends on the dimensionality of the problem. This implies that using linear classification methods becomes more restrictive with the increase in problem dimensionality. It is for this reason that the use of the perceptron alone is not sufficient, whereas a complex layered architecture of such nodes allows a proper approximation of a complex non-linear model.

Many possible configurations can be constructed from complex layered networks of perceptrons. In modeling such a network three unknowns must be taken into consideration: the architecture of the network, the number of perceptrons to be used, and the weights of the input and outputs of each perceptron/node. The simplest possible architecture and the most used in fault diagnosis is the feed forward neural network (FFNN) which allows for information to flow in one direction only. Particularly, the multi-layer perceptron (MLP) is a special type of FFNN which is abundantly used in all forms of pattern recognition problems.

Multilayer perceptrons (MLP) are formed by placing multiple perceptrons in three or more levels. The feed-forward three levelled MLP network is the most common of all since this configuration is sufficient to adapt to any classification problem, no matter how complex. MLP is a very flexible configuration; it can be modeled to be purely memory based or both memory and generalization based. Memory based networks use a heuristic algorithm to learn the configuration and weights of the network whereas memory and generalization based networks typically rely on the back-propagation (BP) algorithm to train them. BP is a supervised learning algorithm that gives good results but has the disadvantage of having slow convergence [Jardine et al. 2006]. [Subrahmanyam & Sujatha 1997] used an MLP network for fault diagnosis in ball bearings using features obtained from time domain vibration signal processing tools. They compared the BP supervised learning MLP to an unsupervised learning adaptive resonance theory ART2 neural networks with faster learning time. The study concluded that the unsupervised learning network is 100 times faster to model with excellent reliability in fault detection; however the BP based MLP showed superior classification power in the multi-class case.

A disadvantage of MLP is the difficulty of determining the number of nodes and layers required to model the classification boundary. Cascade correlation neural networks (CCNN) bypass this problem and, as such, are more suitable for online training scenarios. [Spoerre 1997] tested CCNN in bearing fault diagnostics using the cascade correlation algorithm (CCA) to construct a neural network using indicators obtained from an AR model used on raw vibration signals. The resulting CCNN network is the minimum structure necessary to correctly classify a bearing as either normal or imbalanced.

Another common configuration of ANN is radial based functions (RBF**).** Whereas MLP seek to separate classes via hyperplanes, RBF encircle the classes using radial functions placed at each observation and interpolated in order to achieve a local approximation. An additional difference is the way the network is configured. RBF networks consist of one layer of neurons. Unlike MLP, the number of neurons in a layer depends on the number of observations on which the approximation is based. The activation function of each neuron in the network is different from the other since each of these functions is centered around one observation point. This means, in addition to finding the weights of the inputs and outputs of the neurons, the coefficients of the activation functions must also be found. As a result, the number of coefficients that must be calculated is much higher than in MLP. If the number of possible observations is very high, it is normal to assume that the number of neurons in the network will also be high. Consequently, the number of unknowns will be high as well. [Baillie & Mathew 1996] compared RBF neural networks to BP trained MLPs as well as traditional linear autoregressive models in the diagnosis of faults in rolling element bearings. The results of the study showed that BP trained MLP perform better, are more reliable, and are less complex than RBF.

Other ANN architectures include recurring neural networks (RNN) which have a more complex structure than FFNN as they allow propagation of information forward and backward. [Yam et al. 2001] applied RNN in CBM to develop a decision support system that includes diagnostics. Counter propagation neural networks have also been used in CBM.

Unsupervised learning ANN algorithms have also been developed and discussed in CBM literature. A common unsupervised learning algorithm is the self-organizing maps (SOM). [Saxena & Saad 2004] used SOM for the identification of 8 different types of faults in bearings. [Wu & Chow 2004] used SOM-based RBF neural networks for the detection of faults in mechanical machines.

The most cited disadvantage of neural networks is the black box concept which characterizes its architecture. As such, there can be no physical explanation of how the trained model, and consequently the classification decision, came to be [Jardine et al. 2006, Tu 1996].

Another AI classification approach is expert systems (ES) based software that attempts to mimic the reasoning of a human expert in a specific domain. A common reasoning method applied in such software is rule-based reasoning, whereby an expert presents the problem and his knowledge in the form of a set of rules that are stored by the expert system and used for classification. Other reasoning methods include negative reasoning, case-based reasoning and model-based reasoning [Jardine et al. 2006]. The disadvantage of a decision method based on ES is the potential exponential explosion in the number of rules as the number of input variables increases [Jardine et al. 2006]. Another disadvantage is the involvement of human beings which often leads to imprecise knowledge and inaccurate reasoning. It is for this reason that ES are often combined with tools that can measure uncertainty and curb its effect. An example is fuzzy membership functions which play the role of blending objectivity with flexibility where there is uncertainty. [Monsef et al. 1997] and [Lee et al. 2000] used fuzzy rule-based expert systems for power system fault diagnosis. [Liu et al. 1996] devised a fuzzy expert system for bearing fault diagnosis.

Fuzzy logic has been combined with several other AI and statistical techniques to solve the problem of uncertainty of input data. [Yang & Wang 2006] used a fuzzy neural network approach for fault diagnosis in the bearings of electric machinery. The method used vibration signals as input and the BP method to train the FFNN.

Many papers discuss combinations of different classification tools in building a fault diagnosis decision model. For example, [Hu et al. 2001] discussed the integration of ES with a BP trained MLP neural network in machine tool diagnosis in order to avoid the disadvantages of each of those techniques used alone.

Another AI approach used in CBM is Genetic Algorithms (GA) which are a class of evolutionary algorithms that mimic evolutionary biology such as inheritance and mutation. Genetic algorithms are essentially heuristic global search techniques. GA has been used for fault diagnosis in combination with other techniques such as neural networks in [Saxena & Saas 2007] and SVM [Samanta 2004].

1.1.3 Other Approaches

Some decision models do not fall under either of the two categories mentioned above. We discuss a few of them in what follows. Model based approaches for fault-detection have been used for more than 20 years [Isermann 2005]. These methods express the dependencies between measureable signals using mathematical process models. Mathematical modeling attempts to find a model that closely resembles the real system. In [Loparo et al. 2000] a model-based approach is adopted for the detection of bearing faults caused by rub impact. System identification techniques are known for parameter estimation and have also been studied for fault diagnosis in [Simani et al. 2003]. Mathematical model-based approaches can be effective if a correct model is built. In [Saitta et al. 2005] data mining techniques were used to choose the identification model that best suits a system, from a multitude of models generated by a stochastic global search algorithm. However, with the increase in the number of input variables, the generation of a model becomes more difficult. Mathematical model-based approaches can sometimes use statistical or AI approaches to determine the coefficients of the decision function, as explained in [Isermann 2005, Frank et al. 2000].

Decision trees are another classification approach often used for fault detection in CBM. Decision trees can be considered as statistical classifiers as they rely on the calculation of probabilities from training data to form the classification tree. The most common decision tree algorithm is the C4.5 algorithm. [Sun et al. 2007] used the C4.5 algorithm along with PCA as a data processing tool for the detection of faults in rotating machinery. 18 indicators (7 obtained from the time domain and 11 from the frequency domain) are extracted from vibration signals for training. PCA is used to reduce these to 6 features. The C4.5 algorithm is then used to create the decision tree. [Sun et al. 2007] compared their method to BP FFNN and found that the C4.5 algorithm combined with PCA results in better accuracy and shorter training time. The disadvantage of the decision tree method is that it finds linear relationships between input variables and does not pick up on non-linear relations.

1.2 LAD Literature Review

LAD methodology has evolved and diversified since its inception. Most implementations of LAD are divided into the three main stages mentioned in the Introduction: Binarization, Pattern Generation, and Theory Formation. However, a number of different configurations also exist, namely the discrete data based, and the multi-class LAD configurations. This section of the chapter starts with a review of the techniques used to implement each of the three main stages of LAD. Later, a review of the other configurations found in the literature is presented.

1.2.1 Data Binarization

Features obtained from event and condition monitoring data can generally be divided into 4 categories: binary, discrete unordered, discrete ordered, and numerical. As LAD was conceived to treat observation vectors in their binary form, the most logical way to adapt it to non-binary data is to transform all non-binary features to binary attributes. The binarization of discrete unordered, discrete ordered, and numerical features follows different rules. However, every transformation requires cut-points that separate the values of the resulting attributes to 1's and 0's.

Discrete unordered features, such as the color of a light emitting diode (LED) indicator on a machine (e.g. $green, yellow, red$) can easily be binarized by allocating a binary variable to each discrete value. For example, the feature u describing the LED indicator color green is translated to a binary attribute b_i whose cut-point is defined as $u = green$, such that:

$$b_i(u) = \begin{cases} 1 & if \quad u = green \\ 0 & if \quad u \neq green \end{cases}$$

The binarization of the discrete unordered feature "LED Color" therefore results in 3 binary attributes represented by the variables b_1, b_2, and b_3 with the respective cut-points $u = green$, $u = yellow$, and $u = red$.

Discrete ordered features, such as a technician's level of Experience (e.g. $Novice, Advanced, Expert$) can easily be binarized by a number of Boolean variables equal to the number of discrete values the feature can take less one. The highest discrete value can be implied through the other Boolean variables. For example, feature u describing the technician's expertise as $Novice$ is translated to a binary attribute b_i whose cut-point is defined as $u = Novice$, such that:

$$b_i(u) = \begin{cases} 1 & if \quad u > Novice \\ 0 & if \quad u \leq Novice \end{cases}$$

As such, the binary attributes describing the feature "technician's expertise" are: b_1 and b_2 with respective cut-points $u = Novice$, $u = Advanced$.

Most features encountered in condition monitoring of mechanical equipment are actually numerical in form. The adaptation of LAD to numerical features was proposed in [Boros et al. 1997], which set the theoretical foundation for the use of LAD in many different fields. The procedure for binarizing a numerical feature u requires the alignment of all values α taken by u in the training data in descending order as follows: $\alpha^{(1)} > \alpha^{(2)} > \ldots > \alpha^{(n)}$ where n is the total number of distinct values taken by u in the training data set. It should be noted that $n \leq N$ where N is the total number of observations in the training data set. We differentiate here between the training data observations belonging to the first (positive) class $i \in S^{+}$ and the observations belonging to the second (negative) class $i \in S^-$. Cut-points are then introduced between each pair of values $\alpha^{(i)}$ and $\alpha^{(i+1)}$ for which there exists observations $\alpha' \in S^+$ and $\alpha'' \in S^-$ where $\alpha' = \alpha^{(i)}$ and $\alpha'' = \alpha^{(i+1)}$ or vice-versa. The cut-point can be calculated in several ways, but most commonly are the average of the two values. Thus, for a numerical feature u, a set of binary attributes is obtained with cut-points $a_1, a_2, \ldots, a_p$ such that $a_i = \left(\alpha^{(i)} + \alpha^{(i+1)}\right)/2$:

$$b_i(u) = \begin{cases} 1 & if \quad u \geq a_i \\ 0 & if \quad u < a_i \end{cases}$$

The total number of binary attributes describing a numerical feature depends on the number of transitions between distinct values from positive to negative observations and vice versa, i.e. it is equal to the number of cut-points.

A variation exists on the procedure for binarizing numerical data described above. In [Boros et al. 2000], numerical attributes were binarized with two different types of binary variables. The first type, level variables, is obtained through the process explained above. The second type is called interval variables. As their names suggest, these variables are defined by the intervals between every two cut-points belonging to the same numerical feature. Interval binary variables are composed of the fusion of two level variables belonging to the same numerical feature.

These attributes are not indispensable as they hold redundant information. Although they do improve robustness with the increased Hamming distance between different values, this is not the main purpose of their use. Patterns obtained using interval binary attributes have double the complexity of those obtained using level variables. This is due to the fact that a literal representing an interval attribute contains double the amount of information of a literal representing a level attribute. This is useful in pattern generation approaches that require long computational times.

Feature Selection

In some applications, the feature binarization step results in a large data-heavy binarized training set. A set reduction technique, such as set covering problem solving, is often used to reduce the dimensions of the training set. This must be done without any loss in information that may affect the training process of the LAD decision model. As this is an optimization problem that is NP-

Hard, as proven in [Boros et al. 1997], the procedures proposed in the literature for treating it are heuristic.

One common greedy heuristic algorithm for set reduction using the plain consistency rule is found in [Chvatal 1979, Moreira 2000]. Another heuristic algorithm proposed by [Almuallim & Dietterich 1994] is less demanding in terms of execution time. It is similar in concept to the decision tree algorithm. The procedure involves splitting the observations into clusters with respect to a certain binary attribute which is chosen using a merit function. The algorithm stops when all the clusters contain observations of the same class in them. Three different merit functions were proposed in [Almuallim & Dietterich 1994]: entropy measure, Simple Greedy (SG) measure, and the Weighted-Greedy (WG) measure. Whereas the WG produced the best results according to [Almuallim et al. 1994], it was also the least efficient in terms of execution time. SG was therefore judged as the best option.

[Boros et al. 2000] suggests two other ways to modify the above minimization problem to improve the quality of the support set achieved. [Alexe et al. 2007] Also study the use of support sets for the reduction of the binary training data set. [Moreira 2000] developed another algorithm for this problem that consumes less computational time and could be extended to the multi-class case. IDEAL (Iterative Discriminant Elimination Algorithm) pursues a top-down approach in finding the support set from the initial set of binary attributes. This algorithm differs from those above in that it treats binary attributes based on their origins.

The performance of the above algorithms for reducing the number of binary attributes depends on the nature of the data available and the number of attributes needed to guarantee the consistency of the binary data mapping. If few attributes are needed, then incremental greedy algorithms are more efficient, whereas when a large number of attributes is required, the IDEAL algorithm performs better.

1.2.2 Pattern Generation

Pattern generation is a crucial step in the implementation of a LAD decision model; for this reason, extensive research has been done on this topic. Different types of patterns exist in the literature possessing different properties. Pattern generation techniques can be classified according to the types of patterns they produce. This section discusses the different types of patterns that can be generated and the algorithms used to generate them.

To recap the definitions in the Introduction, patterns are Boolean rules that are distinct to the observations of one class and not the other. A pattern $\boldsymbol{p}$ is a conjunction of *literals* where a literal is a binary variable x or its negation $\overline{x}$. Each pattern $\boldsymbol{p}$ of degree d is a conjunction of d literals. A pattern is true for at least one observation of one class and not true for all the observations of the other class. A pattern that is true for an observation is said to *cover* that observation. The output of the pattern generation step is two sets of patterns covering the training data observation sets S^+ and S^- respectively.

Four special types of patterns exist: *prime*, *spanned*, *strong* and *maximal*. These types are not all mutually exclusive, i.e. some patterns can belong to more than one type. A *prime* pattern has the least number of literals possible such that if any literal is dropped, it will cease to be a pattern. A pattern is qualified as *spanned* if for the same covered observations, it is composed of the maximum number of literals possible, such that if another literal is added, then it ceases to be a pattern. A *strong* pattern $\boldsymbol{p}_i$ is one for which no other pattern $\boldsymbol{p}_j$ exists such that the set of observations in the training set covered by $\boldsymbol{p}_i$ is a subset of the set covered by $\boldsymbol{p}_j$. [Hammer et al. 2004] offers detailed description of these 3 types of patterns. A *maximal* pattern is one which has the most coverage among those patterns covering a certain observation in the training set.

Patterns can be judged based on 3 criteria: degree, prevalence, and homogeneity. As previously explained, the degree of a pattern refers to the number of literals it is composed of. A pattern with a high degree is likely to have a lower coverage, whereas a low degree pattern is more general. The disadvantage of high degree patterns is an increased likelihood of unclassified observations, whereas low degree patterns may suffer from misclassifications.

The prevalence of a positive (negative) pattern is measured as the ratio of positive (negative) observations covered by the positive (negative) pattern to the total number of positive (negative) observations in the training set. For example, a positive pattern which covers 5 positive observations from a total of 25 positive observations found in the binarized training set is said to have a prevalence of 20%.

This definition of a pattern as described above is idealistic in the sense that it prohibits a positive (negative) pattern from covering any negative (positive) observations. However, this definition is sometimes bent for cases when the data is known to include noisy observations. For that purpose it is sometimes allowed for the patterns of one class to cover a limited amount of observations from the other class. The property that describes this phenomenon is called the *homogeneity* of a pattern and is calculated as the ratio of the observations that the pattern covers within the positive class to the total number of observations it covers. For patterns defined in the strictest sense, the homogeneity of positive patterns is 1 and that of negative patterns is 0.

Pattern generation techniques found in the literature are based on three general problem solving techniques: enumeration, heuristics, and linear programming.

Two enumeration techniques exist for the generation of positive and negative prime patterns from a training data set: the bottom-up and the top-down approaches. The bottom-up approach follows a lexicographic order in generating the patterns in order to reduce the amount of computations necessary. The number of possible patterns that could be generated using this approach increases exponentially with the number of binary attributes that constitute an observation vector. For n attributes, the total number of candidate patterns to be searched is calculated as:

$$\sum_{i=1}^{n} 2^i \cdot \binom{n}{i}$$

For example, if a binarized training set is composed of 64 binary attributes, the total number of candidate patterns to be searched is almost infinite. To reduce the computational time to a manageable level this procedure limits the maximum degree a generated pattern can have. The advantage of this approach is that it favours the generation of short *prime* patterns which are simpler to obtain, are more global since they cover more observations, and are easily interpretable. However, with the maximum degree d preventing the study of all possible patterns, some observations in the training set risk not being covered by a pattern. Here enters the role of the top-down pattern generation approach. [Boros et al. 2000] applies the two approaches in tandem; the bottom-up algorithm is used first and then the top-down algorithm follows to generate patterns for all uncovered observations. The top-down pattern generation approach starts by considering all uncovered observations as patterns, of degree n equal to the number of binary attributes in the training set. For each of those patterns, literals are removed one by one, until a *prime* pattern is reached.

The disadvantage of the above algorithm is the exponential time needed to generate a set of patterns, especially when the number of degrees required to reach a meaningful pattern is high. The long execution time of the algorithm led [Boros et al. 2000] to use interval binary variables in addition to level variables when binarizing the training data.

[Hammer et al. 2004] describes polynomial procedures that recognize patterns that are Pareto optimal to one or more of the three pattern properties discussed above. The authors evaluate the performance of different pattern types and draw conclusions that support the performance of strong patterns over prime patterns. In [Alexe et al. 2008], an enumeration based algorithm was described for generating *strong prime* patterns and *strong spanned* patterns. The general idea of this algorithm is to find all the possible patterns that could be generated from the training data set and then decide which of them to keep based on their degree and their coverage. The result of the enumeration is therefore the set of all possible patterns and their positive and negative coverage. The isolation of the *strong prime* patterns becomes a matter of checking which ones satisfy the conditions. [Alexe et al. 2008] also compared between comprehensive (*strong spanned*) patterns and comprehensible (*strong prime*) patterns. The results of their comparison showed better classification accuracy for comprehensive patterns, but with a non-significant difference statistically.

[Hammer & Bonates 2006] described algorithms for generating *maximal* patterns using heuristics and linear approximation. One reason for deciding to use *maximal* patterns is their relative small number, as the maximum number of *maximal* patterns that could be generated from an observation set is equal to the total number of observations in a training set. This significantly lowers the computational load exerted into pattern generation. The *maximal* pattern generation problem can be defined as an optimization problem that attempts to maximize the coverage of a certain pattern covering a certain observation. [Bonates & Hammer 2008] discussed in more details the techniques to solve the optimization problem. The heuristic algorithms proposed in that paper lead to either *prime maximal* and *strong maximal* patterns. The other possible solution discussed is the replacement of the objective function with its best linear approximation (BLA), thus transforming it to a linear set covering problem which can be solved using conventional means. The results of testing in [Bonates & Hammer 2008] showed that BLA is the superior performer in all classification problems involving less than 100 binary attributes. Beyond that threshold, it became more efficient to use heuristics for finding patterns, with the *prime maximal* heuristic algorithm being more sensitive to the number of binary attributes whereas the *strong maximal* algorithm more sensitive towards the number of observations.

The most recent development in pattern generation is an algorithm proposed in [Ryoo & Jang 2009]. Ryoo et al. transforms the problem of generating any type of pattern into a linear set covering problem that can be solved by linear programming without any BLA approximations. Many tools that can solve Mixed 0-1 Integer and Linear Programming (MILP) problems exist on the market, such as the IBM ILOG CPLEX software [ILOG 2003], the Coin-OR Linear Programming solver (CLP) [COIN-OR 2004], and the LP_SOLVE solver [Berklaar et al. 2004]. The performance of the pattern generation technique proposed by [Ryoo & Jang 2009] in terms of execution time and classification accuracy is promising. The progress made on LAD methodology in this thesis builds on the techniques described in [Ryoo & Jang 2009].

1.2.3 Theory Formation

The output of the pattern generation step is two sets of patterns for the two observation sets, S^+ and S^-, that make up the training data set. From these patterns, a decision function called the discriminant is created. Most implementations of LAD adopt the discriminant function form shown in the introduction:

$$\Delta(\mathbf{a}_i) = \sum_{n=1}^{N} w_n^+ \boldsymbol{p}_{\boldsymbol{n}}^+(\mathbf{a}_i) - \sum_{m=1}^{M} w_m^- \boldsymbol{p}_{\boldsymbol{m}}^-(\mathbf{a}_i)$$

The discriminant is created from weighted patterns. The weight assigned to each pattern is calculated in different ways. The simplest, yet not the most optimal, is to set all weights equal to 1. One other method is to compute the weight of a pattern as a normalized function of the number of observations it covers. This is done by dividing the number of observations covered by a pattern by the coverage sum of all patterns of the same class. Consequently, the normalized weights avoid the problem posed in cases where the number of positive patterns outweighs that of negative patterns or vice versa. A variation on the above method would be:

$$w_i^+ = \frac{\left(Coverage\left(\boldsymbol{p}^+\right)\right)^\eta}{\sum_{i=1}^{N} Coverage\left(\boldsymbol{p}^+\right)}$$

The exponent η emphasizes the importance of the patterns' coverage in the classification. Another criterion for assigning weights to patterns is the hamming distance between a particular positive (negative) pattern and its closest negative (positive) counterpart. Other weight calculation methods are described in the literature ranging in sophistication up to linear programming techniques.

If the output of the above discriminant function is positive for a certain observation $\mathbf{a}_i$, then that observation belongs to the positive class. Otherwise, the observation belongs to the negative class. In cases where no found pattern covers the observation $\mathbf{a}_i$, then the output of the discriminant is 0 and no decision can be made.

1.2.4 Alternative Implementations of LAD

The methodology for constructing a LAD decision model as discussed so far consists of the three mandatory steps: binarization, pattern generation, and theory formation. However, some alternative views to the adaptation of LAD to non-binary data have appeared in the literature. One such implementation is found in [Alexe S. & Hammer 2006] where the basic idea is to adapt the pattern generation-step to discrete data instead of transforming the training data to the binary form. The resulting methodology thus consists of the steps: Discretization, Pattern Generation, and Theory Formation.

The advantage of this alternative technique is its speed in comparison with the above methodology. Another advantage is that it bypasses the need for a binarization step as it is replaced by a simple discretization step that transforms numerical attributes to discrete attributes obtained by dividing the values each attribute can take into a number of segments, irrespective of whether the values occur in positive or negative observations. The number of segments is set by the user and is independent of the number of observations in the training set. The total number of new discrete attributes ($b_1, b_2, \ldots, b_n$) is the same as the number of non-transformed attributes, in contrast to the case of binarization.

[Alexe S & Hammer 2006] described an algorithm, which generates patterns from discrete attributes. Upon each iterative step of this algorithm all possible patterns of a certain degree d are studied and evaluated according to their prevalence property. A resulting pattern of degree d can be viewed as a set of d non-redundant intervals. At the end of the algorithm, a list of all possible patterns and their coverage is known. Whereas the patterns generated through this technique do not belong to any of the 4 special pattern types discussed at the beginning of this section, restrictions in terms of pattern homogeneity, prevalence, and degree can be set on a case by case basis when choosing the optimal patterns.

[Alexe G & Hammer 2006] developed an algorithm to generate spanned discrete patterns. In the binary domain, a pattern is qualified as spanned if for the same covered observations, it is composed of the maximum number of literals possible. The advantage of using spanned patterns is that they have stable performance in terms of prevalence and homogeneity when it comes to using the generated patterns on different new data sets. Although spanned patterns might cover fewer new observations, they result in a lower classification error. [Alexe G & Hammer 2006] proved experimentally that spanned patterns are robust and cause less classification errors. The algorithm executed in significantly less time for the same observation sets used in [Alexe S & Hammer 2006].

1.2.5 Multi-class LAD

Similarly to traditional linear classifiers and support vector machines, the LAD implementation described thus far separates observations into two classes only. In the CBM context, this translates to fault detection capability (i.e. detecting whether there is a fault in the system or not) or the identification of two different fault types. However, in many scenarios, it is desirable to have the ability to separate observations into more than two classes. One such scenario is multiple fault identification, which as defined earlier is the determination of the type of fault, its time or place. This knowledge is essential information in maintenance operations. The expansion of LAD to the multi-class case has been studied in the literature using two approaches.

The first approach for adapting LAD to a multi-class case does not require the alteration of the structure of LAD as explained thus far, as it requires applying the same classification algorithm to each pair of classes in the multiclass set. Each implementation of LAD results in a classification function $\hat{f}$ defined by a discriminant that separates the inputs into two distinct classes. The decision function $\hat{f}$ is therefore a dichotomy that maps observations from $\mathbb{R}^n$ to

$\{-1,1\}$ as in the case of two-class LAD. A multi-class classifier $\hat{F}$, on the other hand, must be able to map observations to a set of classes $\mathbb{R}^n \rightarrow \{c_1, c_2, \ldots, c_K\}$ where c_k represents one class k and K is the total number of classes. The number of dichotomies $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_Q$ that is capable of performing the function of a multi-class classifier is determined by the type of decomposition scheme that allows the transformation of the multi-class problem into a series of Q bipartitions solvable using two-class LAD decision models, or dichotomies. The decomposition matrix $\mathbf{D}$ is of the form $Q \times K$. The advantage of the decomposition is twofold: (1) it permits the use of the conventional LAD methodology on a multi-class problem and (2) it breaks down the classification process into easier to model two-class classification processes with improved accuracy.

After decomposing the classification problem and training the different dichotomies using LAD, a reconstruction scheme is necessary to consolidate the results of ever decision model. The reconstruction scheme resembles that of an MLP neural network with a unitary activation function.

Decomposition schemes must have two important properties: (1) validity, which states that each two classes must be distinguishable by at least one dichotomy, and (2) robustness, which requires a large hamming distance between rows and columns of the matrix in order to avoid misclassification and correlated errors respectively.

The simplest decomposition scheme is a series of dichotomies $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_K$, each separating one class from the remaining classes. Such a scheme is called one-per-class (OPC) and is represented by matrix:

$$\mathbf{D} = \begin{bmatrix} +1 & -1 & -1 & \cdots & -1 \\ -1 & +1 & -1 & \cdots & -1 \\ -1 & -1 & +1 & \cdots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & +1 \end{bmatrix}$$

This decomposition scheme requires the use of all the observations in obtaining the LAD classifier for each dichotomy. A number of decomposition schemes were tested in [Mayoraz &

Moreira 1996, Moreira & Mayoraz 1999, Moreira 2000]. In addition, [Moreira 2000] developed a new decomposition scheme whose matrix depends on the training data. The advantage of this scheme is that the dichotomies included in the decomposition are *pertinent*, which means that every dichotomy succeeds in classifying several classes at the same time. Testing results of LAD using different decomposition schemes showed that each scheme has different characteristics and tradeoffs involving accuracy, learning effort and model complexity.

The second approach, proposed in [Moreira 2000], for the adaptation of the LAD algorithm to the multiclass problem involves modifying the architecture of the pattern generation and theory formation steps in LAD. The aim here is to build a single multi-class classification model out of common patterns that are shared by all classes. The relation between the common patterns and the classes is described by a matrix, similar in structure to the decomposition matrix $\mathbf{D}$ described above. The basic requirement for the multi-class LAD decision model to succeed is that all the observations of every class must be covered by one or more patterns such that those same patterns do not cover any observations in another class. For example, a valid relationship matrix $\mathbf{D}$ resulting from the multi-class LAD procedure that creates a decision model separating three classes using seven patterns looks as follows:

$$\mathbf{D} = \begin{array}{c|ccc|} \boldsymbol{p_1} & 0.7 & 0 & 0.25 \\ \boldsymbol{p_2} & 0.3 & 0 & 0 \\ \boldsymbol{p_3} & 0 & 0.6 & 0 \\ \boldsymbol{p_4} & 1 & 0.4 & 0 \\ \boldsymbol{p_5} & 0 & 0.4 & 0.2 \\ \boldsymbol{p_6} & 0 & 0.2 & 0.8 \\ \boldsymbol{p_7} & 0 & 0 & 0.75 \end{array}$$

Where each row of the matrix represents a pattern $\boldsymbol{p_i}$ and the value for that row at each column (representing the class) is the percentage coverage of pattern $\boldsymbol{p_i}$ for the observations of that class. As the above example shows, class 1 in the first column has all its observations covered by patterns $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$. These same patterns do not cover any observations in class 2 as shown in column two. These patterns therefore separate class 1 from class 2. Pattern $\boldsymbol{p_1}$ is also used with pattern $\boldsymbol{p}_7$ to separate all the observations of class 3 from those of class 2. This is achieved by the seven patterns for all class combinations in the example above.

A decision model is trained by creating the class/pattern relationship matrix $\mathbf{D}$. The construction of patterns is done through an iterative procedure where a single pattern is generated in order to separate 2 distinct classes upon each run. The algorithm used in [Moreira 2000] for this special type of patterns is based on the $tabu$ heuristic technique. The algorithm is driven by $moves$, i.e. additions or omission of literals from a Boolean term. It consists of choosing an initial term that is considered a feasible solution and then searching among its neighbours to find the best neighbour to move to. The approach relies on a criterion called the differentiability between two classes, in order to organize the generation of patterns and control the amount of patterns to generate.

The advantage of this method over the previous one is that it generates a less complex general classification model that has better execution time than the previous scheme as proven experimentally in [Moreira 2000]. Another advantage is that, although it produced less accurate classification than the first approach, the patterns it generates are more intuitive as they better represent the complex relationships between more than two classes.

Chapter 2 THESIS ORGANIZATION

The research conducted in this thesis followed an evolutionary approach in order to achieve the objectives stated in the Introduction. The research therefore began by implementing a proven LAD methodology on an innovative industrial application: the detection of Rogue components within an inventory of reparable spare parts at a commercial airline company. As the research progressed, innovative modifications were introduced to the LAD methodology that allowed for its implementation on more complicated yet widely encountered industrial applications; namely the detection and identification of faults in power transformers using DGA and the detection of faults in rotor bearings using vibration signals. The diverse nature of the chosen applications, in terms of the type of data that is dealt with, addresses the first objective of the thesis: the study of LAD's applicability in different CBM applications. The innovations added to LAD throughout the research in order to adapt it to the requirements of the CBM applications and to improve the performance of LAD in CBM address the second and third objectives respectively. Figure 3.1 describes the innovations introduced to the LAD methodology in relation to the application it was applied on.

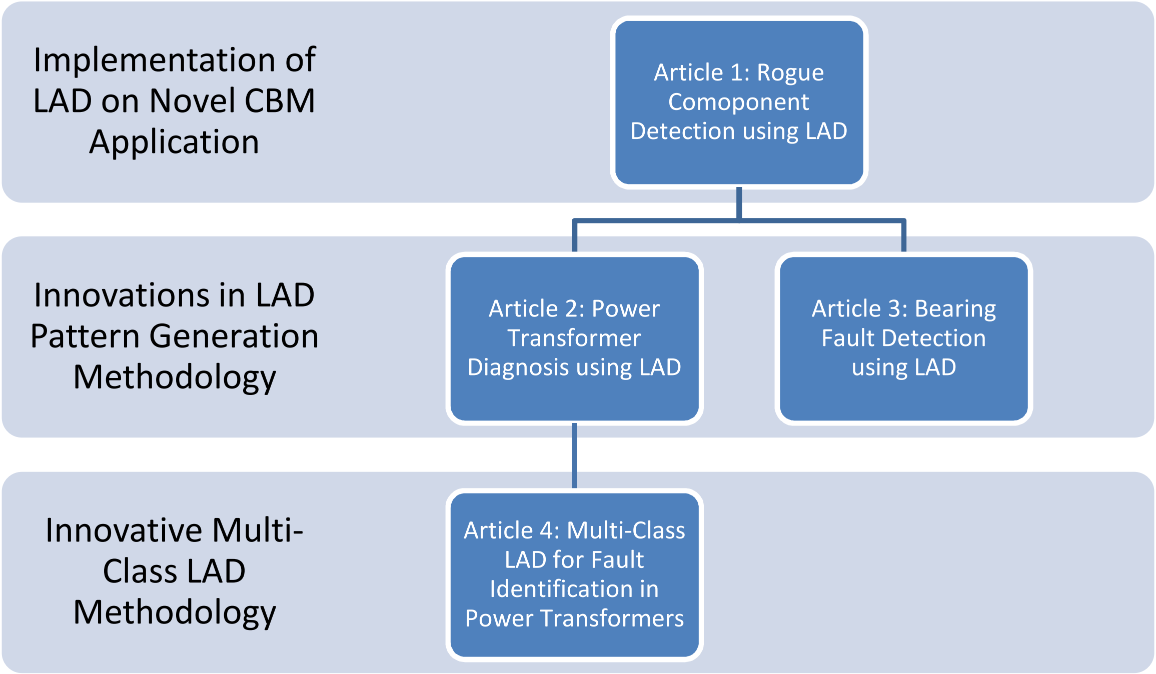
Implementation of LAD on Novel CBM Application
Article 1: Rogue Comoponent Detection using LAD
Innovations in LAD Pattern Generation Methodology
Article 2: Power Transformer Diagnosis using LAD
Article 3: Bearing Fault Detection using LAD
Innovative Multi-Class LAD Methodology
Article 4: Multi-Class LAD for Fault Identification in Power Transformers

Figure 2-1: Progress of LAD methodology throughout the articles and CBM applications

The research conducted was documented in four articles incorporated into this thesis. The first article has been approved and published in the “Journal of Intelligent Manufacturing”. The remaining articles have all been submitted within three months of writing this thesis and are pending decisions by the journals they were respectively submitted to. Following is a description of the articles and their connection with the stated objectives of this thesis.

As previously mentioned, the first article applies a proven LAD methodology on an innovative CBM application, the detection of rogue components in an inventory of reparable spare parts. The nature of the issue tackled presents a challenge as, prior to publishing the article, the automatic diagnosis of rogue components had never been addressed before. Although it is new, this issue displays the symptoms of the two problems that this thesis set out to solve: Improper implementation of PM actions and unknown rationale behind them. To clarify, one reason that the problem of rogue components arises is the fact that the PM actions set in place for the detection and repair of serviceable parts are never sufficient to prevent the components from becoming rogue. Another reason is the lack of profound knowledge by the maintenance teams about the PM procedures and their impact on the equipment. This is aggravated by the lack of knowledge transfer between the maintenance teams of the company using the equipment and the original equipment manufacturer (OEM). This prevents the company from accessing crucial information that may lead to the rationalization and potential updating of PM actions. The first step in solving this problem is the establishing of means to detect the rogue components plaguing the inventory. Maintenance teams relied on their expert knowledge to identify these rogue components and isolate them from the inventory to prevent operational disruptions. The role of LAD here is to automate the decision system using the information obtained through expert elicitation while at the same time providing feedback to the experts in the form of the generated patterns. The type of data used in for this application is historical in nature and requires a unique way of processing it to adapt it to the LAD methodology.

The second article has been submitted to the journal of “Discrete Applied Mathematics”. It addresses a well researched CBM application: the detection and isolation of faults in power transformers using DGA gas analysis data. Similarly to the previous application, the standard practice in the industry for fault diagnosis of power transformers is the use of standardized decision rules based on expert knowledge. The task set out in this article is to use LAD for the automation of the decision process while generating patterns from data processed in a similar manner to that used by maintenance experts. Similarly to the first application, the patterns generated through LAD provide some added insight to maintenance experts that could be used to improve the CBM program put in place for power transformers. In order to reply to the particular demand of this application to generate a large set of patterns to increase the interpretability of the automatic decision model, some innovative modifications were applied to a LAD methodology based on MILP. The modifications allow for the generation of multiple patterns for a single observation.

The third article addresses the application of LAD for the detection of faults in rotor bearings using vibration signals. It has been submitted to the journal of "Quality in Maintenance Eningeering". The particular challenge here is the processing of vibration data prior to its use in LAD for fault diagnosis. We rely on time and time-frequency based processing techniques to assess the signals in the database. The training data used to train the LAD decision model is therefore based on visual inspection and analysis of the processed signals. A new feature is introduced to the LAD methodology which allows for the generation of multiple patterns per single observation that are not subsets of each other. This modification adds value to the interpretability of the patterns. The implementation of LAD to this particular application highlights its potential in the vast field of diagnostics using vibration analysis.

The fourth article has been submitted to the journal "Discrete Applied Mathematics". It revisits the problem of fault diagnosis in power transformers using DGA. This article highlights an innovative multi-class LAD methodology that allows for the identification of faults in power transformers using a single implementation of the LAD decision model. The presented methodology achieves higher accuracy levels than the ones proposed in the literature. In addition, the generated patterns are more intuitive in nature and give maintenance experts helpful insight.

Chapters 3 to 6 of this thesis present the four articles that highlight the novelty of the research conducted. Chapter 7 discusses the findings of the four articles and assesses their effectiveness in resolving the problems discussed in the Introduction. This thesis ends with concluding remarks that discuss the prospects of future research based on the results achieved here.

Chapter 3 Rogue Components - Their Effect and Control using Logical Analysis of Data

Mohamad-Ali Mortada, Thomas Carroll III, Soumaya Yacout, Aouni Lakis

Published in: JOURNAL OF INTELLIGENT MANUFACTURING

DOI: 10.1007/s10845-009-0351-1

3.1 Abstract

There is a small subset of any repairable component population that can develop a failure mode outside the scope of the standard repair and overhaul procedures, which makes them "rogue". When this happens, a Darwinian-like "natural selection" phenomenon ensures that they will be placed in the most disadvantageous position in the asset management program, negatively affecting multiple aspects of the operational and maintenance organizations. Rogue components have long plagued the airline industry and created havoc in their asset management programs. In this paper, we describe how these rogues develop, outline the natural selection process that leads to their hampering the asset management program, and examine some of the negative impacts that ensue. Then we propose a Condition Based Maintenance approach to control the development of these components. We explore the use of a supervised learning data mining technique called Logical Analysis of Data (LAD) in CBM for the purpose of detecting rogues within a population of repairable components. We apply the resulting LAD based decision model on an inventory of turbo compressors belonging to an airline fleet. Finally, we evaluate the applicability of LAD to the rogue component detection problem and review its efficiency as a decision model for this type of problem.

3.2 Introduction

Aircraft maintenance and reliability programs are essential for the safety and airworthiness of airplanes. Maintenance, Repair and Overhaul (MRO) operations take up a large portion of aviation companies' spending. *Aerostrategy* estimated total air transport MRO costs at $40.8 billion in 2006 [Flint 2007]. According to OAG (Official Airline Guide)**,** global MRO spending on military aviation will witness a 14.9% increase over the next decade to reach $67.3 billion a year in 2018 [OAG 2008]. The performance of an aircraft operator often hinges on its capability to provide fast and efficient replacement of defective components in its fleet. For that reason, operators may either carry in-house maintaining capability or subcontract component availability [Kilpi & Vepsalainen 2004]. In both cases the operator's inventory must be composed of ready-to-replace components.

As mandated by many civil aviation authorities, maintenance programs rely on condition monitoring (CM) to track the performance of the different parts and components of an aircraft.

An aircraft operator`s asset management program usually handles many repairable components of the same type. Such components could be in one of three places within the system:

1- In service on one of the aircraft
2- Undergoing repairs in the maintenance shop
3- In the spare part inventory

Each component has a unique serial number S/N_i. Most components have exhibited many installation and removal instances throughout their lifetime within the population. These instances are noted and logged in records (removal records) which are kept for every single component.

The integrity of components installed as replacements to failing parts is essential to the viability of the operator's asset management program. Typically, a repairable component works as expected for its designed lifecycle or between scheduled events [Carroll 2008]. In some cases, a component fails to fulfill these expectations for 3 possible reasons:

1- The component has a manufacturing flaw which can be detected as it exhibits a failure or a series of failures in its early service life.
2- The component is aging and is thus suffering from consecutive failures towards the end of its service life.
3- The component is classified as rogue. Rogue components are repairable parts that develop a failure mode outside the scope of the standard repair and overhaul procedures.

Each of these reasons has identifiable characteristics. Rogue components, however are extremely difficult to identify and can spread throughout the component population.

If an asset management program includes repaired, reconditioned or overhauled parts, there is an ever-present risk of "rogue" components developing in the population. When this happens, there is a compounding negative effect across these aspects of operational and maintenance organizations:

- Operational Reliability
- Asset Management Programs
- Maintenance Effectiveness

- Preventive Maintenance Programs
- Maintenance Support & Training Programs
- Component Repair Facility
- Components themselves
- Mechanical System Hardware
- Operator/OEM Engineering

The main problem occurs when rogue components slip into the asset management system through the operator’s spare parts inventory. The detection of such components is important to ensure the reliability of the system.

A study was performed in 1995 to calculate the financial impact to an airline when a rogue component develops. It was determined that on the average, a single rogue component will cost $50,000 (US) over its life. This number pertains only to the maintenance burden and does not include flight delays and / or cancellations or flight restrictions because of the perpetuated system problems when installed to correct an aircraft system problem. Additionally, due to the high usage of spares as a result of multiple installations to resolve the perpetuated system problem, the asset management program may procure additional spare inventory, resulting in abnormally high inventory levels and an increased manpower hours. In an effort to resolve the unconfirmed failure in the shop, the OEM may elect to modify the component. Most times, these modifications are ineffective, and the airline bears the brunt of the cost. This is an extremely costly failure mode and impacts the effectiveness of many different aspects within the airline maintenance organization.

Condition Based Maintenance (CBM) is defined as perpetual monitoring of a system`s health such that maintenance is performed when an intervention is deemed necessary. Rogue component detection is an example of how CBM can be used in aviation to detect faults. Today commercial airline operators have adopted several aspects of CBM; and with the advancement of technology, they will be able to adopt its full benefits [Teal & Sorensen 2001].

Current Practice in Rogue Component Detection

Current condition monitoring methods rely generally on statistical analysis tools or different combinations of parametric and non-parametric tools in order to evaluate the performance of aircraft components. The main drawback to the use of statistical tools is the precondition that the collected failure data are homogeneous and independent and identically-distributed (i.i.d). Many statistical analysis methods assume that the data belong to a certain probability distribution; such assumptions are not always true. An example is presented in [Leung et al. 2007] where a hybrid parametric and statistical technique is used to classify aircraft components according to their maintenance status using their removal records as input data. The classification decision is done manually based on a visual evaluation of the output charts. In [Carroll 2008], a set of indicators where proposed that would identify whether such a component is rogue or not by assessing its installation and removal history.

This paper is organized as follows: First we define rogue components and explain how they develop by outlining the “natural selection” phenomenon. Next we examine some of the negative impacts caused by rogue components by recounting 2 possible real life scenarios. Then we propose the use of Logical Analysis of Data (LAD) as a decision model for detecting rogue components within a population and describe the indicators involved in rogue component detection. We explain LAD methodology and explain its implementation in rogue component detection. Finally, we test the LAD technique on data obtained from the industry and study the results.

3.3 Rogue Component Definition

A rogue component is defined as an individual repairable component, which repeatedly experiences consecutive short in-service periods, manifests the same mechanical system fault each time it is installed, and when it is removed from service, the mechanical system fault is corrected.

The reason a component develops a rogue failure is because its repair and / or overhaul tests do not *address* 100% of the component’s operating functions, characteristics or environment. Interviews with various Original Equipment Manufacturers (OEM) revealed the test coverage is typically about 85% of the component’s complete functionality. Even if all the functions were covered, the operating environment of the component when it is installed in the mechanical system is usually quite different than the repair facility, so if a failure is dependent upon a particular in-service environmental condition, it is unlikely that it will be duplicated during test.

Additionally, the repair and / or overhaul tests are developed to identify *anticipated* failures, focused on testing things that are expected to fail. For example, it would not make sense to check all the screws or electrical ground straps each time a component comes into the shop, since the chance of failure for those pieces is practically zero and the cost of performing such extensive testing during each shop visit would be exorbitant.

When a component experiences a failure that was either *unaddressed* or *unanticipated* by the testing procedures, a rogue is born. Since every test that is performed misses that specific aspect of the component's functionality, the fault will never be identified and resolved [Leung et al. 2007].

The rogue failure cannot be predicted if, when, and where it will occur. It is a random failure that develops and will remain until definitive action is taken to resolve it. Not every part number population will develop rogue failures. Also, when a rogue failure occurs, not all the individual components within that part number population will necessarily develop that failure. However, any part number population has the potential for individuals to develop rogue failures, regardless of how simple or complex the design and functionality.

Natural Selection Phenomenon

There is a Darwinian-like "natural selection" process that ensures the rogue components will be positioned in the most disadvantageous places in the asset management program. The following depiction demonstrates the mechanics of this "natural selection" phenomenon.

Initially, it starts with a spare inventory and in-service population that are comprised of serviceable (Good) components that function as expected. As a part fails in service, it is removed and replaced with a good part from the spare pool in order to solve the mechanical system problem. The component repair facility tests and duplicates the problem with the failed unit, repairs and returns it to the spare pool.

The "natural selection" process begins when a rogue failure develops in one of the in-service components. When this occurs, the component is removed and sent to the repair facility. It typically tests normally, as "No Fault Found" (NFF), and returns to the spare pool with no corrective action taken to resolve that failure.

As long as there are no failures in the in-service population, the rogue component will remain in the spare pool. If the unique rogue failure mode is not recognized and resolved, then other components may develop the same condition.

Every new rogue component is removed from service and sent to the repair facility where it tests as NFF, and is returned to the spare pool. As such, the potential negative effect of the rogues is multiplied.

Though these rogue components make up a very small part of the general population, the "natural selection" process ensures that they are sorted out to the most critical place in the asset management process – the spare inventory. According to accounts from experts in the industry, there are documented cases where the entire spare pool had been comprised of rogues.

3.4 The Effect of Rogue Components

When rogue components develop within a part number group, there are significant detrimental impacts to various aspects of the operational and maintenance organizations. These impacts will below.

3.4.1 Maintenance Effectiveness

Mechanical system problem resolution relies on the spare inventory being comprised of serviceable components. When a component is installed from the spare inventory and the system problem continues, it is illogical to assume that the replacement was a defective part. When a rogue component is installed, it severely compromises maintenance effectiveness. The following scenario describes an actual case:

Case Study

There is a system that maintains a constant air pressure by adjusting the opening of a vent valve to react to operational and environmental changes. This system is comprised of an electronic control unit, various sensing units, and a vent valve.

A system malfunction occurred that caused the vent valve to intermittently lock up in mid-position during high operational demands. The maintenance technicians could not duplicate the fault, so they replaced the control unit as the most likely component to cause this problem.

The problem repeated. Since the control unit did not resolve the problem, the vent valve was replaced, which required considerable system down time and maintenance resources. Now when the system operated during high demand periods, the valve intermittently oscillated open and closed, when it should remain in a fixed position. This problem could not be duplicated by maintenance.

Since this new issue surfaced immediately after the installation of the valve, it was replaced again in the assumption that it was defective from stock. The system was down again for a considerable amount of time during this second replacement. However, the oscillation problem continued.

All the wiring was checked leading to the valve, and after a number of additional repeat complaints, all the valve electrical connectors and sensors were replaced, with no result. The control unit was replaced again and the oscillation problem was resolved.

Root Cause Analysis

The root cause of the initial system malfunction (when the valve would stop during operation) was a faulty vent valve. The control unit first installed was a rogue component, which had an existing failure that would cause the valve to intermittently oscillate during high operational demands.

However, this rogue failure could not manifest itself until a serviceable vent valve was installed, since the original defective valve would lock up during operation, thus preventing the oscillation from occurring.

This type of compound problem is not common. Usually the introduction of a rogue component causes the original system problem to continue, which results in the replacement of the associated system components, extensive system troubleshooting and repeat replacement of the rogue component until a “good” spare is installed.

3.4.2 Mechanical support

When a chronic system problem that is caused by the introduction of a rogue component persists after all the components have been replaced, the next logical step is to troubleshoot the interconnecting wiring or plumbing.

It is very likely that much of this hardware is located in areas that are very difficult or time consuming to access, possibly requiring special tooling or OEM expertise to disassemble and reassemble. In some cases, OEM engineering drawings or wiring schematics are also needed in order to proceed with the next phase of troubleshooting, which can take a considerable amount of time and / or expense to acquire.

Since the root cause of the continuing problem is actually rogue component, this in-depth troubleshooting and extensive maintenance support will not resolve the system malfunction.

3.4.3 Operational Reliability

When the maintenance effectiveness is compromised by the presence of rogue components, the mechanical system operational reliability naturally suffers. There are repeat events of system failures and associated down time, along with extended periods of in-depth troubleshooting.

Case Study

An Auxiliary Power Unit (APU) provides electrical and pneumatic power, comprised of a turbine and a generator, with a main electronic control, and a number of external sensors. One of these sensors is located in an actuator that opens a door to allow air to enter the APU during operation. It is a switch that provides a signal to the electronic control unit that the door is open, so the APU can be started and allowed to run.

If the door should start to close at any time, the switch will immediately signal the electronic control unit to shut the APU down to prevent catastrophic damage. The electronic control unit also has a monitoring circuit to record which stage of the start or run cycle had failed, providing direction for system troubleshooting.

In this case, a door actuator had failed, which caused the APU to shut down when it was running. It was replaced with a rogue door actuator. Now the APU would intermittently shut down during various stages of the start cycle. This problem could not be duplicated during system troubleshooting, so maintenance reacted to the fault codes recorded by the electronic control unit.

Since there were different fault codes each time, a considerable amount of various components were replaced and the interconnecting wiring was checked a number of times. New wires were strung between the electronic control unit and the APU.

When another door actuator was installed, the problem stopped. This recurring problem generated 45 complaints that spanned a period of 344 days, with a total of 46 days of complete system shut down.

Root Cause Analysis

The first door actuator that was installed was a rogue component that had an intermittent failure of the sensor, which would indicate the APU door was closing when it was open.

Because this malfunction intermittently happened during different stages of the APU start cycle, the electronic control unit's fault recording system would record the each stage of the start sequence that was interrupted and list the most likely device that could be responsible for causing that failure at that particular time in the start cycle.

Unfortunately, the monitoring system would not record that the door actuator switch had signaled the door was closing during the start cycle.

3.4.4 Asset Management

When a significant portion of the spare inventory is comprised of rogue components, traditional asset management models are no longer effective.

Typically, multiple spares must be withdrawn to resolve in-service problems, resulting in sporadically high spare usage and low spare levels. If the available spare inventory repeatedly reaches critically low levels, then more spares will be added. As more rogue components develop, this process will repeat until there is an abnormally high number of spares, which cannot be managed effectively.

Case Study

An operator had a fleet of 40 aircraft, each having an autopilot system comprised of a control panel, pitch computer, roll computer, and a number of servomotors and sensors. The asset management program determined that 6 pitch computers were needed for the spare inventory to maintain a satisfactory level of support.

After a number of years, it was difficult to keep the spare levels up, so more computers were procured. It was assumed that the equipment was getting older, so the increased usage of the spares was a natural progression. This chain of events repeated as the years went on until there were 28 spare computers to support the 40 that were in service

Root Cause Analysis

Initially, the pitch computer population developed a small number of rogue components, which was a substantial percentage of the spare population. The result was a recurring low spare level, so more computers were procured to offset the demand. As new computers were added to the spare inventory, the percentage of rogue to non-rogue spares was reduced, so it was possible to maintain a satisfactory spare level, despite the rogue component presence.

Over time more rogue components developed, again increasing the rogue to non-rogue percentage in the spare inventory, with the same reaction from the asset management program, which diluted the rogue component impact to the asset management program. This cycle continued with the incremental increases to the spare inventory until extremely high levels were obtained.

After an analysis of the pitch computer population's in-service performance, it was discovered that 20 of the 28 spare computers were rogue components. Once these were identified and resolved, it was possible to surplus 20 of the spares. Each computer was valued at approximately $12,000 (US), so the cost of acquiring the excess inventory totaled around $240,000 (US). When the excess components were sold on the surplus market, only a small fraction of the initial expense was recovered.

3.4.5 Preventive Maintenance Programs

Some major components receive regularly scheduled preventive maintenance to ensure they operate through their designed life cycle, such as oil and filter changes. If rogue failures develop in these components, then an increasing number of in-service failures will occur despite these preventive maintenance actions. In an effort to eliminate these failures, typically the interval between preventive maintenance actions will be reduced from what was originally set. This is a very expensive action to take, as it could double or triple the recurring maintenance burden and cost. If the rogue failure mode is not corrected, then the failures will still continue despite the additional preventive maintenance.

Case Study

On a turbine engine, the Constant Speed Drive (CSD) gearbox drives the electrical power generator at a constant RPM, regardless of the engine RPM. This gearbox has a preventive maintenance program in place to replace the oil and filter every 1000 operating hours. After several years of operating these engines, several CSDs exhibited failure mode that resulted in oil starvation and catastrophic failure.

The immediate plan to resolve this situation was to change the oil and filter every 500 hours, instead of the original 1000 hours.

With a CSD population of 240 units that had the filter and oil replaced about 3 times a year at a cost of $150 and 2 man-hours labor, the total annual maintenance burden was approximately $108,000 (US) and 1440 man-hours per year. Reducing the preventive maintenance interval to 500 hours would double the cost and man-hour consumption.

Root Cause Analysis:

Of the total CSD population, only 10 had exhibited this fault, but had done so repeatedly. These individuals developed a rogue failure that caused the oil pressure to fluctuate and damage an oil pressure relief valve, which then starved the CSD of oil.

An analysis of the rogue components revealed the unusual failure, which was resolved. The oil and filter interval remained the same and the reliability returned to the previous level.

3.4.6 Maintenance Training Programs

If mechanical system problems become chronic, it appears that maintenance efforts are ineffective, so the formal maintenance training programs are typically reassessed in an effort to raise the technical expertise.

When no formal technical training exists for those troublesome mechanical systems, then courses may be created to improve the overall understanding of system description, operation, troubleshooting and repair. If a formal technical training program exists, then the course material must be lacking, so a great deal of time and effort is spent to amplify the various aspects of the training to provide more detail.

When the maintenance effectiveness still does not improve, then the technical personnel may be required to attend recurrent training, assuming that repeated exposure to the same information will improve their expertise.

In all these situations, the expanded / additional / recurrent training will typically have little positive effect, since the root cause of the issue is not system knowledge, but rogue components. In addition, the maintenance personnel generally have a good understanding of the systems they work with, so subjecting them to additional training can convey the impression that management believes they are technically deficient, rather than taking action to identify and resolve the root cause of the problem. This can create or compound a division between management and the technical workforce.

3.4.7 Component Repair Facility

Rogue components cause a sporadic rate of removals, so the component repair facility has a correspondingly sporadic workload. Typically, there are periods of relative inactivity that are punctuated by high demands, can exceed the repair facility's manpower and testing capability.

The resulting low spare inventory levels, high repair backlog and extended lead times can force a selective type of testing that centers on the components that require the least amount of work, as satisfying the demand for serviceable spares outweighs the need to perform the necessary in-depth analysis of rogue failures. This tactical approach perpetuates the existing rogue component population and allows more to develop, which amplifies the demands and difficulties for the repair facility.

Case Study

A certain component required 3 elapsed hours to perform a serviceability test, 12-14 hours to calibrate, and 20-24 hours to overhaul. Any failure that was above and beyond the typical overhaul could take upwards of 40 hours to repair.

Because of a rogue component presence, the unserviceable components arrived in batches, which severely taxed the two test stations in the repair facility and created a significant backlog. Additionally, the resulting low spare inventory levels pressured the repair facility to produce serviceable components quickly in order to support the needs of the operation. If one or more of the components required overhaul or extensive repair, then the remaining backlog was audited to determine which ones could be turned around, that is, tested with no adjustment, repair or overhaul required.

Once several components were returned to the spare inventory, the production pressure lessened and the more time-consuming repairs or overhauls could resume.

The unserviceable components that could be turned around were the ones that had been replaced as a result of poor troubleshooting and the rogue components, as they both tested normally. Since there was no in-depth analysis of the rogue failures, the rogue component population grew, increasing the volume of the sporadic returns and intensifying the pressures on the repair facility to produce serviceable spares more quickly.

The turn-around methodology became a standard operating procedure, which became a self-feeding rogue component problem.

3.4.8 Operator / OEM Engineering

When a significant rogue population develops, the number of system complaints grows and the repair facility has a high rate of No Faults Found (NFF). As the operational reliability continues to decrease despite all the maintenance technical expertise improvements, then the operator or OEM engineering may be tasked with identifying the root causes.

Since a definite problem cannot be identified, then the efforts turn to theorizing what could be a root cause and component or system design modifications may be developed in an attempt to resolve the assumed shortcomings. Generally, the reliability improvement modifications do not address the true root cause, which is the rogue failure.

The poor operational reliability continues, with the risk that the incorporated change can also negatively impact the reliability of the general population.

Rogue components can present another challenge when a modification is introduced to enhance the operation of in-service components, such as a functionality or performance change. When these upgrade modifications are started, the spare inventory is modified as “seed” units, and then placed into service to remove the next wave of components to be modified. This process continues until all the modifications have been accomplished.

If the spare inventory contains a significant amount of rogue components, it will critically impact the modification campaign. The “natural selection” phenomenon ensures all the rogue components are in the spare inventory. When these components are modified and placed in service at the same time, they create their natural system failures. Since multiple system faults appear coincidentally as the modification was introduced, it is logical to assume the modification was the root cause of this sudden spike in operational problems.

The engineering group will typically halt the modification, so they can analyze each aspect of the modification, looking for something that was introduced that could cause such an adverse reaction. However, since the analysis does not focus on the rogue failure, it will consume a tremendous amount of manpower and resources for nothing. In some cases, a completely new modification will be developed – with the same results.

Case Study

The heart of an engine indicating system is a computer that processes all the various inputs and displays the operational parameters on a monitor. This computer has an internal testing system that continually checks its functionality. If it detects an internal or external anomaly, it will display a fault message. In this case, there were 46 of these computers in service, with excellent overall operational reliability.

A modification to the computer software was introduced that changed the display characteristics. As the first batch of modified units was placed into service, a high number of system failures immediately occurred. When the modified computers were removed from service and tested, there were no faults found.

It was assumed that something in the new software must be the cause of these anomalies. A great deal of engineering time was spent reviewing all the software changes, but nothing could be identified as a root cause. The modification was halted.

Root Cause Analysis

The spare inventory had a significant number of rogue components (approximately 75%). When all the modified rogue components were placed into service at the same time, there was an abnormal spike in the amount of system faults, and when the modified computers were removed, the system reliability returned to normal.

The engineering group could find no problems with the modification, so an analysis of the in-service performance of all the modified components was initiated. It revealed that approximately 25% of them did not exhibit any problems when placed into service.

If the software modification was the root cause, then all the modified components should have exhibited faults. Since a segment of the modified components had no faults, the modification was exonerated. However, a considerable amount of engineering time and resources were expended needlessly analyzing the modification.

3.4.9 Mechanical System Hardware

For the most part, rogue components create intermittent system faults. When a system problem persists after all the components have been replaced, the next logical step is to suspect an intermittent malfunction of the interconnecting wiring and connectors that could be caused by dynamic operational conditions, such as vibration, flexing, heat, cold, water ingression, etc.

Generally, the maintenance technician will attempt to replicate these conditions by subjecting it to physical stress and environmental conditions that could immediately create a new problem or weaken the wiring or connectors so another intermittent problem will develop in the future.

Typically, an ohmmeter is used to check the continuity of the wiring, which is measured with two metal probes. In order to accomplish the checks, one probe might be inserted into the female pins of the electrical connectors, which can consist of a high number of very small gauge pins. If the probe is not the exact size of the male counterpart, when it is inserted it can spread the internal contact points of the female pin, which will create an intermittent connection when the connectors are rejoined. When this occurs, the troubleshooting of this induced fault is extremely difficult to locate and resolve.

Another method of identifying a wiring problem that is intermittently shorting to ground is to use test equipment known as a "megger", which uses a high voltage to determine if the wiring insulation is breaking down. If it is not used correctly, it could damage the insulation. Additionally, if all the interconnected electrical components are not disconnected, it will damage their internal workings, creating additional system faults.

Components Themselves

As an inordinate amount of components are replaced to resolve a single system problem caused by a rogue component, damage can occur during the removal, installation, and shipping of the components to and from the repair facility. Additionally, damage can occur during installation from electrical or pressure surges during the connection / disconnection of the components, which could create another intermittent fault. All of these scenarios are very expensive and time consuming to resolve.

3.5 Control of Rogue Components

Rogue components cannot be prevented. It is impossible to proactively anticipate a failure that could occur and develop a new test to identify it before it happens. Therefore, the only action that can be taken is reactive, which is to detect and isolate rogue component from the population they're embedded in. Once detected and isolated, their unique failure modes can be analyzed in order to develop tests to identify them in the future.

The first step in the detection of rogue components is to develop a data collection system that captures system maintenance events and tracks the installed / removed components by part and unique serial number.

By monitoring certain indicators in the data collection system, patterns that are unique to rogue components can be discovered. [Carroll 2008] reported the following patterns that are unique to rogue components after years of manually monitoring repairable component removal records:

1) Repeated short in-service installation periods. Shortness of the period is determined by comparison to a typical service life time of a component. A third consecutive short in-service time triggers the rogue flag.
2) Repeated identical reasons for removal. If the component exhibits identical system fault manifestation for the last 3 removals, then the rogue flag is triggered.
3) Shop records indicate that the failure cannot be detected by standard testing procedures: No Failure Found.
4) Removal of the failing component from the operating system resolves the system fault. If the system is still at fault even when the component is removed, then this means the rogue condition is not satisfied.

If these patterns or occurrences are found in a certain component's removal record, then that component can be classified as rogue. It takes the presence of all the above criteria to be able to classify a component as rogue.

Current practice in the identification of rogue components involves searching through thousands of removal records manually and detecting visually the above mentioned patterns in order to extract these outlier components. The automation of this process through an automatic decision model that classifies repairable components into two classes: (1) Rogue and (2) Non-Rogue, provides a better solution to this problem. LAD, as a decision model that is capable of automatically generating patterns from input data, is an ideal method to automate the above process.

In what follows is a description of the LAD methodology and its implementation in rogue component detection.

3.5.1 LAD Methodology

LAD is a data mining technique that classifies observations into the categories they are associated to. The history of this method goes back to 1988 where it was first proposed in [Cama et al.

1998] as a method for classifying binary data. LAD has been proven to give comparable and even superior results in some cases to the traditional decision models used in CBM, such as neural networks and support vector machines [Salamanca 2008, Boros & Hammer 2000]. The main advantages of LAD are:

1- It is not based on statistical analysis. Consequently, it does not assume that the data belong to a specific statistical distribution. The method therefore does not require statistical analysis of data prior to its use.
2- LAD automatically extracts features and generates patterns from the indicators collected from the observations and, accordingly, sorts the components into separate classes based on the patterns generated.
3- Unlike other data analysis techniques, such as neural networks and support vector machines, LAD is a transparent method; the output of LAD can be traced back to the specific root causes that resulted in the categorization of a specific observation into a certain class. This explanatory power, a potential asset to maintenance experts, is attributed to the patterns that LAD can generate from the observation and analysis of criteria that are pertinent to the classification problem.

As LAD is a supervised learning technique, it relies on the presence of training data, already sorted into the existing classes, in order to generate the patterns. Training data are a learning set of pre-classified observations based on which the algorithm develops its decision function. In the case of rogue component detection, these observations are the records of installation and removal (removal records) of some components in the population whose *rogueness* or *nonrogeness* is already confirmed. A typical training set is composed of two subsets: a positive observation subset composed of rogue observations and a negative observation subset composed of non-rogue observations.

After the acquisition of training data, the LAD algorithm can be divided into 3 steps:

1- Data Binarization,
2- Pattern Generation
3- Theory Formation.

Data Binarization

The information extracted from the training observations is binarized prior to analysis. Each observation can be considered as a vector of *m* indicators. As LAD is based on discrete mathematics and combinatorial enumeration, its input, the observation vectors formed by the non-binary indicators, are transformed into Boolean observation vectors of *n* binary attributes.

The binarization of non-binary indicators depends on their type. Indicators can be divided to 2 categories: descriptive indicators (e.g. code type) and numerical indicators (e.g. time, temperature, etc...). Descriptive indicators can take up many possible values. Binarization, in this case, occurs by allocating to each value v_n of the indicator x a Boolean variable $b(x, v_n)$such that [Boros & Hammer 2000]:

$$b(x, v_n) = \begin{cases} 1 & if \quad x = v_n \\ 0 & otherwise \end{cases} \tag{1}$$

Numerical indicators are binarized using two types of binary variables: Level variables and Interval variables. Level binary variables are obtained by first sorting the values of the numerical indicator in the observation set in descending order and then introducing 1 cut-point between each interval $v_n < v_{n-1}$ such that $v_n \in S^+$ and $v_{n-1} \in S^-$ or vice versa, where S^+ and S^- represent the positive and negative observation subsets respectively and the cut-point t is calculated as [Boros & Hammer 2000]:

$$t = 0.5(v_n + v_{n-1}) \tag{2}$$

The resulting binary attributes are Boolean variables defined by each cut-point t such that [Boros & Hammer 2000]:

$$b(x, t) = \begin{cases} 1 & if \quad x \geq t \\ 0 & if \quad x < t \end{cases} \tag{3}$$

Interval binary variables, as the name implies, take the value of 1 when the value of the numerical indicator is within a certain interval and 0 otherwise. These intervals are formed by the cut-points calculated for the level variables. An interval binary variable of a numerical indicator is therefore obtained from every two cut-points found for that indicator while calculating the level variables, and would have the following form [Boros & Hammer 2000]:

$$b(x,t_1,t_2)=\begin{cases}1 & \mathit{if} \quad t_1 \le x \le t_2 \\ 0 & \mathit{otherwise}\end{cases} \tag{4}$$

The cut-points t_1 and t_2 belong to the level binary attributes obtained for the numerical indicator.

The outcome of the binarization of an observation set is a set of Boolean observation vectors with a number of attributes n exceeding the initial number of non-binary indicators m (m>n). For a total of R observations, we thus obtain R Boolean observation vectors $O_1, O_2, ..., O_R$ of dimension n. A Boolean observation vector has the form $O_r = y_1 y_2 y_3 ... y_n$ where y_i is a binary digit.

Pattern Generation

After transforming the observation set into Boolean observation vectors of dimension *n*, a bottom-up pattern generation approach is implemented to generate the patterns. This approach starts by finding a binary variable that *covers* one or more observations. Such a variable is called a *literal* in algebraic terms. A literal that covers an observation O_r has the form b_i if the value of y_i in O_r is 1 and the form $\overline{b}_i$ if the value of y_i is 0. A combination of literals is referred to as a *term*. A term is said to cover a certain observation when all the literals of that term cover the Boolean observation vector. For example, the term $\overline{b}_2 b_3 b_4$ covers the observation $O_r = 1011101$ since the value of that binary observation vector at the digits $y_2 y_3 y_4$ is 011. Similarly, the term $\overline{b}_1 b_2 \overline{b}_3 b_4$ covers an observation $O_r = 0101011$. A term is said to be of degree k if it is composed of k literals. For example, the terms in the examples above are of degree 3 and 4 respectively.

If a literal covers both positive and negative observations, then it is considered a *candidate*. More literals are added to it progressively, each time checking whether it still covers observations. If by adding more literals, the number of observations covered becomes zero, then that particular term is discarded. Otherwise, the term keeps its candidate status as long as it covers at least one positive observation and one negative observation or vice versa. If, by adding another literal, the resulting term covers only positive (negative) observations, then it is considered a positive (negative) prime pattern. This methodology favors the generation of small patterns, thus following the simplicity principle [Boros & Hammer 2000]. In order to reduce the amount of computations necessary, the lexicographic order is followed in generating the patterns [Boros & Hammer 2000]:

$$b_1 < \overline{b_1} < b_2 < \overline{b_2} < ... \tag{5}$$

The number of terms to be searched for patterns increases exponentially with the number of binary attributes that constitute a Boolean observation vector. For n attributes, the total number of terms is given as:

$$\sum_{i=1}^{n} 2^i \cdot \binom{n}{i} \tag{6}$$

For example, for a number of attributes $n = 45$ which is typical of a problem of this nature, the total number of terms to be searched is 2.95431×10^{21}. Therefore, due to computational and time constraints, a limit is set on the maximum degree of terms to be searched for patterns.

Theory Formation

The generated positive patterns which cover the positive observations are denoted by $P_1, P_2, ..., P_k, ..., P_K$, whereas the negative patterns are denoted by $N_1, N_2, ..., N_l, ..., N_L$. These positive and negative patterns are used to produce a discriminant function which, in the context of this paper, can separate rogue components from non-rogue ones. This function is of the form [Boros & Hammer 2000]:

$$\Delta(O_r) = \sum_{k=1}^{K} w_k^+ P_k(O_r) - \sum_{l=1}^{L} w_l^- N_l(O_r) \tag{7}$$

Where the value $P_k(O_r)$ is one if the positive pattern P_k covers observation O_r and zero otherwise. Similarly, the value $N_l(O_r)$ is one if the negative pattern N_l covers observation O_r and zero otherwise. The resulting discriminant Δ is thus the weighted sum of the values of all the generated positive and negative patterns for a certain observation O_r. The weights w_k^+ and w_l^-

can be calculated in multiple ways. The method used here is to compute the weight of a pattern as a normalized function of the number of observations it covers [Salamanca 2008]:

$$w_k^+ = \frac{\sum_{r=1}^{R} P_k(O_r)}{\sum_{k=1}^{K}\sum_{r=1}^{R} P_k(O_r)} \tag{8}$$

The negative weights are calculated similarly.

The output of the discriminant function shown in (7) is therefore a value between -1 and +1. If the value of Δ is closer to $+1$, then the observation is classified as positive (Rogue). If the value of Δ is closer to -1, then the observation is classified as negative (Non-Rogue). A value close to 0 indicates that the results are inconclusive, therefore no classification takes place. A threshold $\pm\tau$, set by the user, is the smallest value beyond which the observation is regarded as unclassified. The LAD algorithm decision function can therefore be formulated as:

$$f = \begin{cases} Positive & if \quad \Delta \geq +\tau \\ Negative & if \quad \Delta \leq -\tau \\ Unclassified & if \quad -\tau < \Delta < +\tau \end{cases} \tag{9}$$

The above decision function can be used to test for the rogueness of any new observation:

The necessary indicators are extracted from its removal records and binarized. Then, the resulting Boolean attributes are plugged into the discriminant function to get Δ. The decision function then reveals to what class that specific component belongs to.

3.6 Implementation

In Condition Based Maintenance, the detection of a fault can only be achieved if there exists a set of indicators that can reveal information about the status of the asset by monitoring them.

LAD, as a supervised learning decision model, has only recently been adopted in CBM in [Salamanca 2008]. Implementing LAD for the purpose of detecting rogue components requires the preparation of training data in the form of observation vectors before binarization can occur. As explained in the previous section, these observation vectors are formed by the indicators used to monitor the component's status in CBM. The binarization step then transforms these observation vectors to Boolean observation vectors.

In the case of repairable components of an aircraft fleet, the indicators that form the observation vectors are extracted from the indicators in the removal records of these components. Judging from the criteria that characterize rogue components, the following indicators found in the components' removal records can be extracted and used to form the LAD observation vectors:

1- Fault Confirmation Codes (FCC): When a component is removed, it is taken to shop for check-up and repair. After each repair, a "Fault Confirmation Code" is added to the component's record. There are 9 possible removal confirmation codes: F_1, F_2... and F_9. As shown in Figure 1, these codes describe what kind of removal had occurred, whether the removal was scheduled or not, whether a failure was justified or not, whether it was induced or not, etc... A combination of those codes will describe the removal [Leung et al. 2007].

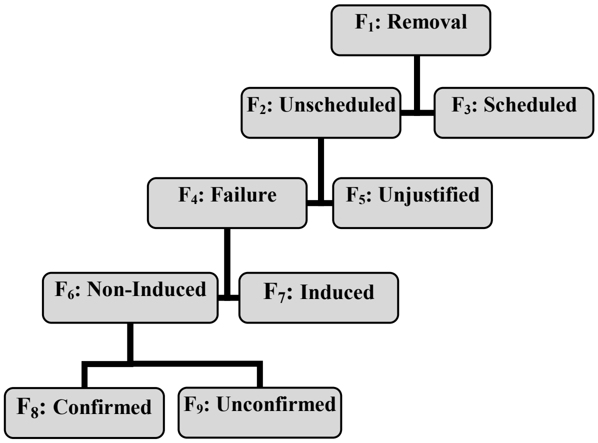

Figure 3-1: Fault Confirmation Codes as presented in [Leung et al. 2007] describe the nature of the removal

2- Reason for Removal Codes (RRC): These codes describe the cause or mode of failure of the component (e.g. leak in sealing area, wear in bearing, etc...). One component can have a mixture of reasons for removal describing the same failure incident. For a given component type, q known possible RRC codes may exist.

3- Time-to-Removal (TTR): This is the amount of time (i.e. number flight hours) the component spent in service before it was removed. This is measured as the time between installation and removal. This number is sometimes multiplied by a constant d between 0.5 and 1 that is chosen based on some known utilisation characteristics of the aircraft the component was used in [Leung et al. 2007].

Classification within the Maintenance Process

The ability to use the indicators mentioned above depends on where, in the maintenance process, rogue component detection occurs. Implementation of the LAD algorithm can take place at one of two points in the process: before or after the component enters the repair shop.

By performing the detection before the repair stage, any unnecessary resources that may be expended on a rogue component can be saved. However, the disadvantage of detecting rogue components at this point is that Fault Confirmation Codes cannot be used as indicators. Consequently, the LAD algorithm would have to rely on the two remaining indicators to come up with a decision about the rogueness of a certain component.

Performing classification after the component undergoes repairs allows for the utilization of the FCC codes as inputs to the LAD algorithm. The presence of additional evidence leads to a more educated judgment of the components` maintenance status. The disadvantage, however, is that these codes are hard to procure given the current structure of the aircraft maintenance process. In many cases aircraft component maintenance is administered by the OEMs themselves. Communication between the aircraft operator and the OEMs on maintenance matters is usually minimal. Consequently, obtaining information regarding what occurs in the repair shop may not always be possible.

It is worth mentioning that the extraction of maintenance data from an aircraft operator`s logs is in many cases a tedious task. This is largely due to the fact that most maintenance data is generated for the goal of record keeping and not for utilization as an asset for the purpose of condition based maintenance.

LAD Training Table

To our knowledge, previous uses of LAD did not require taking into account historical values of the same indicator in generating the patterns and decision functions. However, in this situation, the nature of the observations from which a classification decision is obtained necessitates the incorporation of historical data into the set of LAD attributes.

Values for the 3 indicators mentioned above are recorded for every single removal instance of a single component. In the case of rogue detection, each component in the population has exhibited many removals in its life time. Therefore, the removal records of a certain component contain values for these indicators for every removal instance. Additionally, some components are older than others, and some have exhibited more failures than others. Therefore not all removal records contain the same amount of data.

In view of the above, it is difficult to obtain input observation vectors having a unique form if all the available information for each component is used. As such, the observation vector used to train the LAD algorithm is limited to 9 non-binary indicators representing the 3 most recent FCC, RRC, and TTR values of a component. Example: we are given a training set of 4 rogue components and 4 non-rogue components, where each component has a recorded number of removals ranging between 3 and 9. We choose to limit the removal data we are going to look at to the 3 most recent removal incidents.

The reasoning behind this is that whatever pattern we would find will be clear to us by looking at the most recent removal data. This reasoning is deduced from [Leung et al. 2007] where, in the visual graph obtained through the CH-method, the most relevant and pertinent data are the ones found in the top right corner, which actually represent the data obtained from the 3 most recent removals of the components.

The number three (i.e. the last three removals) is used in many cases in [Leung et al. 2004] and [Carroll 2008] when calculating factors or triggering rogue flags. While we will use this for illustration purposes throughout this paper, this number can be modified within the algorithm without any major structural change. Ultimately, the goal is to be able to consider the entire history of a certain component in the classification process.

The LAD methodology explained above has been adapted into a software program called CBM-LAD written in C++ at École Polytechnique de Montréal. This software is capable of treating the rogue detection problem explained above.

3.7 Results

The CBM-LAD software was used on real component data obtained from the maintenance department of NetJets Inc. The data was extracted from the maintenance records of 61 airplanes during a period stretching from March 28, 1999 to June 20, 2009. These records consist of 576 removal instances belonging to 150 turbo compressors. From the records of each component an observation vector was obtained as explained in the sections above. Of the available 150, 68 were used to train the LAD decision model and 74 to test the resulting model; the rest were discarded as incomplete records. The data shown in table 1 show a portion of the training data. Two of the components shown in the table were judged as rogue by maintenance professionals. There are, in all, 13 negative observations representing normal components (grey) and 2 positive observations representing rogue components (white). Each observation represents information obtained from the removal records of one component with a unique serial number S/N$_i$. It is assumed here that the LAD algorithm is implemented before the component enters the repair shop. FCC codes are consequently absent from the table.

Table 3-1: Non-Binarized Training Data

	Reason-for-Removal Code			**Time-to-Removal Codes**			
	Last	2nd Last	3rd Last	Last Removal	2nd Last Removal	3rd Last Removal	
1	2	0	0	413.73	99999	99999	Negative (Non-Rogue)
2	2	0	0	21.99	99999	99999	
3	2	0	0	366.81	99999	99999	
4	3	2	0	194.72	477.67	99999	
5	2	3	2	1288.99	196.70	125.15	
6	2	0	0	266.76	99999	99999	
7	2	0	0	1503.23	99999	99999	
8	2	0	0	0	99999	99999	
9	5	2	2	1045.42	1451.63	133.41	
10	2	0	0	212.47	99999	99999	

11	3	0	0	616	99999	99999	
12	2	3	0	284.08	539.97	99999	
13	3	0	0	304	99999	99999	
14	2	2	2	144.08	57.6	132.7	Positive (Rogue)
15	2	2	2	204	281.20	83.7	

The 150 components did not enter into service at the same time, thus not all components exhibited 3 removals within their lifespan as most components exhibit one or two removals per 3 years for this type of part. This phenomenon is dealt with in table 3 by placing close to infinity Time-to-Removal values (99999 days) and the 0 code for reason-for-removal to illustrate the absence of such events. The LAD table is then used for training the algorithm and producing a decision function.

The decision model was trained 3 times, each time with a different maximum allowable pattern degree. The degrees used were 2, 3, and 4. The resulting 3 decision models were tested in each case using the data set composed of 74 observations reserved for that purpose. The value τ was randomly set to 0.2 for all three decision models. The number of binary attributes obtained and the number of positive and negative patterns found for each decision model are shown in table 2.

Table 3-2: Pattern Numbers found for each Decision Model

	Max. Degree 1	**Max. Degree 2**	**Max. Degree 3**
No. Binary Attributes	49		
No. Negative Patterns	25	125	125
No. Positive Patterns	7	274	330

The values of the discriminant function Δ for the 15 observations shown in the previous table are presented in table 3 for the 3 decision models obtained. The table shows that the score of the discriminant function is positive for the positive observations and negative for negative observations.

Table 3-3: The value of Discriminant function Δ for all 3 Decision Models

Observation	**Max. Degree 2**	**Max. Degree 3**	**Max. Degree 4**
1	-0.7617	-0.8542	-0.8542
2	-0.6191	-0.6093	-0.6093

3	-0.6925	-0.6194	-0.6194
4	-0.8513	-0.8664	-0.8664
5	-0.5479	-0.7999	-0.7575
6	-0.6130	-0.6793	-0.6793
7	-0.7617	-0.8542	-0.8542
8	-0.6925	-0.6194	-0.6194
9	-0.6864	-0.6893	-0.6893
10	-0.6486	-0.4982	-0.4918
11	-0.6864	-0.6893	-0.6893
12	-0.6884	-0.8441	-0.8441
13	-0.6864	-0.6893	-0.6893
14	0.3794	0.5035	0.4137
15	0.1978	0.4535	0.3958

The results, part of which is shown in table 2, reveal that the detection has been done successfully. The scores of the discriminant function for all the observations of the testing set give a negative value for the normal (non-rogue) components and a positive value for rogue components. However, since the threshold for considering an observation unclassified is ± 0.2, the result was not 100% successful for all pattern decision models.

In order to evaluate the performance of the resulting decision models, a number of performance measures are calculated using the proportions shown in table 4. Each observation classified by the LAD decision model can be in one of the 6 situations shown in the table. The letters *a*, *b*, *c*, *d*, *e*, and *f* represent the proportions of classified observations found in each of these 6 situations.

Table 3-4: Calculating the Quality of Classification

		Classification Result		
		Positive	Negative	Unclassified
True Class	Positive	*A*	*c* (Type II error)	*e*
	Negative	*b* (Type I error)	*d*	*f*

The values *a* and *d* represent the proportion of positive and negative observations that are correctly classified, respectively. The values *c* and *b* are the proportion of positive and negative observations that are falsely classified, respectively. The values *e* and *f* represent the proportion of positive and negative observations that remain unclassified, respectively. The performance measures obtained from these values are:

Quality of Classification:
$$Q = \frac{a+d}{2} + \frac{e+f}{4} \tag{10}$$

The true positive rate:
$$TP = \frac{a}{a+c+e} \tag{11}$$

The false positive rate:
$$FP = \frac{b}{b+d+f} \tag{12}$$

The true negative rate:
$$TN = \frac{d}{b+d+f} \tag{13}$$

The false negative rate:
$$FN = \frac{c}{a+c+e} \tag{14}$$

The results for the three decision models obtained are shown in table 5.

Table 3-5: Performance Measures of the 3 Decision Models

	Max. Degree 2	**Max. Degree 3**	**Max. Degree 4**
Q	0.8263	0.9965	0.9965
TP	0.3333	1	1
FP	0	0	0
TN	0.9718	0.9859	0.9859
FN	0	0	0

The results in table 5 show that the 3 decision models obtained have a high classification quality Q. The classification quality increases significantly with the increase in maximum pattern size from 2 to 3 bits. Degree 3 and degree 4 show an equal performance. Additionally, all three models resulted in zero false alarms; i.e. no rogue components were misclassified as non-rogue and vice versa. The true positive and true negative values also increased with the increase in maximum patterns size from 2 to 3. However, these values will change if the threshold τ is changed from the set value of ± 0.2. If τ is decreased, for example, the number of false alarms will increase and the quality of classification measure will change.

In comparing the discriminant function values obtained from the models with maximum pattern degrees 2, 3, and 4, we notice that the scores for the positive observations increase in the degree 3 model and then decrease slightly for the degree 4 model. The rise in the values of Δ can be explained by the fact that a much higher number of positive patterns was found in the degree 3 model (274) compared to the degree 2 model (7). The scores, however, decrease slightly again in the degree 4 model even though the number of positive patterns found increases to 330. This decrease can be attributed to the fact that the third model generated degree 4 positive and negative patterns which are too specific, thus leading to a decrease in the discriminatory power of Δ, as a higher degree pattern has a lower chance of covering an observation than a lower degree one. In addition, judging from the rogue component characteristics discovered manually by experts and discussed in the sections above, any pattern we expect to find must relate 3 consecutive events to each other, as explained in the sections above. Degree 3 patterns therefore are more meaningful than patterns of the other degrees.

The advantage of the decision models obtained through LAD, besides their accuracy, is the interpretability of the decisions obtained from it. For example, one negative pattern found in the second decision model (degree 3) is: $b_{14}b_{18}b_{22}$. This pattern translates verbally to the statement: *"The three last reason-for-removal codes are all of value 2".*

Such a pattern is exactly what we would expect to have given the characteristics for rogue components explained above. The ability to translate the patterns leading to the decision to logical statements that could be understood by any maintenance technician is unique to the LAD technique.

3.8 Conclusion

In this paper, we studied rogue components, which plague the asset management programs in the aviation industry. We explained how these rogues develop and discussed their impact on the entire asset management program. We then described how to control such components and proposed the use of LAD as a decision model to solve the problem of detecting them.

Testing results showed that the LAD technique is capable of detecting rogue components automatically through feeding the components` performance history into the LAD algorithm. The automatic detection of rogue components solves the problem of having to sift through thousands of removal records in order to evaluate each component visually. A major advantage of its utilization in rogue component detection is, therefore, the huge amount of time and resources that it can potentially save. LAD is capable of accomplishing in seconds something which takes days currently in the industry.

The financial benefits are also evident. By a 1995 estimate, the maintenance burden to an airline of one rogue component is $50,000 (US). If 100 rogue components are detected using the LAD decision model, an aircraft operator's asset management system saves $5 million in maintenance costs alone. In addition to saved costs, early detection of such components also increases the safety and overall performance of the operator.

In applying LAD to rogue component detection, we were capable of generating the patterns that maintenance experts expected to see. As such, the ability of LAD to reduce dependence on their subjective opinions was demonstrated. The advantage of LAD, though, is that it is capable of detecting new patterns without previous knowledge or any aid from maintenance experts.

The automation of the evaluation of records for rogue component detection is a big step towards achieving condition based maintenance in aviation. It is however apparent that for achieving full CBM implementation in the industry, maintenance records must be regarded as assets and not as mere tracking logs.

Further work is going on in developing the LAD algorithm to include more sophisticated pattern recognition techniques. Further investigation of more effective measures to deal with incomplete data is also underway.

3.9 References

Boros E., Hammer P. (2000). An Implementation of Logical Analysis of Data. *IEEE Transactions on Knowledge and Data Engineering*, 12(2), 292-306.

Cama Y., Hammer P.L., Ibaraki T. (1988). Cause-effect relationships and partially defined Boolean functions. *Annals of Operations Research*, 16(1), 299-325.

Carroll T. (2008). The Statistical Outliers are in Control of Asset Management. *The Maintenance and Reliability Conference MARCON 2008*. Tennessee, USA.

Flint P. (2007). Balancing Act: Rising demand for MRO services occurs against a backdrop of steady market evolution. *Air Transport World*. November 2007, 46-47. http://www.atwonline.com/magazine/article.html?articleID=2115 Accessed: 24 May 2009.

Kilpi J., Vepsalainen A.P.J. (2004). Pooling of spare components between airlines. *Journal of Air Transport Management,* 10(2), 137-146.

Leung T. , Carroll T., Hung M., Tsang A., Chung W. (2007). The Carroll-Hung Method for Component Reliability Mapping in Aircraft Maintenance. *Quality and Reliability Engineering International*, 23, 137-154.

OAG, (2008). Global MRO spend on military aviation to increase by 14.9% over the next decade, reports OAG. *Official Airline Guide.*

http://www.oag.com/oagcorporate/pressreleases/08+GLOBAL+MRO+SPEND+ON+MILITARY+AVIATION+TO+INCREASE.html Accessed 29 April 2009.

Salamanca D., (2008). Logical Analysis of Data Applied in Condition Based Maintenance. *M.Sc. Thesis, Department of. Industrial Engineering, Ecole Polytechnique de Montreal*.

Teal C., Sorensen D. (2001). Condition based maintenance [aircraft wiring], *The 20th Conference on Digital Avionics Systems, Daytona Beach , FL*, 1, 3B2/1-3B2/7

Chapter 4 Fault Diagnosis of Power Transformers Using Logical Analysis of Data

Mohamad-Ali Mortada, Soumaya Yacout, Aouni Lakis

Submitted to: DISCRETE APPLIED MATHEMATICS
Manuscript Number: DA572

4.1 Abstract

Logical Analysis of Data (LAD) is a machine learning data mining approach based on pattern recognition that has been relatively untested in the field of Condition Based Maintenance (CBM). This paper proposes a novel multi-layer LAD classification approach based on Mixed 0-1 Integer and Linear Programming (MILP) for pattern generation. The generated patterns are used for the diagnosis of faults in power transformers. The LAD based classifier is applied on two sets of transformer data using different processed input features. The results of LAD are then compared with other classification approaches. The results show that LAD offers a performance that is comparable to most conventional classifiers with the added advantage of result interpretability.

4.2 Introduction

Condition Based Maintenance (CBM) is defined as the monitoring of an asset's health in order to judge whether and when maintenance is required. The end goal of a CBM strategy is either the diagnosis or prognosis of an asset's health or both. Whereas prognostics deal with the probability that an asset is going to fail, diagnostics deals with the detection and identification of the fault before the complete failure of the asset. A CBM strategy consists of three stages: Data Acquisition, Data Processing, and Decision Making. In the first stage, data that could be helpful in judging the state of the system is collected. In the following stage, meaningful information, in the form of attributes, is extracted from the collected data using suitable processing techniques. Finally, a classification technique is used to decide on what maintenance action to take.

Many different classification techniques have been used for fault diagnosis within Condition Based Maintenance. Some commonly used classification techniques are based on hypothesis testing [1], cluster analysis [2], Hidden Markov Models (HMM) [3], and Support Vector Machines (SVM) [4-6]. While each of these statistical approaches possesses some advantages, they all share a common disadvantage which is the large assumptions they make about the nature of the input data; for example, by regarding all features as independent and identically distributed. Other popular classifiers are Expert Systems (ES) [7], Fuzzy Logic [8], and Artificial Neural Networks (ANN) [9-11]. The main disadvantage of ES and Fuzzy Logic systems is their large reliance on knowledge that is not always available and that, in many cases, is subjective. The common disadvantage of all neural networks is the black box concept which characterizes their architecture. As such, there is no obvious physical explanation of how the trained model, and consequently a classification decision, came to be. Other classification techniques combined different approaches to come up with hybrid classifiers with relative success [12, 13].

4.3 Problem Description: Power Transformer Fault Diagnosis

The application of CBM to electromechanical equipment through indicator data monitoring has proven an effective method for safeguarding expensive machinery from failure and assuring its continuous operation. One such application is power transformers which are high-priced items that require continuous monitoring in order to detect any fault in their operation before any safety hazards, which may affect the equipment itself and the related power systems, are produced. The most effective attributes used for fault diagnosis in power transformers are obtained using dissolved gas analysis (DGA). DGA was discovered in the late 1960s by R.R Rogers and others in collaboration with a large transformer manufacturer [14]. The analysis relies on the phenomenon of chemical breakdown of oil into hydrocarbon gases at certain environmental conditions to detect faults in the transformer. The composition of the gases produced can be related to the type of fault that has occurred even though many non fault-related factors have a considerable influence. Several expert based diagnostic interpretations of the patterns relating gas composition to specific faults have been published in the last decades. Most of them take into account the presence of the gases H_2, CH_4, C_2H_6, C_2H_4 and C_2H_2 and their ratios with respect to each other. The most common diagnostic interpretations are the Burton & Davis ratios (1972), the Rogers ratios (1974), the Duval Triangle (1970s), the Dornenburg ratios, and most recently, the revised IEC 60599 (1999) of the International Electrotechnical Commission [14,15].

IEC 60599 uses the outputs of three ratios obtained from the above five gases to classify power transformers into 5 fault states [15]:

1. Partial Discharges (PD): Discharges of either cold plasma (corona) type, which could possibly result in X-wax deposition, or sparking type.
2. Low Energy Discharges (D1): Discharges of low energy resulting in larger paper perforations, tracking, or carbon particles in oil.
3. High Energy Discharges (D2): Discharges of high energy resulting in extensive carbonization, metal fusion, and, possibly, tripping of the equipment.

4. Thermal Faults (T1/T2): Below 300°C (T1) when paper has turned brownish, above 300°C (T2) when paper has carbonized.
5. Thermal Faults above 700°C (T3): When oil carbonization, metal coloration, or fusion occurs.

Traditionally, gas levels are obtained by either taking oil samples manually or through an on-line gas monitor connected to the oil circuit and arranged to acquire samples and analyze them at regular intervals. After obtaining the data and calculating the required ratios, classification is done based on one of the expert based diagnostic approaches mentioned above. However, in the past years, several researchers have studied ways to automate this CBM classification process by applying different data processing techniques combined with the classification approaches described in the previous section.

4.3.1 Data Processing

A good classification is only as good as the information it is based on. For this reason, relevant information is extracted from raw gas data using different processing techniques and fed to the classifiers. For many automated classification approaches, this processing step is simply the manual selection of key gases and the normalization of the gas content levels [16, 17]. Lv et al. [6] processed the raw gas data by calculating the relative content of the five characteristic gases in addition to the absolute content of each sample. The relative content of a gas is obtained by dividing its content in a sample by the highest content among the measured gases in that sample. Absolute content is calculated as the logarithm of the highest gas content within the sample. For other classification approaches, the data is processed by calculating the same ratios described by the expert based classifiers. In Yang et al. [18], the three gas ratios proposed by Rogers and the four described by Dornenburg were each tested as inputs. In Cho et al. [19] the three IEC ratios were calculated in addition to their mean, root mean square (rms), variance, skewness, kurtosis, and normalized fifth to tenth moments; after which a clonal selection algorithm (CSA) was used for feature extraction. Naresh et al. [20] used a neural fuzzy model in order to select the most important gases for the diagnosis of the transformer faults. As will be discussed later, this paper uses several of the data processing techniques mentioned above to test the proposed LAD based classification approach.

4.3.2 Automated Classification Approaches

Many of the classification approaches described in the Introduction have been used in automating the power transformer fault diagnosis process. The most common approaches have been fuzzy logic [21], ANN [19, 22], combined fuzzy neural techniques [16, 17, 20], and SVM [5, 19]. Castro et al. [16] used a fuzzy logic system to translate the implicit knowledge obtained from a neural network based classification approach that distinguishes between thermal and discharge faults into explicit rules. Naresh et al. [20] used Self-Organizing Maps (SOM) and Fuzzy Logic (FL) on a data set of defective transformers to identify the 5 IEC fault types exclusively without any fault detection. Cho et al. [19] compared ANN and SVM based classifiers that perform fault diagnosis based on the IEC fault types. Lv et al. [6] developed an SVM cascade classifier that detects transformer faults and identifies three types of failure: Low Energy Discharges (D1), High Energy Discharges (D2), and Thermal Faults (T1/T2/T3). They compare their results to ANN, ES, Fuzzy Logic, and ANN/ES based classifiers. The results posted in the above studies show that SVM based classifiers give the best classification accuracy, however, at the expense of decision result interpretability. Fuzzy Logic and Fuzzy Neural approaches have shown good performance with the possibility of the generation of rules similarly to expert based manual classifiers. Yang et al. [18] tested an automatic diagnostic technique based on Association Rule Mining (ARM) that generated association rules which can be used to classify data into thermal, PD, arcing, and no fault. They compared their results to ANN, SVM, and k-nearest neighbour based classifiers.

Logical Analysis of Data (LAD) is a supervised learning pattern recognition approach that was first developed by PL Hammer in 1986 [23] as a Boolean technique that identifies the effects of a certain event by investigating a set of factors representing all the possible effects of that event. LAD has been successfully used as a classification technique primarily in medical applications [24-27]. The first application of LAD in condition based maintenance of mechanical systems was reported recently in [28, 29]. To the best of our knowledge, this is the first time that LAD is used for fault diagnosis using dissolved gas data in power transformers. Several variations on the LAD technique have been developed over the years targeting particularly the pattern generation step, which is the corner stone of the LAD classification approach. In this paper, we propose a modified version of the pattern generation algorithm that better suits the machine fault diagnosis problem. We study its performance by testing it on power transformer data and comparing it to the performance of other classifiers used on the same application. In the first part of this paper we explain the LAD based classification approach and introduce the modified MILP based pattern generation algorithm. Then we present the implementation of the classification approach for the diagnosis of faults in power transformers using DGA data and test it on data sets obtained from two different sources. We compare the obtained results with those from other classifiers. Finally, we analyse the results and give a brief conclusion.

4.4 LAD Classification Approach

A LAD classifier is a supervised learning data mining approach that functions by finding distinctive patterns which can separate data into 2 classes. Training based on LAD requires the presence of a pre-classified database from which the classification model can be extracted. This pre-classified database is referred to in this text as a *training data set* and consists of instances or observations whose outcome is already known. The training phase of a LAD classifier can thus be divided into 3 broad steps: Data Binarization, Pattern Generation, and Theory Formation.

4.4.1 Data Binarization

As LAD operates by finding patterns in Boolean data, the binarization of input data is the first step in training the classifier. The input data for any classifier consists of a set of features or attributes and their values at different instances or for different observations. Attributes can be divided into three categories: discrete unordered, discrete ordered, and numerical. The most common type of attributes encountered in fault diagnosis of machinery is numerical. A numerical attribute (e.g. Vibration Amplitude, H_2 Gas Content, etc...) can take any real number for a value. The binarization of such attributes depends on the different values they take in the training data set. The binarization method used in this paper starts by aligning the observed values of the numerical attribute in increasing order. For an attribute A, the result of the alignment could be displayed as follows: $u_A^{(1)} < u_A^{(2)} < \ldots < u_A^{(m)}$ where $u_A^{(m)}$ is the highest value observed for feature A in the training data set and m is the total number of distinct values that numerical feature A has taken within the training data set. Naturally, the inequality $m \le N$, such that N is the total number of observations in the training data set, holds. After the alignment, cut-points are introduced between each consecutive pair of values $u_A^{(i)}$ and $u_A^{(i+1)}$ for which there exists observations $u'_A \in S^+$ and $u''_A \in S^-$ such that $u'_A = u_A^{(i)}$ and $u''_A = u_A^{(i+1)}$ or vice-versa, where S^+ is the set of observations in the training data set belonging to the first (positive) class and S^- is the set representing the second (negative) class. The easiest method to calculate the cut-point is by averaging the two consecutive values. As a result, the cut-point is calculated as $\alpha_A = \left(u_A^{(i)} + u_A^{(i+1)}\right)/2$, and the binary attribute b_A created by this cut-point is defined as:

$$b_A = \begin{cases} 1 & if \quad u_A \geq \alpha_A \\ 0 & if \quad u_A < \alpha_A \end{cases} \tag{1}$$

The total number of binary attributes describing a numerical attribute A is simply equal to the number of cut-points. After data binarization, the total amount of binary attributes that represent the numerical attributes of the training data set is usually substantially higher than the number of original numerical features.

4.4.2 Pattern Generation

After binarizing the training data set, a pattern generation algorithm is used to extract patterns from it. This is arguably the most critical stage of the LAD algorithm, and as such, has been the subject of a large amount of research. A pattern of degree d, in its strictest sense, is defined as a conjunction of d *literals* such that it is true for at least one observations of a class $S^+(S^-)$ and not true for the observations of the other class $S^-(S^+)$. A literal is a binary variable x or $\bar{x}$ where x is true for a certain observation in the binarized training set if its corresponding binary attribute b is equal to 1 for that observation and false otherwise. Consequently $\bar{x}$ is true when binary attribute b is 0 and false when it is 1. A pattern that is true for some observations of one class is said to *cover* these observations and as such, belongs to that particular class. Consequently, for a two class classifier, a generated pattern can be one of two types: a positive pattern ($\boldsymbol{p}^+$) or negative pattern ($\boldsymbol{p}^-$). Throughout this text, as the operations involving the generation of positive and negative patterns are symmetric, we shall refer to a pattern belonging to a certain class and its opposite by the notations $*$ and $\bar{*}$, where $*$ can be replace by $+$ and $\bar{*}$ by $-$ when referring to positive pattern generation and vice versa. Consequently, a pattern of a certain class is referred to by the notation $\boldsymbol{p}^*$, and the set of observations of the opposite class is referred to by the notation $\bar{S}^*$. However, when the class of the pattern is irrelevant, we shall refer to it in the text by the basic notation $\boldsymbol{p}$.

Four special non-mutually exclusive types of patterns have been defined in [29-33]: *prime*, *spanned*, *strong* and *maximal.* A *prime* pattern has the least number of *literals* possible such that if any *literal* is dropped, it will cease to be a pattern. Prime patterns are more global since they cover more observations, and are easily interpretable. A pattern is qualified as *spanned* if, for the same covered observations, it is composed of the maximum number of literals possible; i.e. if any other literal is added, then it will cease to be a pattern. Spanned patterns, by definition, possess less generalization power than other pattern types. A pattern $\boldsymbol{p}_i$ is defined as *strong* if no other pattern $\boldsymbol{p}_j$ exists such that the set C_{p_i} of observations that are covered by $\boldsymbol{p}_i$ is a subset of C_{p_j}. Hammer et al. [30] offers a detailed description of these three types of patterns. A *maximal* pattern $\boldsymbol{p}_i^*$ for a certain observation in S^* is one which has the most coverage among all the patterns covering that specific observation.

Many techniques for pattern generation have been described in the literature. The earliest of these techniques were enumeration based [31-33]. Enumeration based techniques lead to the generation of all possible patterns of a certain type from the training data set, which takes up large computational time. Other pattern generation techniques are based on heuristics and linear approximation [34, 35]. Most recently, Ryoo et al. [36] explored algorithms that generate patterns using Mixed 0-1 Integer and Linear Programming (MILP). This MILP based pattern generation approach has been shown to offer equivalent performance with a lower computational time than other pattern generation techniques. A modified version of the MILP algorithm is used in this paper for pattern generation.

Strong Pattern MILP Algorithm

Ryoo et al. proposed different formulations of linear set covering problems to generate different types of patterns. Of interest to this paper is the strong pattern generation set covering problem which will be illustrated here. The set covering problem varies the elements of a vector $\mathbf{w}$, among other variables, in order to minimize an objective function. The Boolean *pattern vector*

$\mathbf{w}(w_1, w_2, \ldots, w_{2q})$ has a dimension that is double the number q of binarized attributes that make up a binarized observation in the training data set. The solution of the set covering problem is a pattern that can be deduced from vector $\mathbf{w}$ in the following manner. The elements $w_1, w_2, \ldots, w_q$ of vector $\mathbf{w}$ are relative to the literals that make up the pattern such that if $w_j = 1$ then the literal x_j is included in pattern $\boldsymbol{p}$. Similarly, the elements $w_{q+1}, w_{q+2}, \ldots, w_{2q}$ are such that if $w_{q+j} = 1$ then literal $\overline{x}_j$ is included in pattern $\boldsymbol{p}$. For example, for a binary training data set composed of 4 binary attributes $q = 4$, a pattern $\boldsymbol{p} = \overline{x}_1 \wedge x_2 \wedge \overline{x}_4$ is represented by the Boolean vector $\mathbf{w}(0,1,0,0,1,0,0,1)$. Naturally, a pattern cannot include both the literal x_j and its negation $\overline{x}_j$ at the same time. For that reason the following condition must hold:

$$w_j + w_{q+1} \leq 1 \quad j = 1,2,\ldots,q$$

Each observation i in the binarized training set either belongs to the class set S^* or $\overline{S}^*$. Assuming we want to generate a strong pattern $\boldsymbol{p}$, we associate each observation i with the Boolean vector $\mathbf{a_i}(a_{i,1}, a_{i,2}, \ldots, a_{i,q}, \ldots a_{i,2q})$ such that $a_{i,j} = 1$ $(j = 1,2,\ldots,q)$ if $b_j = 1$ in i and $a_{i,j+q} = 1$ $(j = 1,2,\ldots,q)$ if $b_j = 0$ in i.The same condition of mutual exclusivity holds for $a_{i,j}$ and $a_{i,j+q}$ where both cannot be 1 at the same time.

A linear set covering algorithm that generates one pattern $\boldsymbol{p}^*$ has, as variables: the Boolean pattern vector $\mathbf{w}$ associated with $\boldsymbol{p}^*$, the degree d of $\boldsymbol{p}^*$, and the coverage vector $\mathbf{y}$. For generating a pattern $\boldsymbol{p}^*$, $\mathbf{y}$ is a Boolean vector whose number of elements N^* equals the number of observations in the binarized training set S^*. The elements y_i of vector $\mathbf{y}$ are the variables to minimize in the set covering problem such that $y_i = 0$ when observation $i \in S^*$ is covered by pattern $\boldsymbol{p}^*$ and 1 otherwise. The resulting MILP model for generating a strong pattern as described in [36] is as follows:

$$\min_{\mathbf{w},\mathbf{y},\mathrm{d}} \sum_{i \in S^*} y_i$$

$$s.t. \begin{cases} \sum_{j=1}^{2q} a_{i,j} w_j + q y_i \geq d & \forall i \in S^* & (a) \\ \sum_{j=1}^{2q} a_{i,j} w_j \leq d - 1 & \forall i \in S^{\bar{*}} & (b) \\ w_j + w_{q+1} \leq 1 & j = 1,2,\ldots,q & (c) \\ \sum_{j=1}^{2q} w_j = d \quad (d) & & \\ 1 \leq d \leq q \quad (e) & & \\ \mathbf{w} \in \{0,1\}^{2q} \quad (f) & & \\ \mathbf{y} \in \{0,1\}^{N^*} \quad (g) & & \end{cases} \qquad (2)$$

The objective of the above MILP problem is to minimize the number of observations in S^* that are not covered by pattern $\boldsymbol{p}^*$ while at the time satisfying the following 2 major sets of conditions:

1. A pattern $\boldsymbol{p}^*$should cover observations in S^*but does not have to cover all the observations inS^*(condition (a)). If the resulting pattern covers an observation $i \in S^*$, then $\sum_{j=1}^{2q} a_{i,j} w_j = d$, where $d$is the degree of the pattern. However, if an observation $i \in S^*$ is not covered, then $\sum_{j=1}^{2q} a_{i,j} w_j < d$ and the value qy_iadded to the left side of condition (a) is there to compensate.
2. A pattern $\boldsymbol{p}^*$ should not cover any observation $i \in S^{\bar{*}}$, and for that reason, condition (b) $\sum_{j=1}^{2q} a_{i,j} w_j \leq d - 1$ should hold for all such observations.

The proof that the above MILP model generates a strong pattern is provided by Ryoo et al. [36]. To generate a strong positive pattern we replace $*$ and $\bar{*}$ in the above functions by $+$ and $-$ respectively. The opposite applies for a strong negative pattern. An MILP solver tool is used to solve the above set covering problem and generate the resulting strong pattern. For generating a different pattern type, minor modifications to the above model are required.

Modified MILP Algorithm

The MILP problem shown in (2) generates the pattern that covers the highest number of observations from the training data set. Ryoo et al. [36] suggests a scheme that loops the above algorithm as many times as necessary until all the observations of the training data set are covered by a pattern. This however is inconvenient for two reasons:

1. The setup proposed in Ryoo et al. [36] generates the minimum number of patterns required to cover the training data set. As a result, only one pattern is sufficient to cover each observation. The classifier resulting from this small number of patterns has a low discriminating power; a term described in Boros et al. [31] and refers to the differentiating power between the 2 classes of the data set. This is due to the fact that the diagnosis of a certain observation would be based on the presence or absence of a single or a few patterns.
2. For applications such as the diagnosis of power transformers using dissolved gas analysis the explanatory power of the classifier is very important as the generated patterns give the conditions under which the machine will be faulty. The MILP models and setups developed in [36] generate the minimum amount of patterns sufficient to cover a training data set. For our application, a more reasonable number of patterns is needed in order to find all the conditions under which a certain defect appears in a transformer.

The modifications proposed in this paper aim to enhance the performance of the MILP pattern generation model by tackling the weaknesses discussed above. To achieve that, two alterations are suggested, to the MILP model and to the looping scheme that generates the entire pattern set, in order to increase the amount of patterns that cover each observation in the training data set.

Starting from the strong MILP pattern generation algorithm explained in the previous section we introduce a series of constraints to allow it to generate, in addition to the strongest pattern, the subsequent strong patterns iteratively. To do that, we need to save each vector $\mathbf{w}_k$ associated with pattern $\boldsymbol{p}_k^*$ formed by one solution of the MILP algorithm as vector $\mathbf{v}_k$ in the set of vectors $\mathbf{V}$. Naturally, the first iteration of the algorithm generates the strongest pattern possible and does not contain any added constraints. However, one constraint is added to the MILP model each time a new pattern is generated. The added constraints simply prevent the algorithm from finding the same pattern found in the previous solutions of the MILP algorithm. This set of constraints (h) is added to the model shown in (2) and can be represented as follows:

$$\sum_{j=1}^{2q} r_{k,j} w_j \le d_k - 1 \quad \forall \mathbf{r}_k \in \mathbf{R} \qquad \text{(h)}$$

Where a vector $\mathbf{r}_k \left(r_{k,1}, r_{k,2}, \ldots, r_{k,q}, \ldots r_{k,2q} \right)$ in set $\mathbf{R}$ is assigned to each $\mathbf{v}_k$ in $\mathbf{V}$ such that:

$$r_{k,j} = \begin{cases} 1 & if \quad v_{k,j} = 1 \\ -1 & if \quad v_{k,j} = 0 \end{cases} \quad j = \{1,2,\ldots,2q\}$$

If the candidate new pattern represented by the Boolean vector $\mathbf{w}\left(w_1, w_2, \ldots, w_{2q}\right)$ is identical to an existing pattern $\boldsymbol{p}_k^*$, then the sum $\sum_{j=1}^{2q} r_{k,j} w_j$ will be equal to the number of degrees d_k of pattern $\boldsymbol{p}_k^*$. In all other cases the inequality holds. We shall refer to the new MILP model formed by the addition of the set of constraints (h) as MILP-h.

Theorem 1. *Let* $\boldsymbol{P}^*$ *be a set of the strongest patterns covering* S^*. *Let* $\boldsymbol{\mathsf{P}}^*$ *be the set of all patterns covering* S^* *.If* $\boldsymbol{P}^* \neq \boldsymbol{\mathsf{P}}^*$ *then MILP-h, admits a feasible solution* $(\mathbf{w}, \mathbf{y}, d)$ *that can be translated to a pattern* $\boldsymbol{p}^*$ *of degree* d *:*

$$\boldsymbol{p}^* = \bigwedge_{\substack{w_j=1 \\ 0 \le j \le q}} a_j \bigwedge_{\substack{w_j=1 \\ q < j \le 2q}} \bar{a}_j \qquad (3)$$

Proof. Ryoo et al. [36] proved, without the presence of the set of conditions (h), that the MILP model generates an optimal feasible solution that constitutes a strong pattern. Following the logic of this proof, we demonstrate that the new MILP-h model has at least one feasible solution that is a pattern. As explained previously, a conjunction of literals is said to be a pattern $\boldsymbol{p}^*$ if it covers at least one observation in S^* and no observations in $\bar{S}^*$. As $\boldsymbol{P}^* \neq \boldsymbol{\mathsf{P}}^*$, then there exists at least one observation in S^* for which not all the patterns that cover it are in $\boldsymbol{P}^*$. Let us take an observation $\alpha \in S^*$ that satisfies that particular criterion. We set the elements of vector $\mathbf{w}$ in MILP-h as equal to those of vector $\mathbf{a}_\alpha$ associated with observation α; therefore $w_j = 1$ if $a_{\alpha,j} = 1$ and $w_j = 0$ otherwise. We also set $y_i = 1$ for all observations $i \in S^*$, $i \neq \alpha$. As a result, the variable d representing the candidate pattern`s degree is automatically set to q. The above solution satisfies the conditions (a) through (g) of MILP-h. As proven in Ryoo et al. [36], such a solution is a pattern $\boldsymbol{p}^*$ of degree q as it covers at least one observation $\alpha \in S^*$ and no observations in $S^{\bar{*}}$ through conditions (a) and (b) respectively. This yields:

$$\boldsymbol{p}^*(\mathbf{a}_\alpha) = \prod_{\substack{w_j=1 \\ 0\leq j\leq 2q}} a_{\alpha,j} = 1$$

$$\boldsymbol{p}^*(\mathbf{a}_i) = \prod_{\substack{w_j=1 \\ 0\leq j\leq 2q}} a_{i,j} = 0 \quad \forall i \in \overline{S^*}$$

Where $\mathbf{a}_\alpha$ is the Boolean vector associated with observation $\alpha \in S^*$ and $\mathbf{a}_i$ is the Boolean vector associated with any observation $i \in S^{\bar{*}}$. As the set $\boldsymbol{P}^*$ contains the strongest patterns and as observation $\alpha \in S^*$ admits more than one pattern, the solution shown to satisfy conditions (a) to (g) also covers condition (h) as such a pattern is the least possible strong pattern since it covers only observation $\alpha \in S^*$ and the exact similar observations. Consequently, MILP-h admits at least one solution that qualifies as a pattern.

Theorem 2. *If $(\mathbf{w}, \mathbf{y}, d)$ is a feasible solution of MILP-h, then the resulting pattern $\mathbf{p}^*$ is the strongest possible pattern whose degree is d.*

Proof. This proof is similar to what was demonstrated in Ryoo et al. [36] for the model in (2). As the objective function of the MILP-h model minimizes the sum $\sum_{i \in S^*} y_i$, the optimal solution to the model ultimately tries to minimize the number of observations in S^* that are not covered by the constructed pattern. As a result, an optimal solution to MILP-h is the strongest pattern that does not exist in the set $\boldsymbol{P}^*$ of already found patterns.

The modified MILP-h model generates a single pattern. Following is the modification of the iteration scheme used for looping the MILP-h model in order to cover each observation by at least l patterns:

```
begin
  for $* \in \{+,-\}$ do
    $\boldsymbol{P}^* = \phi$
    $\boldsymbol{C}^* : c_i = 0 \quad \forall i \in S^*$
    while $S^* \neq \phi$ do
      Formulate and solve MILP – h
      Form $\boldsymbol{p}^*$ using (3)
      $\boldsymbol{P}^* \leftarrow \boldsymbol{P}^* \cup \{\boldsymbol{p}^*\}$
      Update Vector $\boldsymbol{C}^*$
      $S^* \leftarrow S^* \setminus \{i \in S^* : c_i = l\}$
      Update MILP - h Constraints
    end While
  end for
end
```

The preset constant value$l$, which we will call the *discriminating factor*, represents the minimum number of strong patterns that each observation in the training data set must be covered by. Vectors $\boldsymbol{C}^+$ and $\boldsymbol{C}^-$ are, respectively, of equal dimension to the total number of positive and negative observations in the training data set, and their elements are initially set to zero. Upon each loop of the above procedure, a pattern is generated. The looping stops when each observation is covered by l patterns.

In training a LAD classifier using the procedure described above, we have the option of altering the discriminating power of the resulting classifier by modifying the value of the constant l. A larger value of l leads to more patterns covering every observation in the training set, thus increasing the discriminating power, however, at the expense of increased computation time.

The above modifications are based on the strong pattern generation MILP model proposed in Ryoo et al. [36]. The same modifications can also be made to the models generating strong prime and strong spanned patterns also described in Ryoo et al. [36].

4.4.3 Theory Formation

The final step in LAD after the patterns are generated is the formation of a classification rule based on the patterns found. This is achieved by forming a discriminant function $\Delta(\mathbf{a}_i)$, composed of normalized weighted patterns, that has a score ranging between -1 and +1. New observations from outside the training data set are binarized according to the same rules created in the data binarization step and associated with a Boolean vector $\mathbf{a}$ similarly to the training set observations. Consequently, the new observation vectors are input to the discriminant function for classification. Observations with a positive score are classified into class 1 (positive class) and those with a negative score are classified into class 2 (negative class). A score of zero means no decision has been reached. For any observation vector $\mathbf{a}_i$, the discriminant function has the following form:

$$\Delta(\mathbf{a}_i) = \sum_{\substack{k=1 \\ p^+ \in P^+}}^{K} m_k^+ \boldsymbol{p}_k^+(\mathbf{a}_i) - \sum_{\substack{j=1 \\ p^- \in P^-}}^{J} m_j^- \boldsymbol{p}_j^-(\mathbf{a}_i)$$

Where K and J are the numbers of positive and negative patterns found, respectively. $\boldsymbol{p}_k^*(\mathbf{a}_i) = 1$ if the observation associated with $\mathbf{a}_i$ is covered by pattern $\boldsymbol{p}_k^*$ and $\boldsymbol{p}_k^*(\mathbf{a}_i) = 0$ otherwise. The value $m_k^* \geq 0$ is a normalized weight assigned to each pattern. These weights can be calculated in different ways. The method used in this paper obtains the weight of a pattern $\boldsymbol{p}_k^*$ by counting the number of observations it covers within the training data set and dividing that amount by the total coverage of all the patterns that belong to the set $\boldsymbol{P}^*$. This would create a normalized set of weights such that $\sum m_k^* = 1$.

4.5 Implementation

To demonstrate the performance of LAD in the diagnosis of faults in transformers, we setup two configurations of multilayer LAD classifiers and test these configurations on two sets of data obtained from Duval & DePablo [15] and Lv et al. [6]. The chosen configurations are similar to those setup in Lv et al. [6] and Naresh et al. [20] so as to compare the results obtained from the multilayer LAD classifier with those obtained using other techniques.

4.5.1 Databases

Database 1: The first database obtained from Duval & DePablo [15] is composed of 117 samples of defective transformers compiled from different sources. Each sample is composed of the content of the 5 gases discussed in section 1 along with CO and CO_2 gas content, in addition to the fault state of each sample as diagnosed by industry experts using diagnostic engineering tools [15]. In total 5 fault states are identified as per the IEC standards mentioned in section 1. Of the 117 samples, 9 samples are identified as PD, 26 as D1, 48 as D2, 16 as T1/T2, and 18 as T3.

Database 2: The second database describes historical samples of the 5 characteristic gases obtained from a 500kV transformer located in a substation of the South China Electric Power Company [6]. From a total of 75 samples in the database, 9 are reported to be normal, 38 suffer from thermal heating, 21 are faulty due to high energy discharge, and 7 due to low energy discharge [6].

4.5.2 Data Processing Techniques

The databases described above contain observations of the content levels of the 5 gases described in section 1. Before using this data to train and test the classifier, they are first processed by extracting new numerical attributes from the 5 gas content values. As mentioned earlier, different processing methods are used in the literature. In this paper, four processing methods are used to extract 4 distinct processed data sets from each database. The results from each data set are compared in section 5 to test the impact of the processing methods on the accuracy. The first processing method extracts the 4 Dornenburg ratios from the 5 characteristic gases. The second processing method extracts the Rogers ratios. The third case combines the 5 unique Dornenburg and Rogers ratios. The last processing method calculates the relative content of the five characteristic gases in addition to the absolute information of each observation as done in Lv et al. [6]. The data sets obtained from the four processing techniques described above is composed of four, three, five, and six numerical attributes respectively. Table 1 shows the numerical features obtained from each processing technique. The table shows the numerical attributes $\mathrm{y_i}$ resulting from each processing technique. c_i in the Absolute Content method represents the absolute gas content of the five characteristics gases were $i = 1,2,\ldots,5$.

Table 4-1: The numerical attributes of the four data sets obtained using the four different processing methods.

Processing Scheme	Numerical Attributes
Dornenburg Ratios [37]	$y_1 = \frac{\mathrm{CH_4}}{\mathrm{H_2}}; y_2 = \frac{\mathrm{C_2H_2}}{\mathrm{C_2H_4}}; y_3 = \frac{\mathrm{C_2H_6}}{\mathrm{C_2H_2}}; y_4 = \frac{\mathrm{C_2H_2}}{\mathrm{CH_4}}$
Rogers Ratios [37]	$y_2 = \frac{\mathrm{C_2H_2}}{\mathrm{C_2H_4}}; y_2 = \frac{\mathrm{CH_4}}{\mathrm{H_2}}; y_3 = \frac{\mathrm{C_2H_4}}{\mathrm{C_2H_6}}$
Combined Ratios (Dornenburg & Rogers)	$y_1 = \frac{\mathrm{CH_4}}{\mathrm{H_2}}; y_2 = \frac{\mathrm{C_2H_2}}{\mathrm{C_2H_4}}; y_3 = \frac{\mathrm{C_2H_6}}{\mathrm{C_2H_2}}; y_4 = \frac{\mathrm{C_2H_2}}{\mathrm{CH_4}}; y_5 = \frac{\mathrm{C_2H_4}}{\mathrm{C_2H_6}}$

Absolute Content Method [6]	$y_1 = \frac{H_2}{\max\limits_{i=1\to5}(y_i)}; y_2 = \frac{CH_4}{\max\limits_{i=1\to5}(y_i)}; y_3 = \frac{C_2H_6}{\max\limits_{i=1\to5}(y_i)}; y_4 = \frac{C_2H_4}{\max\limits_{i=1\to5}(y_i)};$ $y_5 = \frac{C_2H_2}{\max\limits_{i=1\to5}(y_i)}; y_6 = \log_{10}\left(\max\limits_{i=1\to5} c_i\right)$

4.5.3 Multilayer LAD configurations

Two configurations of multilayer LAD classifiers are tested each using one of the data sets described above. The reason for choosing these two configurations is the possibility of comparing our results with those obtained in Naresh et al. [20] and Lv et al. [6].

Configuration 1 [20]: Four LAD classifiers are placed in three layers and trained using database 1. The first classifier is trained to separate thermal defect observations from non-thermal ones. The second classifier is trained to separate the non thermal defects into partial discharge defects (PD) and energy discharge defects. The third classifier is trained to classify thermal defects into high temperature (T3) ad low temperature (T1/T2) defects. Finally, classifier four separates energy defect observations into high energy (D1) discharge and low energy discharge (D2) defects. As a result the global classification scheme performs fault identification by separating observations into the 5 fault types described in the IEC standard. This configuration, shown in figure 1, is similar to that used in [20] where a neural fuzzy classifier was used on data set 1 to identify the 5 IEC transformer fault types.

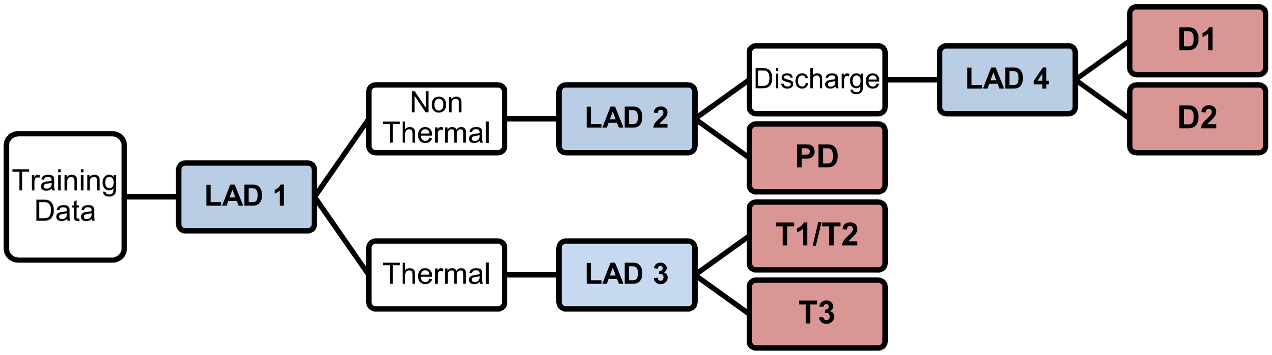

Figure 4-1: LAD multilayer classifiers: configuration 1. In order to classify observations into 5 fault types, 4 LAD classifiers must be trained and applied in 3 layers.

Configuration 2 [6]: Three LAD classifiers are placed in cascade and trained using database 2. The first classifier is trained to separate normal observations from faulty ones. The second classifier is trained to separate thermal heating defects from energy discharge defects. Finally, the third classifier is trained to classify high energy discharge and low energy discharge defects. As a result, the global classification scheme performs fault detection and fault identification. Normal data is separated from faulty data, which is in turn classified into 3 fault types: Thermal, High Energy Discharge, and Low Energy Discharge defects. This configuration, shown in figure 2, is identical to the one used in Lv et al. [6], where a multilayer SVM classification scheme is tested on data set 2 described above and compared with ANN, FL, expert system (ES), and ANN/ES based classifiers.

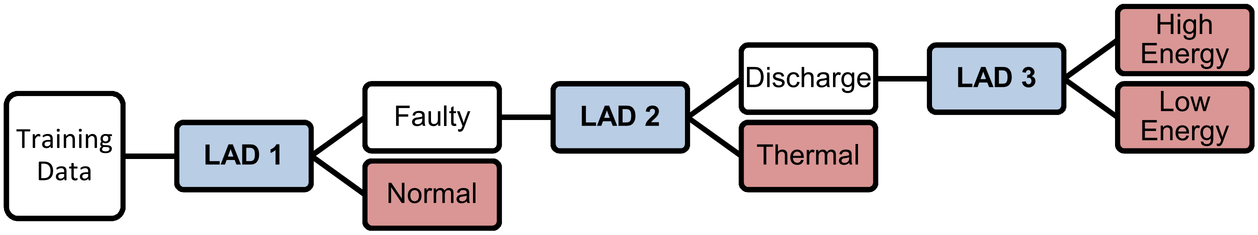

Figure 4-2: LAD multilayer classifiers: configuration 2. In order to identify faulty observations and classify them into 3 fault types, 3 LAD classifiers must be trained and applied in cascade.

The LAD based classifiers were implemented in VS2008 C++ programming language using the *LP_Solve* C++ library [38]. The software, called cbmLAD, takes as input training data in the form of tables written in excel or text files, generates the patterns, and calculates the discriminant function automatically. Testing data is entered in a similar way. The output is a text file containing the classification result for each element in the data set, the patterns found, and their interpretation in terms of the original attributes; in this case the gas content ratios.

4.6 Experimental Results

Experiment 1: In this experiment, configuration 1 was used on database 1. As described above, the data was processed in four different ways to produce 4 distinct data sets. In Naresh et al. [20] a data set of 87 samples was chosen randomly from the database of 117 samples in order to train their proposed classification model. The remaining 30 samples were used to test the model. In order to be able to compare the experiment results with those obtained in Naresh et al. [20], we adopt the same training set and testing set sizes referenced there. As a result, of the 117 samples, 87 were picked randomly to train the multilayer LAD classifier and the remaining 30 samples were used for testing. Four runs of training and testing were performed on each of the four processed data set. On each run the discriminating factor l was changed to 1,5,10, or 20. Consequently, 16 sets of results were obtained from this configuration. The accuracy of each classification is calculated according to the following formula:

$$Accuracy = \frac{a}{a+b}$$

Where a is the number of correctly diagnosed observations and b is the number of falsely diagnosed observations. The accuracy results and percentages of non-classified observations (No Decision Rate) of this experiment are shown in table 2.

Table 4-2: The Diagnostic Accuracy and No Decision Rate (NDR) of the 16 classification results obtained in experiment 1.

	Dornenburg Ratio Data		Rogers Ratio Data		Combined Ratios Data		Absolute Content Data	
	Accuracy	NDR	Accuracy	NDR	Accuracy	NDR	Accuracy	NDR
$l=1$	66.67%	10.00%	65.52%	3.33%	72.00%	16.67%	52.17%	23.33%
$l=5$	70.00%	0.00%	72.41%	3.33%	80.00%	16.67%	59.26%	10.00%
$l=10$	76.67%	0.00%	72.41%	3.33%	80.00%	16.67%	57.69%	13.33%
$l=20$	76.67%	0.00%	70.00%	0.00%	**81.48%**	10.00%	57.69%	13.33%

The results show that the level of accuracy increases by increasing the discriminating factor l from 1 to 5. Four two processed data sets the accuracy continues increasing with l whereas it reaches a maximum and then decreases again for the other two data sets. The accuracy levels obtained using the Combined ratios data set are the highest for all levels of l. The absolute highest accuracy of 81.48% was achieved using the Combined Ratios data set at $l=20$. However, Table 3 shows that the highest number of correct classifications was achieved using Dornenburg Ratio Data at $l=20$. Table 3 also shows that, for three of the four classifiers obtained using the four processed data sets, the hardest defect state to identify is the thermal faults above 700°C (T3), where 1 observation out of 4 was correctly classified. For the Dornenburg ratio data set, this can be justified by the fact that the Dornenburg expert designed ratios were not intended for detecting difference between high temperature and low temperature thermal faults. Similarly the Absolute Content ratios were not used in Lv et al. [6] to differentiate between the different thermal states. If the split between high temperature fault state (T1/T2) and low temperature fault state (T3) is disregarded, the accuracy level of the best classifier would jump to 92.6%.

Table 4-3: Number of correctly classified observations for each fault type in experiment 1 in the 4 data sets trained at $l = 20$.

	Dornenburg Ratio	Rogers Ratio Data	Combined Ratios Data	Absolute Content Data
PD	3/3	3/3	3/3	3/3
D1	8/9	7/9	7/9	2/9
D2	8/10	8/10	8/10	3/10
T1/T2	3/4	3/4	3/4	1/4
T3	1/4	1/4	1/4	3/4
Total	23/30	22/30	22/30	12/30

Figure 3 gives a comparison between the classification accuracy obtained using the best LAD based classifier and those obtained in Naresh et al. [20] using Rogers Ratio Expert Method (RREM) (76.67%), Fuzzy C-means method (FCM) (50%), Generalized Regression Neural Network method (GRNN) (80%), Fuzzy Clustering and Radial Basis Function Neural Network (RBNN) (60%), and the Integrated Neural Fuzzy Approach with feature selection (INF+FS) (96.67%) and without (50%) (INF). It should be noted here that despite the fact that the data source is the same for all the classifiers being compared, different data processing techniques were applied in most cases. The comparison shows that the LAD based classifier fairs well at 81.48% in relation to the conventional classification methods, with the second highest result. The highest accuracy among all the compared methods is the Integrated Neural Fuzzy Approach combined with a competitive learning feature selection technique to process the data before classification [20].

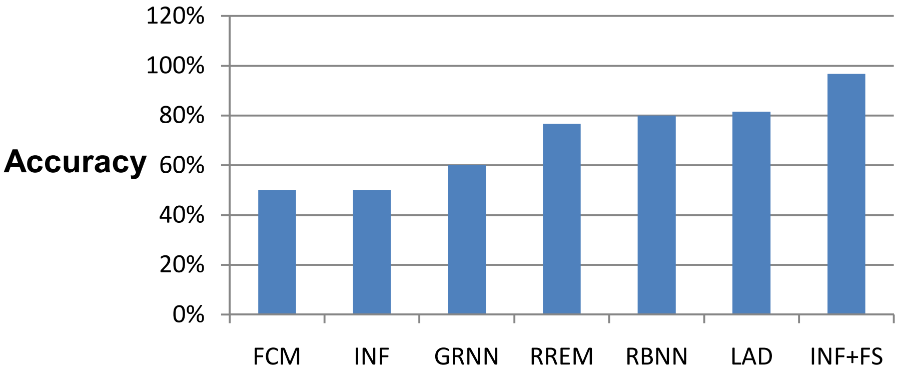

Figure 4-3: Classification Accuracy Comparison: Experiment 1. The best LAD based classifier accuracy obtained in experiment 1 using the Combined ratios data set at $l = 20$ compared to the accuracy of other classifiers obtained from Naresh et al. [20]. LAD achieves the second highest result.

In the case of this experiment, the total amount of training time ranged between 1 and 15 seconds for each classifier on an ordinary computer depending on the level of l and type of data set used.

Experiment 2: For this experiment, the 3 layer cascaded classifier, labelled as configuration 2 above, was trained and tested using data sets obtained from database 2 which is the same as the one used in Lv et al. [6]. The data was processed using the four techniques described earlier in order to produce 4 distinct data sets. In order to compare our results with Lv et al. [6], we used the same training and testing data sets published there [6]. As a result, 50 samples were used for training the multilayer LAD classifier and 25 samples were used for testing, 4 of which are normal, 13 defective due to thermal heating, 2 defective due to high energy discharge and 6 due to low energy discharge. Before using the testing data white noise at 5% was added to compensate for the small size of the set, as done in Lv et al. [6]. The LAD based classifiers were trained at $l = 1,5,10$ and 20 to investigate the effect of the modifications to the algorithm. Table 4 displays the results of the classification in each case. As with experiment 1, the diagnosis accuracy increased with the increase in the discriminating factor l from 1 for every data set type, after which the accuracy continued to increase in two cases. The best classification accuracy, 92%, was obtained using the absolute content data at $l = 10$ and $l = 20$.

Table 4-4: The Diagnostic Accuracy and No Decision Rate (NDR) of the 16 classification results obtained in experiment 2.

	Dornenburg Ratio Data		Rogers Ratio Data		Combined Ratios Data		Absolute Content Data	
	Accuracy	NDR	Accuracy	NDR	Accuracy	NDR	Accuracy	NDR
$l = 1$	75.00%	3.33%	78.26%	6.67%	80.95%	13.33%	86.96%	6.67%
$l = 5$	79.17%	3.33%	83.33%	3.33%	86.36%	10.00%	90.91%	10.00%
$l = 10$	79.17%	3.33%	83.33%	3.33%	86.36%	10.00%	**92.00%**	0.00%
$l = 20$	83.33%	3.33%	83.33%	3.33%	82.61%	6.67%	**92.00%**	0.00%

As with the previous experiment the multilayer LAD classifier accuracy was compared to the accuracies obtained using other classification techniques. Figure 4 shows the result of the comparison of LAD with classifiers based on neural networks (ANN) (92.76%), expert systems (ES) (89.34%), fuzzy logic (FL) (92.32%), neural expert systems (ANNES) (93.54%), and support vector machines (SVM) (100%) as published in Lv et al. [6]. The result of the comparison shows that LAD gives a comparable performance to most of the techniques mentioned above, with SVM standing out as the best performer at 100% accuracy. Table 5 shows a comparison of the training times of the classifiers when the same training set is used. LAD and SVM outperform the remaining classification techniques with a training time of less than 1 second.

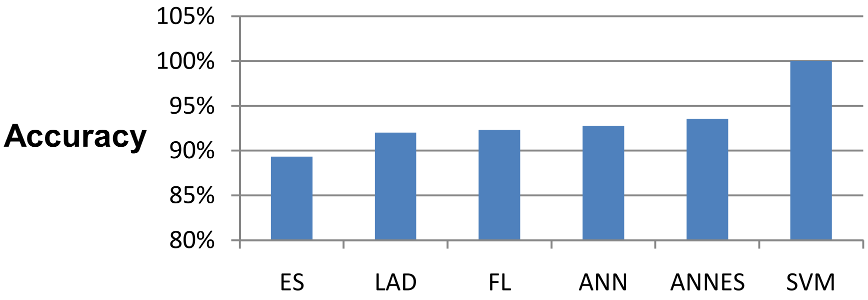

Figure 4-4: Classification Accuracy Comparison: Experiment 2. The best LAD based classifier accuracy obtained in experiment 2 using the Absolute Content data set at $l = 10$ compared to the classification accuracy of other classifiers obtained from Lv et al. [6].

Table 4-5: Best training time for LAD classifiers compared to the times reported for other approaches in Lv et al. [6].

Classification Approach	ANN	ES	FL	ANNES	SVM	LAD
Training Time (s)	81	Absent	82	44	<1	<1

4.7 Pattern Interpretability

In comparing LAD based classification to those studied in Naresh et al. [20] we realise that LAD has performed well against neural networks and fuzzy logic based approaches. In fact, LAD was outperformed only when a novel feature selection approach was paired with an integrated neural fuzzy approach. Comparing LAD to the results in Lv et al. [6] we find that the accuracy achieved was similar to ANN, ES, and Fuzzy Logic approaches but was evidently less than that achieved by SVM. However, LAD possesses the advantage of result interpretability which most other classification approaches do not provide. The patterns generated by LAD can be translated to rules similar to those used in expert systems. To illustrate this advantage we consider, in Table 6, two patterns generated from the LAD classifier using Absolute content input data and at $l = 1$. These two patterns together cover the 20 samples that are defective due to an energy discharge (D1/D2) and none of the 25 samples that are defective due to thermal heating.

Table 4-6: Two patterns for D1/D2 type defects obtained from experiment 2 at $l = 1$ using absolute content data.

	Class	Absolute Content Data Ratios					
		Y1	Y2	Y3	Y4	Y5	Y6
1	D1/D2					>0.16	
2	D1/D2		<0.85		<0.69	<3.35	

Each of the two patterns above covers 85% of the 20 samples from which they are extracted. When transformed into a meaningful rule, pattern 1 states that faulty transformers that have a relative content of gas C_2H_2 that is greater than 0.16 appears always to be defective due to energy discharge (D1/D2) and never due to thermal heating. Pattern 2 suggests that faulty transformers that have a relative content of CH_4 less than 0.85, a relative content of C_2H_4 less than 0.69, and a relative content of C_2H_2 greater than 3.35 appear always to be defective due to energy discharge (D1/D2). Comparing these 2 rules for class D1/D2 to those given by the Rogers Ratio method [37] in Table 7, we notice an overlapping of the ranges.

Table 4-7: The Rogers Ratio fault diagnosis rules used to identify different defect types [37].

Fault Type	Rogers Ratio Method		
	CH_4/H_2	$C2H_2/C_2H_4$	C_2H_4/C_2H_6
No Fault	0.1-1.0	<0.1	<1.0
PD	<0.1	<0.1	<1.0
D1/D2	0.1-1.0	0.1-3.0	>3.0
T1	0.1-1.0	<0.1	1.0-3.0
T2	>1.0	<0.1	1.0-3.0
T3	>1.0	<0.1	>3.0

By using similar features for training the multilayer LAD classifier, we were able to generate a more elaborate set of rules that can classify faults into the five IEC defect types. To illustrate this point, we show, in Table 8, a partial list of patterns deduced from the LAD classifier in experiment 1 using the combined ratios data set at $l = 5$. These patterns can be regarded as a set of rules similar to the Rogers Ratio Method rules. The classification scheme from which these rules where deduced resulted in an 80% accuracy rate. The information obtained from these rules is valuable for the technician on the ground as well as for the engineers working on developing and improving the performance of the transformer. Therefore the slightly lower accuracy rate that LAD suffers from in comparison to SVM is compensated by a gain in knowledge from the interpretation of the patterns generated by LAD.

Table 4-8: Patterns deduced from experiment 1 using the Combine ratios data set at $l = 5$ listed according to the percentage coverage of a pattern with respect to the training data set.

No.	Class	Coverage	Combined Ratios				
			CH_4/H_2	C_2H_2/C_2H_4	C_2H_6/C_2H_2	C_2H_2/CH_4	C_2H_4/C_2H_6
1	PD	100.0%	-	-	>4.79	-	-
2	PD	100.0%	-	-	>4.79	<0.12	-
3	PD	100.0%	>0.01	-	>4.79	<0.12	-
4	PD	100.0%	>0.01	-	>4.79	-	-
5	PD	100.0%	0.01< <0.70	-	>4.79	-	-
6	T3	92.9%	>0.11	-	<22.9	-	>3.85
7	T3	92.9%	>0.11	>0.003	<22.9	-	>3.85
8	T1/T2	83.3%	<255	-	-	-	<3.85
9	T1/T2	83.3%	<255	<0.20	>2.66	-	<3.85
10	T1/T2	83.3%	<255	<0.20	-	-	<3.85
11	T1/T2	83.3%	<255	<0.20	>1.28	-	<3.85
12	D2	73.7%	-	<2.328	<0.366	>1.271	-
13	D2	73.7%	-	<2.328	-	>1.271	>1.591
14	D2	73.7%	-	<2.328	-	1.271<	>1.591

						<4.87	
15	D2	73.7%	-	<2.328	<0.366	1.271< <4.87	>1.591
16	D2	73.7%	-	<2.328	<0.366	>1.271	>1.591
17	D1	64.7%	-	>2.33	<0.04	-	-
18	D1	64.7%	-	>2.33	<0.07	>0.96	>7.94
19	D1	64.7%	-	>2.33	>0.07	-	>7.94
20	D1	64.7%	-	>2.33	>0.07	>0.32	>7.94
21	T1/T2	25.0%	-	>0.01	-	<0.0145	<8.83
22	T1/T2	25.0%	-	>0.01	>1.2	<0.0145	<8.83
23	T1/T2	25.0%	-	>0.01	>1.2	0.015< <0.02	<8.83
24	D2	23.7%	-	<1.84	0.027< <0.366	<1.24	>2.81
25	D2	23.7%	-	<1.84	0.019< <0.366	<1.24	>2.81
26	D1	23.5%	<0.512	>0.97	>0.066	<1.465	<2.81
27	D1	23.5%	0.094< <0.337	<2.19	>0.066	0.315< <1.465	<2.81
28	D1	23.5%	<0.512	0.97< <2.19	>0.066	0.315< <1.465	<2.81
29	D1	23.5%	0.094< <0.512	<2.19	>0.066	<1.465	<2.81
30	D1	23.5%	<0.512	0.97< <2.19	>0.066	<1.465	<2.81
31	D1	17.7%	<0.318	-	-	0.958< <1.271	-
32	D1	11.8%	>0.318	1.106< <1.256	-	0.958< <1.271	-
33	D2	7.89%	>0.257	-	-	-	>5199
34	T3	7.14%	>255	-	>1.288	-	-
35	T3	7.14%	>255	>0.0016	>1.288	-	-
36	T3	7.14%	>255	>0.0016	>2.663	-	-
37	T3	7.14%	>255	>0.0016	>331	-	-

4.8 Conclusion

In this paper, we proposed a novel approach to solve the problem of diagnosis of faults in power transformers using dissolved gas analysis. We modified the application of LAD in order to suit the particular constraints by allowing for more patterns to be generated per observation in the training data set. The results showed a considerable improvement in the classification accuracy due to the proposed improvements, as shown in tables 2 and 4. The LAD based classification approach demonstrated good performance in the detection of faults using conventional feature extraction techniques for the training sets and gave comparable results to other classification approaches as shown in figures 3 and 4. The contribution of the LAD classification approach is that it produces patterns that can be easily interpreted and translated into rules that can be highly beneficial to maintenance experts.

The good performance of LAD in the diagnosis of power transformer faults proves that LAD is a promising approach in Condition Based Maintenance of mechanical systems. The algorithms discussed in this paper can easily be adapted to different diagnostic applications in CBM.

4.9 References

[1] J. Ma, J. Li, Detection of localised defects in rolling element bearings via composite hypothesis test, Mechanical Systems and Signal Processing, 9 (1995) 63-75.

[2] W. Staszewski, K. Worden, G. Tomlinson, Time-frequency analysis in gearbox fault detection using the Wigner-Ville distribution and pattern recognition, Mechanical Systems and Signal Processing, 11 (1997) 673-692.

[3] Z. Li, Z. Wu, Y. He, C. Fulei, Hidden Markov model-based fault diagnostics method in speed-up and speed-down process for rotating machinery, Mechanical Systems and Signal Processing, 19 (2005) 329-339.

[4] K. Christian, N. Mureithi, A. Lakis, M. Thomas, ON THE USE OF TIME SYNCHRONOUS AVERAGING, INDEPENDENT COMPONENT ANALYSIS AND SUPPORT VECTOR MACHINES FOR BEARING FAULT DIAGNOSIS, in: First International Conference on Industrial Risk Engineering, Montreal, 2007.

[5] A. Widodo, B. Yang, T. Han, Combination of independent component analysis and support vector machines for intelligent faults diagnosis of induction motors, Expert Systems with Applications, 32 (2007) 299-312.

[6] G. Lv, H. Cheng, H. Zhai, L. Dong, Fault diagnosis of power transformer based on multi-layer SVM classifier, Electric power systems research, 75 (2005) 9-15.

[7] P. Purkait, S. Chakravorti, An expert system for fault diagnosis in transformers during impulse tests, in: Power Engineering Society Winter Meeting, 2000, pp. 2181 - 2186.

[8] M. Islam, A novel fuzzy logic approach to transformer fault diagnosis, IEEE Transactions on Dielectrics and Electrical Insulation, 7 (2000) 177-186.

[9] J. Spoerre, Application of the cascade correlation algorithm (CCA) to bearing fault classification problems, Computers in Industry, 32 (1997) 295-304.

[10] R. Yam, P. Tse, L. Li, P. Tu, Intelligent predictive decision support system for condition-based maintenance, The International Journal of Advanced Manufacturing Technology, 17 (2001) 383-391.

[11] A. Saxena, A. Saad, Fault diagnosis in rotating mechanical systems using self-organizing maps, Artificial Neural Networks in Engineering (ANNIE04), (2004).

[12] W. Hu, A. Starr, Z. Zhou, A. Leung, An intelligent integrated system scheme for machine tool diagnostics, The International Journal of Advanced Manufacturing Technology, 18 (2001) 836-841.

[13] H. Lee, D. Park, B. Ahn, Y. Park, J. Park, S. Venkata, A fuzzy expert system for the integrated fault diagnosis, IEEE Transactions on Power Delivery, 15 (2000) 833.

[14] M. Heathcote, The J & P transformer book: a practical technology of the power transformer, Elsevier, 2007.

[15] M. Duval, A. DePablo, Interpretation of gas-in-oil analysis using new IEC publication 60599 and IEC TC 10 databases, IEEE Electrical Insulation Magazine, 17 (2001) 31-41.

[16] A. Castro, V. Miranda, An interpretation of neural networks as inference engines with application to transformer failure diagnosis, International Journal of Electrical Power & Energy Systems, 27 (2005) 620-626.

[17] A. Akgundogdu, A. Gozutok, N. Kilic, O. Ucan, FAULT DIAGNOSIS OF POWER TRANSFORMER USING NEURO-FUZZY MODEL, JOURNAL OF ELECTRICAL & ELECTRONICS ENGINEERING, 8 (2008) 699-706.

[18] Z. Yang, W. Tang, A. Shintemirov, Q. Wu, Association rule mining-based dissolved gas analysis for fault diagnosis of power transformers, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 39 (2009) 597-610.

[19] M.-Y. Cho, T.-F. Lee, S.-W. Gau, C.-N. Shih, Power Transformer Fault Diagnosis Using Support Vector Machines and Artificial Neural Networks with Clonal Selection Algorithms Optimization, in: B. Gabrys, R. Howlett, L. Jain (Eds.) Knowledge-Based Intelligent Information and Engineering Systems, Springer Berlin / Heidelberg, 2006, pp. 179-186.

[20] R. Naresh, V. Sharma, M. Vashisth, An integrated neural fuzzy approach for fault diagnosis of transformers, IEEE Transactions on Power Delivery, 23 (2008) 2017.

[21] W. Flores, E. Mombello, J. Jardini, G. Rattá, A Novel Algorithm for the Diagnostics of Power Transformers Using Type-2 Fuzzy Logic Systems, in: IEEE Transmission and Distribution Conference and Exposition, Chicago, 2008, pp. 1-5.

[22] K. Thang, R. Aggarwal, A. McGrail, D. Esp, Analysis of power transformer dissolved gas data using the self-organizing map, IEEE Transactions on Power Delivery, 18 (2003) 1241-1248.

[23] P. Hammer, Partially defined Boolean functions and cause-effect relationships, in Proceedings of International Conference Multi-Attrubute Decision Making Via OR-Based Expert Systems, 1986.

[24] G. Alexe, S. Alexe, D. Axelrod, T. Bonates, I. Lozina, M. Reiss, P. Hammer, Breast cancer prognosis by combinatorial analysis of gene expression data, Breast Cancer Research, 8 (2006) R41.

[25] G. Alexe, S. Alexe, D. Axelrod, P. Hammer, D. Weissmann, Logical analysis of diffuse large B-cell lymphomas, Artificial Intelligence in Medicine, 34 (2005) 235-267.

[26] G. Alexe, S. Alexe, L. Liotta, E. Petricoin, M. Reiss, P. Hammer, Ovarian cancer detection by logical analysis of proteomic data, Proteomics, 4 (2004) 766-783.

[27] S. Abramson, G. Alexe, P. Hammer, J. Kohn, A computational approach to predicting cell growth on polymeric biomaterials, Journal of Biomedical Materials Research Part A, 73 (2005) 116-124.

[28] M. Mortada, T. Carroll, S. Yacout, A. Lakis, Rogue components: their effect and control using logical analysis of data, J Intell Manuf, 1-14.

[29] D. Salamanca, S. Yacout, Condition based maintenance with logical analysis of data, in: 7e Congrès International de genie industriel, Quebec, 2007.

[30] P. Hammer, A. Kogan, B. Simeone, S. Szedmák, Pareto-optimal patterns in logical analysis of data, Discrete Applied Mathematics, 144 (2004) 79-102.

[31] E. Boros, P. Hammer, T. Ibaraki, A. Kogan, E. Mayoraz, I. Muchnik, An implementation of logical analysis of data, IEEE Transactions on Knowledge and Data Engineering, 12 (2000) 292-306.

[32] G. Alexe, S. Alexe, P. Hammer, A. Kogan, Comprehensive vs. comprehensible classifiers in logical analysis of data, Discrete Applied Mathematics, 156 (2008) 870-882.

[33] G. Alexe, P. Hammer, Spanned patterns for the logical analysis of data, Discrete Applied Mathematics, 154 (2006) 1039-1049.

[34] P. Hammer, T. Bonates, Logical analysis of data—an overview: from combinatorial optimization to medical applications, Annals of Operations Research, 148 (2006) 203-225.

[35] T. Bonates, P. Hammer, A. Kogan, Maximum patterns in datasets, Discrete Applied Mathematics, 156 (2008) 846-861.

[36] H. Ryoo, I. Jang, Milp approach to pattern generation in logical analysis of data, Discrete Applied Mathematics, 157 (2009) 749-761.

[37] IEEE Guide for the Interpretation of Gases Generated in Oil-Immersed Transformers, IEEE Std C57.104-1991, (1992) 0_1.

[38] M. Berkelaar, K. Eikland, P. Notebaert, lp solve, open source (mixed-integer) linear programming system, in: (GNU LGPL (Lesser General Public Licence) Version 5.5, 2004.

Chapter 5 Diagnosis of Rotor Bearing Defects Using Logical Analysis of Data

Mohamad-Ali Mortada, Soumaya Yacout, Aouni Lakis

Submitted to: JOURNAL OF QUALITY IN MAINTENANCE ENGINEERING

5.1 Abstract

Purpose – The aim of this paper is to test the applicability and the performance of an approach called Logical Analysis of Data (LAD) in the detection of faults in rotating machinery using vibration signals.

Design/Methodology/Approach – LAD is a supervised learning data mining technique that relies on finding patterns in a binary database to generate decision functions. The hypothesis is that a LAD-based decision model can be used as an effective tool for automatic detection of faults in rolling element bearings. A novel Multiple Integer Linear Programming approach is used to generate patterns for the LAD decision model. Frequency and time-based features are extracted from rotor bearing vibration signals and are pre-processed to be suitable for use with LAD.

Findings – The results show good classification accuracy with both time and time-frequency features.

Practical Implications – The diagnostic tool implemented in the form of software in a production or operations maintenance environment can be very helpful to maintenance experts as it reveals the patterns that lead to the diagnosis in interpretable terms which facilitates efforts to understand the reasons behind the components' failure.

Originality/Value – The proposed modifications to the LAD based decision model which is being tested for the first time in the field of fault detection in rotating machinery lead to improved accuracy results in addition to the added value of result interpretability due to this distinctive property of LAD.

5.2 Introduction

Bearing fault diagnosis using vibration signals is an important tool in detecting faults in rotating machinery. Studies have shown that the pace of crack propagation in rolling element bearings occurs too quickly for any effective maintenance intervention to take place after the cracks are initiated (Qiu et al. 2006). Condition monitoring of bearings using vibration signals can lead to the detection of bearing defects at a much earlier point than the crack propagation stage. Early detection of faults allows enough time for scheduling maintenance, thus preventing catastrophic failure. Vibration signals can be acquired relatively easily using sensors (accelerometers), which make them a valuable source of information for non-invasive diagnostics and condition-based maintenance. Diagnostic decision models can automatically analyze vibration signals and diagnose the state of the bearing, thus constituting an important tool for online condition monitoring of rotating machinery.

Many of the most popular decision models have already been applied for the purpose of automatic fault diagnosis in rotating machinery. For example, Ma and Li (1995) developed a bearing fault detection algorithm using a Hypothesis test with the Neyman-Pearson test statistic. The amplitude of the vibration signals was modelled as a normal distribution of zero mean and a variance determined experimentally. Another application (Sun et al. 2004) used cluster analysis combined with neural networks on processed vibration signals obtained from rotor bearings. The resulting features were fused into a two-dimensional feature space using neural networks, after which clustering was used to obtain a piecewise linear boundary function separating 6 classes representing 5 fault types and the normal state (Sun et al. 2004). Use of a Support Vector Machine (SVM) in Condition Based Maintenance (CBM) was discussed in Christian et al. (2007). In their study, after collection of a set of indicators from vibration signals using independent component analysis (ICA), SVM was used to detect the existence of bearing faults. In Abbasion et al. (2007) SVMs were used for detecting and classifying bearing faults into 7 types. The inputs in this case where 2 time-domain wavelet de-noised vibration signals modelled using negative log Weibull probability density functions. Subrahmanyam and Sujatha (1997) used a Multi-Layer Perceptron (MLP) neural network for fault diagnosis in ball bearings using indicators obtained from time-domain vibration signal processing tools.

Neural networks have been widely used for diagnostic purposes in rotating machinery. Different network architectures and training techniques have been tested with varying levels of success. Subrahmanyam and Sujatha (1997) compared the back propagation (BP) supervised learning MLP neural network to an unsupervised learning adaptive resonance theory ART2 neural network with faster learning time. The study concluded that the unsupervised learning network is 100 times faster to model with excellent fault detection; however the BP-based MLP showed superior classification power in the multi-class case. Baillie and Mathew (1996) compared the performance of radial based functions (RBF) neural networks to BP trained MLPs as well as traditional linear autoregressive models when applied to the diagnosis of faults in rolling element bearings. The results of the study showed that BP-trained MLP’s perform better, are more reliable and are less complex than RBFs. A set of unsupervised learning neural network algorithms have been applied to bearing diagnostics. A common unsupervised learning algorithm is the self-organizing map (SOM), used in Saxena and Saad (2004) for the identification of different types of faults in bearings. A detailed discussion of the different decision models used for fault diagnosis in rotating machinery is presented in Jardine et al. (2006).

The above diagnostic techniques possess distinct advantages that justify their use, but all share one common disadvantage: they are all based on statistical processes which inevitably require some impractical statistical assumptions to be made. These assumptions impose further impractical conditions on the nature of the input data used in forming the automatic diagnostic model. This mainly means that the input data to these decision models must be independent and identically distributed.

Logical Analysis of Data (LAD) is a supervised learning data mining methodology first conceived by P.L. Hammer in 1986 as a Boolean technique to identify the causes of a certain event through investigating a set of factors representing all the possible causes of that event. It has since evolved as an effective data mining technique that relies on extracting patterns from binarized data in order to formulate decision rules that classify data into two classes. The advantage of LAD over traditional techniques is that it is not based on statistical analysis and hence can deal with any form of input data. In addition, the decision model is such that it leads to a clear interpretability of the classification results in terms of the features (attributes) used in achieving the results.

LAD has achieved considerable success in diagnostics in medical applications as noted in Alexe G. et al.(2004), Alexe G. et al. (2005), Alexe G. et al. (2006), and Abramson et al. (2005). However, its application in the field of maintenance has only recently been tested in Salamanca and Yacout (2007), Mortada et al. (2009), and Yacout (2010). The advantages that LAD presents in comparison to the more conventional techniques used so far in the diagnosis of rotating machinery lead us to believe that LAD is a perfect fit for this type of application. The aim of this paper is to apply and test LAD for the first time for automatic detection of faults in rotating machinery. For this purpose, a specially designed LAD-based decision model is used to detect faults in bearings. In order to demonstrate the performance of the resulting diagnostic tool, we test the LAD-based decision model on a practical application. We use a database of vibration signals collected from a test rig of roller element bearings which are run to failure to train and test the LAD-based decision model. We use signals from 2 different bearings that experience different levels of deterioration as well as different combinations of input features to test the diagnostic power of the decision model. The paper starts with a description of the pre-processing mechanisms used here for extracting features from rolling element bearing vibration signals. Then the design of the LAD-based diagnostic tool for automatic detection of faults in bearings is explained. A presentation of the application on which the LAD-based automatic diagnosis tool is tested follows. Test results are described in detail and the performance of the diagnostic tool is evaluated. Finally the achievements are summarized and a conclusion is drawn in the last section of this paper.

5.3 Analysis of Signal Features for fault detection

Vibration signals can be represented in one of three forms: time domain, frequency domain, and time-frequency domain. Experts inspect these vibration signals visually in order to detect whether a component such as a rolling element bearing is defective or not. In many cases, features are extracted from these signals to aid these experts in the decision process. Ideally, an automatic diagnostic decision model that replaces the need for visual inspection will have an entire vibration signal as input. However, in all 3 types of representation domains, the amount of data each signal holds is too large, leading to an unreasonably high computational cost. In addition, classification efficiency is poor due to the inclusion of unnecessary redundant information. To solve this problem, feature extraction techniques are applied to vibration signals to extract more meaningful, condensed information. In this manner, a vibration signal composed of 20,000 data points can be reduced to a feature vector of less than 20 elements.

Feature extraction techniques can be categorized according to the type of signal representation to which they are applied. A detailed study on the advantages and disadvantages of each feature extraction technique in vibration signal analysis is presented in Lakis (2007) and Safizadeh (1999). One conclusion from the above study is that no single technique works best in all situations, which is why it has become the norm in many fault diagnosis research investigations to rely on a set of feature extraction methods to extract the information from the signal database that is input to the classifiers. For example, Abbasion et al. (2007) and Subrahmanyam et al. (1999) used a set of time-domain features for their automatic diagnosis tools whereas Sun et al.

(2007) used a combination of time and frequency-domain features. For the application used in this paper, two sets of features obtained from the time domain and time-frequency domain will be used in different configurations for reasons revealed below. The following sections include a description of these features.

5.3.1 Time-Domain Features

Most time-domain features calculated from the vibration signal are descriptive statistics or high-order statistics. Six time-domain features are used in this paper:

(1) $PeakValue = \max(z(t))$

(2) $RMS - Value = \sqrt{\left(\frac{1}{T}\int_0^T z(t)dt\right)}$

(3) $Std - Deviation = \frac{1}{T}\int_0^T \left(z(t) - \bar{z}\right)^2 dt$

(4) $Crest - Factor = \frac{PeakValue}{RMS - Value}$

(5) $Kurtosis = \frac{\frac{1}{T}\int_0^T \left(z(t) - \bar{z}\right)^4 dt}{\left(\frac{1}{T}\int_0^T \left(z(t) - \bar{z}\right)^2 dt\right)^2}$

(6) $Skewness = \frac{\frac{1}{T}\int_0^T \left(z(t) - \bar{z}\right)^3 dt}{\left(\frac{1}{T}\int_0^T \left(z(t) - \bar{z}\right)^2 dt\right)^{3/2}}$

where $z(t)$ is a periodic signal of period T and mean $\bar{z} = \frac{1}{T}\int_0^T z(t)dt$. The Crest factor is a good indicator of the early stages of bearing failure; a ratio of less than 3 indicates a normal bearing, 3 to 8 indicates fault initiation and 8 to 10 signal fault growth (Archambault et al. 1989). However, in cases where the vibration signal is distorted or when multiple defects are present, these thresholds are less effective in fault detection (Safizadeh 1999).

Kurtosis (Ku) is a high-order statistic that detects peaks in the vibration time signal. If the bearing is functioning normally the PDF of its signal is a Normal distribution and the value of Ku equals 3 (Dyer and Stewart 1977). Faulty bearings generate higher Ku values. The advantage of this technique is its robustness towards variations in the transmission path of vibration signals and its independence from the load and speed of the bearing components (Safizadeh 1999). The disadvantage, similarly to the other methods, is its weak performance with modulated non-stationary signals (Safizadeh 1999). Martin and Honarvar (1995) studied the use of Kurtosis for early detection of faults in rotor bearings. Skewness is a high-order statistic similar to Kurtosis, involving the third order moment.

The main shortcoming of time-domain features as mentioned above is their weak performance with non-stationary signals. Another disadvantage is their vulnerability to what is defined as the "healing" phenomenon which was first described in Williams and Ribadeneira (2001). This occurrence results from the smoothing of newly-formed surface defects by the continuous rolling contact of the bearing. This results in a false decrease in the values of features such as the RMS and Kurtosis values. This decrease continues until more significant damage occurs and spreads to a larger surface. The use of time-domain features on their own is therefore insufficient for early diagnosis of bearing faults.

5.3.2 Time-Frequency Domain Features

Time-Frequency signal representations display the energy distribution of a signal with respect to both time and frequency. Because of this, they are most suited to handle non-stationary signals which are common when machine faults occur. Some common time-frequency representations are the Short-time Fourier Transform (STFT) and the Wigner-Ville distribution. The wavelet transform is a variable resolution time-frequency distribution that possesses certain advantages over other transforms as discussed in Young (1992). In its continuous form, the equation of the wavelet transform is:

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t) \psi * \left(\frac{t-b}{a} \right) dt$$

where $x(t)$represents the vibration signal and the function ψ represents the mother wavelet function. The mother wavelet function can take any possible form provided it satisfies the conditions of admissibility; that is it must be a finite zero average oscillatory function centered at zero with a finite energy. The parameter a in the above equation represents the scale parameter which controls the size of the mother wavelet while b represents the translation (time) parameter which is responsible for the position of the wavelet as it is convoluted with the input signal.

At each scale, $a = {}^{1}\!/\!{}_{f}$, where f is the frequency, the mother wavelet passes over the input signal and is translated each time by a distance b. This is repeated for a variety of a's. The result is a two-dimensional transform of varying time and frequency resolutions. The resulting transform is very data heavy, especially if a continuous wavelet transform is used. In most cases, as in this paper, a dyadic discrete wavelet transform (DWT) is applied. The difference is the discretized shifts in the scale and translation parameters; the signal remains continuous. Interpretation of the transform is done by plotting the Scalogram (obtained by squaring the transform output $|W(a,b)|^2$ to generate a two dimensional mapping where each row represents one scale and contains a set of energies obtained at the scale`s frequency range at particular time intervals.

A traditional feature-extraction method has been applied to extract features from the wavelet transform. The procedure calculates the per-scale total energies of the wavelet transform of the vibration signal. The number of resulting features is equal to the number of scales used in the transform. The obvious advantage of this technique is a huge reduction in the dimensionality of the problem and a consequent increase in processing speed. In summary, feature extraction techniques applied to the vibration signal database can be divided into 2 sets: a time domain features set composed of the 6 time features explained above, and a time-frequency domain features set composed of 12 features representing the scale energies of the discrete wavelet transform. The type of wavelet used and the reasons for using 12 scales will be explained in Section 4.

It is expected that wavelet-based features alone should be sufficient to allow the detection of faults using the LAD-based decision model. To investigate this theory, we shall compare the results obtained from the decision model using wavelet features alone to those obtained using a combination of time-domain and wavelet features. As a result, we will end up with two data sets, one composed of the 12 wavelet features and one composed of the total of 18 features.

The data sets obtained from the vibration signal database will be entered into a decision model that will analyze the features extracted from a vibration signal and automatically diagnose the bearing to which the signal belongs as either normal or defective. As mentioned at the beginning of this paper, a LAD-based decision model is specially designed to perform this task. Many reasons lead us to believe that LAD makes a suitable automatic diagnostic tool for this particular application. The statistical nature of the features extracted from the vibration signals means that a non-statistical technique, such as LAD, is best suited to analyze the data sets obtained from them so as not to make any impractical assumptions. A LAD-based decision model is capable of generating decision rules that can be interpreted in terms of the features forming the data set. The benefit of such a property is that decisions can be understood by experts, unlike other black-box decision models such as neural networks and SVMs. In the following section we describe the LAD diagnostic tool used in this paper.

5.4 LAD Decision Model

A LAD-based diagnostic approach is based on analysis of a training data set containing observations whose state is already classified as either positive $\left(\in S^+\right)$ or negative $\left(\in S^-\right)$, where S^+ and S^- are the sets of positive and negative observations in the training data set. Each observation is composed of the values of certain characteristic features measured or calculated at a certain instant in time. The LAD diagnostic approach consists of three 3 steps: Data Binarization, Pattern Generation, and Theory Formation. In the binarization step the training data set is transformed into a binary (0, 1) data set, one feature at a time. For example, a numerical training set composed of the features (Kurtosis, RMS, and Crest Factor) will be transformed into a binary set where each numerical feature is substituted by at least one binary attribute. The binarization of a continuous numerical feature A and the number of binary attributes needed to replace it are dependent on the number of distinct values that A takes in the training data set. The binarization procedure starts by aligning, in decreasing order, all the distinct values u_A of A in the training data set as follows: ${u_A}^{(1)} > {u_A}^{(2)} > \ldots > {u_A}^{(K)}$ where K is the total number of distinct values ($K \leq N$, where N is the total number of observations). Cut-points are then introduced between each pair of values $u_A^{(i)}$ and $u_A^{(i+1)}$ for which there exists observations $u'_A \in S^+$ and $u''_A \in S^-$ where $u'_A = u_A^{(i)}$ and $u''_A = u_A^{(i+1)}$ or vice-versa. The easiest way to compute the cut-point is by averaging the two values. Thus, for a feature A, a set of cut-points $\alpha_{A,1}, \alpha_{A,2}, \ldots, \alpha_{A,j}, \ldots$ will be obtained such that $\alpha_{A,j} = \left(u_A^{(i)} + u_A^{(i+1)}\right)/2$. As a result, each cut-point $\alpha_{A,j}$ has a corresponding binary attribute value $b_{A,j}$ with defined values:

$$b_{A,j} = \begin{cases} 1 & if \quad u_A \geq \alpha_{A,j} \\ 0 & if \quad u_A < \alpha_{A,j} \end{cases}$$

The total number of binary attributes describing a numerical feature A depends on the number of transitions between distinct values from positive to negative observations and vice versa, and is equal to the number of cut-points.

The next step after transforming the numerical data set into a binary data set is pattern generation. A positive pattern is defined as a conjunction of literals that is true for at least one positive observation and false for all negative observations in the training data set. A negative pattern is defined similarly. A literal is a Boolean variable x or its negation $\bar{x}$. For example, a literal x_i represents a binary attribute b_i such that it is true when $b_i = 1$ and false when $b_i = 0$. Similarly literal $\bar{x}_i$ is true for $b_i = 0$ and false for $b_i = 1$. A pattern composed of the conjunction of d literals is said to have degree d. A pattern is said to cover a certain observation in the binarized training set if all of its literals are true for the binary attribute to which they correspond. For example, consider a binarized data set consisting of 4 binary attributes (b_1, b_2, b_3, b_4). A conjunction of literals $\bar{x}_1 x_2 \bar{x}_3$ is said to be a positive pattern if there happens to be at least one positive observation having the respective values (0,1,0) at attributes (b_1, b_2, b_3) and no negative observation having these values.

Many pattern generation techniques have been studied within LAD, ranging from enumeration to heuristics and linear programming. The latter technique has proven the most promising as demonstrated in Ryoo and Jang (2009). Ryoo and Jang (2009) succeeded in transforming the problem of pattern generation into a set covering problem that can be solved by multiple integer linear programming (MILP) without any approximations. The pattern generation algorithm used here is a novel technique that is based on the one used in Ryoo and Jang (2009). The modifications applied in this paper to the original approach were motivated by the reasons explained below. It is to be emphasized here, however, that the MILP based algorithm is used for pattern generation only. This does not mean that the decision model developed using the obtained patterns is a linear model. Although MILP is used at a certain stage of the model generation, the LAD based decision model remains a non-linear, non-statistical technique.

The original approach MILP based pattern generation approach depicted in Ryoo and Jang (2009) finds the minimum amount of patterns necessary to cover a training data set. This often leads to a decision rule composed of a very small number of patterns. Such a decision model lacks differentiating power between the 2 states of normal and defective classes. This is due to the fact that the diagnosis of a certain observation would be based on the presence or absence of a single or a few patterns. The modifications proposed in this paper increase the number of patterns generated from the same training data set without a significant increase in training time. This change increases the differentiating power between normal and defective classes and improves the interpretability of the results. The following is a description of the modified approach.

To assist in illustrating this algorithm, we repeat some of the description given by Ryoo and Jang (2009) We assume a binarized training data set composed of q binary attributes. The method used for binarizing maintenance data is given at the beginning of this section. We associate each generated pattern $\boldsymbol{p}$ with a Boolean *pattern vector* $\mathbf{w}(w_1, w_2, \ldots, w_{2q})$ whose size n is double that of the binary observation vector, i.e. $n = 2q$. The linear programming problem that is to be created below is a set covering problem that varies the elements of vector $\vec{w}$ in order to minimize an objective function. The pattern resulting from the solution of the set covering problem is deduced from vector $\mathbf{w}$. As such, the elements $w_1, w_2, \ldots, w_q$ of $\mathbf{w}$ are relative to the attributes such that if $w_j = 1$ then the literal x_j is included in pattern $\boldsymbol{p}$. Similarly, the elements $w_{q+1}, w_{q+2}, \ldots, w_{2q}$ are such that if $w_{q+j} = 1$ then literal $\overline{x}_j$ is included in pattern $\boldsymbol{p}$. Naturally, a pattern cannot include both the literal x_j and its negation $\overline{x}_j$ at the same time, hence the condition:

$$w_j + w_{q+1} \leq 1 \quad j = 1,2,\ldots,q \tag{1}$$

Each Boolean pattern vector $\mathbf{w}$ obtained from one solution of the set covering problem is stored as vector $\mathbf{v}$ in the set $\mathbf{V}$ containing all the Boolean pattern vectors of the patterns generated by previous solutions of the problem.

Each positive observation $i \in S^+$ is associated with the Boolean *observation vector* $\mathbf{a_i}(a_{i,1}, a_{i,2}, \ldots, a_{i,q}, \ldots a_{i,2q})$ such that $a_{i,j} = 1$ $(j = 1,2,\ldots,q)$if $b_j = 1$ in i and $a_{i,j+q} = 1$ $(j = 1,2,\ldots,q)$ if $b_j = 0$ in i. The same condition of mutual exclusivity holds for $a_{i,j}$ and $a_{i,j+q}$ where both cannot be 1 at the same time.

The process of generating one positive pattern $\boldsymbol{p}^+$ is formulated into a set covering minimization problem whose decision variables are the pattern vector $\mathbf{w}$, the degree d, and the coverage vector $\mathbf{y}$. For positive pattern generation, $\mathbf{y}$ is a Boolean vector whose number of elements equals the number of positive observations in the binarized training data set. The elements y_i of vector $\mathbf{y}$ are the variables to minimize in the set covering problem such that $y_i = 0$ when observation $i \in S^+$ is covered by pattern $\boldsymbol{p}^+$ and 1 otherwise. The resulting objective function is therefore:

$$\min_{\mathbf{w},\mathbf{y},\mathbf{d}} \sum_{i \in S^+} y_i$$

This logically leads to the minimization of the number of positive observations that are not covered by the generated positive pattern $\boldsymbol{p}^+$ while at the same time satisfying the following 3 major conditions:

(1) If the resulting positive pattern covers a positive observation $i \in S^+$, then the dot product of the pattern vector $\mathbf{w}$ and the observation vector $\mathbf{a_i}$ of each observation covered by $\boldsymbol{p}^+$ must be equal to the degree of $\boldsymbol{p}^+$.

$$\sum_{j=1}^{2q} a_{i,j} w_j = d$$

However, a positive pattern is allowed to not cover all the positive observations, albeit at the expense of a higher value of the objective function. This condition can be described as:

$$\sum_{j=1}^{2q} a_{i,j} w_j + q y_i \geq d \quad \forall i \in S^+ \tag{2}$$

(2) A positive pattern should not cover any negative observations. For this reason the dot product of the pattern vector $\mathbf{w}$ and the observation vector $\mathbf{a}_\mathbf{i}$ of each negative observation $i \in S^-$ must be less than the degree d of $\boldsymbol{p}^+$:

$$\sum_{j=1}^{2q} a_{i,j} w_j \leq d - 1 \quad \forall i \in S^- \tag{3}$$

(3) The set covering problem must not generate the same pattern generated in previous iterations. Additionally, in order to increase the diversity of the patterns, the newly-generated pattern must not be a subset of any of the patterns that have already been generated. The set of conditions that prevents this from happening is:

$$\sum_{j=1}^{2q} v_{k,j} w_j \leq d_k - 1 \quad \forall \mathbf{v}_k \in \mathbf{V} \tag{4}$$

Initially, the set $\mathbf{V}$ is empty and this constraint is not considered. However, upon every implementation of the set covering problem, a positive pattern $\boldsymbol{p_k}^+$ of degree d_k is generated and the Boolean pattern vector $\mathbf{v}_k$ associated with it is added to the set $\mathbf{V}$. As such, a new condition is added for each pattern already found.

The resulting set covering problem, shown below, can be solved using mixed integer linear programming software.

$$\min_{\mathbf{w},\mathbf{y},\mathbf{d}} \sum_{i \in S^+} y_i$$

$$s.t. \quad \begin{cases} (1),(2),(3),(4) \\ \sum_{j=1}^{2q} w_j = d \quad (5) \\ 1 \le d \le q \quad (6) \\ \mathbf{w} \in \{0,1\}^{2q} \quad (7) \\ \mathbf{y} \in \{0,1\}^r \quad (8) \end{cases}$$

Where r is the number of positive observations in S^+. The linear set covering problem above generates the next strongest positive pattern. A pattern $\boldsymbol{p_i}^+$ with a set C_i containing all observations it covers is defined as strong if no other pattern $\boldsymbol{p_j}^+$ exists such as $C_i \subset C_j$. The generation of the next strongest negative pattern proceeds in a similar way by switching the places of the positive observation set S^+ and the negative observation set S^- in the above formulas.

As the above linear problem generates a single pattern, an iterative mechanism is needed to generate an entire set of patterns from which a decision model can be obtained. The original approach involves a mechanism that iterates the above set covering problem as many times as necessary until each observation in the training data sets is covered by a single pattern. The modified approach allows the user to specify the minimum number of patterns that should cover each observation. For this purpose we establish a parameter k which is a user defined number that specifies the minimum number of patterns required to cover each observation in the training data set. For the sake of simplicity we refer to the linear programming problem presented above as $MILP-k$. Additionally, we define R^+ and R^- as the respective sets of positive and negative observations in the binarized training data set that have already been covered by k patterns.

The algorithm starts with two empty sets R^+ and R^- as no patterns have yet been found. Then, upon each solution of $MILP-k$ a new pattern is formed and added to the pattern lists $\boldsymbol{P}^+$ or $\boldsymbol{P}^-$. Next, the coverage of the observations in the training data sets S^+ and S^- is evaluated. Each positive observation in S^+ that is found to be covered by k patterns from the list $\boldsymbol{P}^+$ is removed from the set S^+ and added to R^+. Negative observations are treated in a similar way. At the end of the iterative process all observations in the training set have been covered by at least k patterns.

The final step in the LAD-based decision model is theory formation, whereby the patterns generated in the previous step are used to form a decision function that generates a score ranging between -1 and +1. New observations are binarized and input to the decision function. A negative score output indicates a negative diagnosis which means that the observation belongs to a normally functioning bearing. Similarly, a positive score is a positive diagnosis which refers to the existence of a fault in the bearing. The patterns in the decision function are weighted according to the number of observations they cover in the training data set and then normalized to ensure the final score is within the interval (-1, 1)**.** The resulting decision function has the following form:

$$\Delta(\mathbf{o_i}) = \sum_{\substack{j=1 \\ \boldsymbol{p}_j^+ \in \boldsymbol{P}^+}}^{J} m_j^+ \boldsymbol{p}_j^+(\mathbf{o_i}) - \sum_{\substack{l=1 \\ \boldsymbol{p}_l^- \in \boldsymbol{P}^-}}^{L} n_l^- \boldsymbol{p}_l^-(\mathbf{o_i})$$

Where the Boolean vector $\mathbf{o_i}$ is a binary representation of a non-classified observation, the values $m_j^+ \geq 0$ and $n_l^- \geq 0$ are the weights assigned to a positive pattern $\boldsymbol{p}_j^+$ or a negative pattern $\boldsymbol{p}_l^-$ respectively, where $\boldsymbol{p}_j^+(\mathbf{o_i}) = 1$ if the pattern $\boldsymbol{p}_j^+$ covers the binarized observation $\mathbf{o_i}$ and 0 otherwise. The weight m_j^+ of a positive pattern $\boldsymbol{p}_j^+$ is equal to the number of positive observations covered by $\boldsymbol{p}_j^+$ divided by the sum of coverage of all positive patterns. Negative weights are calculated similarly. As such the set of all positive weights and that of all negative weights are normalized to 1. It is to be reiterated here that the resulting LAD decision system is not a linear function and that linear programming was only used to generate the individual patterns that make up the LAD decision function.

5.5 Application

The LAD based bearing diagnostics model explained in the sections above was tested on a database of signals obtained from the Nationial Science Foundation`s Industry/University Cooperative Research Center for Intelligent Maintenance Systems (IMS) through the NASA prognostic data repository (Lee et al. 2007). The signals were obtained from a test rig setup composed of a motor running at a constant speed of 2000 rpm coupled to a shaft carrying 4 identical bearings, 2 of which are under a constant load of 6000 lb. Accelerometers are placed on each of the four bearing housings. The 4 identical bearings on the shaft are Rexnord ZA-2115 double-row bearings with 16 rollers in each row, 2.815 inch pitch diameter, 0.331 inch roller diameter, and a tapered contact angle of 15.17° (Lee et al. 2007). All the bearings on the shaft were in "brand new" state upon installation.

In Lee et al. (2007), a data acquisition card was used to collect 1 second 20Khz sampled signals every 10 or 20 minutes until catastrophic failure of one of the bearings. The test was carried out during a period of 35 days. According to the test administrators, all failures occurred after exceeding the designed lifetime of 100 million revolutions of the bearings. The signals therefore portray the status of the bearings from their "brand new" state throughout their deterioration phase until complete failure occurs in one of the bearings.

From the characteristics of the bearing the outer race (BPFO), and inner race failure (BPFI) frequencies can be calculated using the formulas found in Shahan and Kamperman (1976) to be 236.4Hz and 296.9Hz respectively.

At the end of the 35 day test period, Qiu et al. (2006) reported a catastrophic failure in bearing 3 with visual evidence of an inner race defect. Additionally, visual evidence of a combined roller element and outer race defect was spotted on bearing 4. The total number of signals collected over the 35 days is 2156.

A LAD-based diagnostic tool should be capable of detecting the faults in the bearings days before the actual catastrophic failure occurs. In order to test the performance of such a diagnostic tool, we use the signals collected from bearings 3 and 4 in the test rig due to the fact that their faultiness had been confirmed visually at the end of the test run time. However, in order to train the LAD decision model and, later-on, test its performance using the database of signals from bearings 3 and 4, we first need to separate the signals collected from these bearings over the entire 35 day period into normal and faulty signals. Expert knowledge is therefore used to classify the data before using it to test the automatic diagnostic decision model described in section III. To do so we rely on visual inspection of some of the features mentioned in Section II of this paper.

As a first measure we plot the kurtosis values of all 2156 signals for both bearings 3 and 4, shown below in Figures 1 and 2. A close examination of these graphs reveals abnormalities starting on day 32 for bearing 3 and day 20 for bearing 4. As kurtosis by itself is not enough to confirm the existence of a fault we corroborate this evidence with further analysis of signals taken from both bearings as follows.

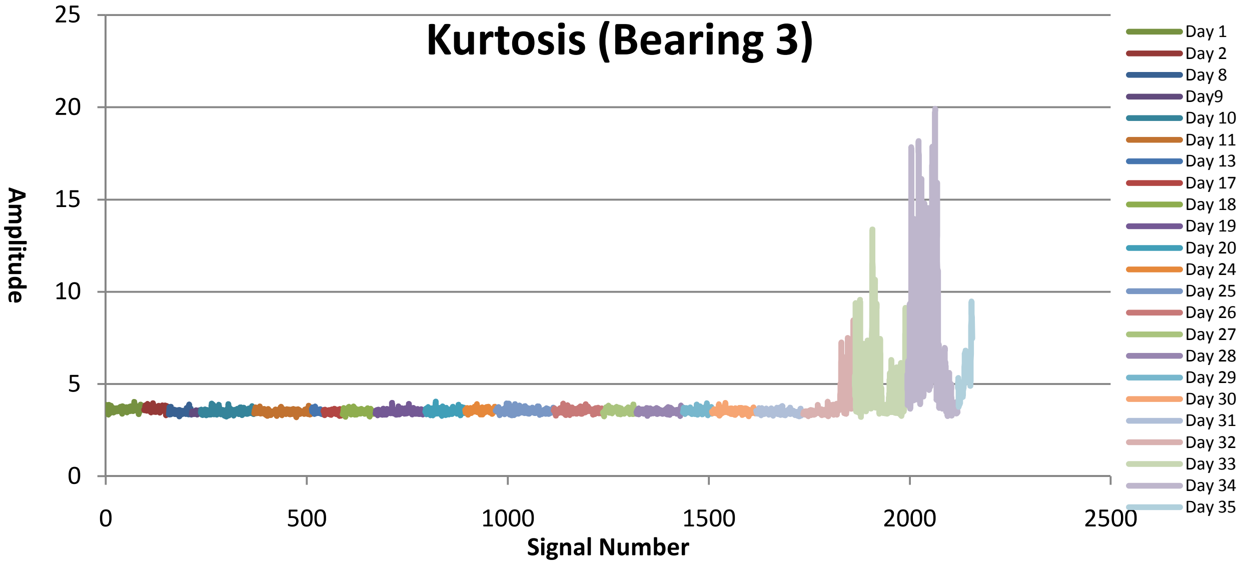

Figure 5-1: The kurtosis plot for signals obtained from bearing 3 over 35 days of testing suggest that the defect started on day 32 of operation, with the kurtosis levels beginning to surge in signals acquired during that time. The kurtosis values calculated in MATLAB were plotted using Excel.

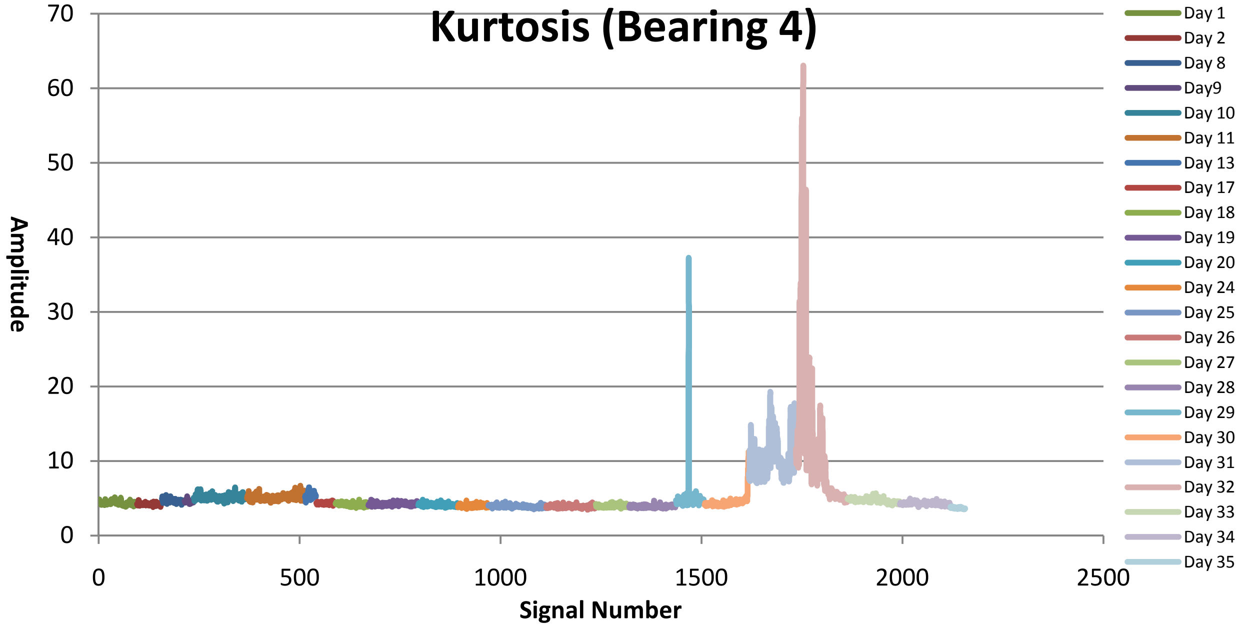

Figure 5-2: The kurtosis plot for signals obtained from bearing 4 over 35 days of testing reveals that a defect happens during day 20 of the test as shown in the first spike of kurtosis level. The graph later becomes flat and smooth, revealing the healing phenomenon before another incident causes another surge (Excel).

The kurtosis graph of bearing 3 presents a surge in the kurtosis value starting on day 32 of the test and continuing until a catastrophic failure is registered on day 35. Examining the RMS value graph of the signals reveals a similar pattern as Kurtosis. A Signal collected from the accelerometer on bearing 3 on day 32 and its power spectrum plotted using MATLAB reveal that the BPFO frequency and its harmonic are visible. However, the amplitude of these frequencies is not high enough compared to the frequencies surrounding them. A plot and power spectrum from a signal collected on day 35 from the same bearing displays more clearly the BPFI and BPFO frequencies. For more proof, we plot the STFT of the signal on day 32 as shown in Figure 3. This reveals high energy levels around the failure frequency areas. As a result of these observations it may be safe to conclude that the signals of bearing 3 from day 32 onwards are faulty signals.

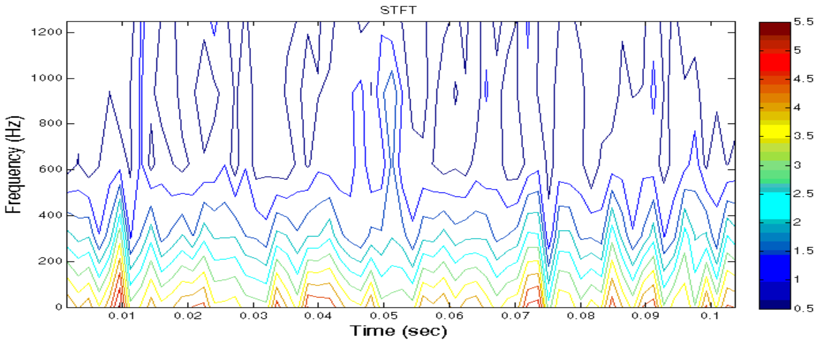

Figure 5-3: A plot of the STFT using MATLAB of a signal collected from bearing 3 on day 32 reveals a peak of high energy at the defective frequencies at each rotational cycle 0.03 seconds of the bearing.

Bearing 4 did not exhibit a catastrophic failure during the 35 day test run. However visual inspection at the end of the test revealed the existence of a fault. The point at which this fault has occurred during the 35 days of testing was investigated using signal analysis. By looking at the kurtosis plot of bearing 4, shown in Figure 2, we notice a surge in amplitude at around day 20 of the test. The kurtosis values return to normal after this incident revealing the healing phenomenon described by Williams and Ribadeneira (2001). This continues until another incident occurs around day 25 of the test. However, after that the kurtosis values drop down again. Therefore, a look at kurtosis values of signals obtained on day 24 or 35 does not decisively reveal the existence of a defect. This provides proof that kurtosis alone is not sufficient for the detection of faults. Moreover, the sudden surge in kurtosis level on day 20 may signify an external shock that is not relevant to the status of the bearing. In order to verify whether this surge is relevant or not we apply STFT to the signal from which the high kurtosis value was obtained. The result of the STFT shown in Figure 4 reveals that the peak exhibited at a specific time does not have a constant amplitude over the entire frequency range. This means that this peak can be interpreted as a defect that has occurred in the bearing itself. As a result, we can conclude that all signals collected from the bearing from this point onwards are defective signals.

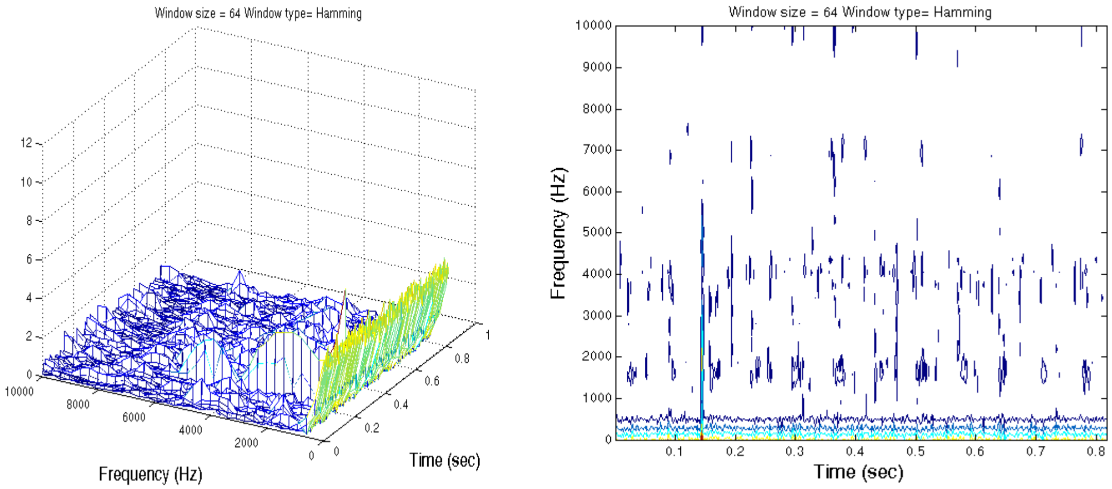

Figure 5-4: (a) STFT of the signal that generated the first peak in the Kurtosis graph of Figure 2. The STFT plot reveals a shock occurring at a specific point in time (left). (b) Further examination of the plot reveals that the shock does not span the entire frequency range with the same amplitude (right). This rules out the interpretation that the shock is irrelevant to the state of the bearing itself (MATLAB).

As a result of the analysis above we can separate the database of signals of bearing 3 into a set of 1734 normal signals collected during the first 31 days of bearing operation, and 422 faulty signals collected in the 4 remaining days of operation before catastrophic failure occurred. Similarly, the database of signals collected from bearing 4 can be separated into 1467 normal signals collected during the first 20 days of operation and 689 remaining faulty signals. Using these two pre-classified databases we create training data sets to train the LAD-based diagnostic tool and testing data sets to test the performance of the tool.

5.6 Implementation

LAD based bearing diagnostics software called cbmLAD was developed to automatically detect early stage bearing defects in order to have enough time to initiate maintenance operations prior to a catastrophic failure. This software was introduced for the first time by Salamanca and Yacout, (2007). In this research it is adapted to deal with the special application of fault diagnostics in bearings. To test the effectiveness of the LAD approach, we attempted to diagnose the state of bearings 3 and 4 in the application explained above as either defective or normal using the signal databases collected for each bearing. To do so we divided the signal database of each bearing in several ways into training data sets and testing data sets.

The database of 2156 signals for bearing 3 was divided in different ways to create 5 different data sets, each composed of a training set and a testing set. The sizes of the sets and their composition are displayed in Table 1. The purpose behind the 5 different sets is to study the effect of different training set sizes on the performance of the diagnostic tool. The selection of the 5 training sets in the data sets was done by selecting every 40th, 50th, 60th, 75th, and 85th signal from the database for each set. This was done to obtain a training set containing signals collected at all the lifetime stages of the bearing. Similarly, the signal database of bearing 4 is divided in 5 different ways into 5 data sets, whose composition is shown in Table 2. It is to be noted here that the sizes of the training data sets range between 1.2% and 2.5% of the total amount of signals in the data sets.

A MATLAB tool was created to pre-process the data using the feature extraction techniques described in Section II of this paper. The tool creates two pre-processed versions of each data set: one contains time-frequency features exclusively and the other contains both time and time-frequency features. The reason for this is to test whether wavelet-based time-frequency features are sufficient on their own for detecting faults.

In addition to the 6 time-domain features described in Section II, the MATLAB tool extracts 12 time-frequency domain features using a discrete wavelet transform with the Daubechies db8 mother wavelet at a scale range of 12. The reason for choosing the Daubechies mother wavelet is the orthogonality requirement of DWT. The scale of 12 was chosen so that the frequency ranges that each energy scale represents are detailed enough to be sensitive to defect frequencies. Figure 5 shows the DWT of signal 2120 of bearing 3 obtained through MATLAB.

As a result, the MATLAB pre-processing tool extracts 2 versions of the training set and testing set from each of the 10 data sets described above,. The first version is composed of the 12 wavelet-based features alone while the second version is composed of 18 features which include the wavelet-based features as well as the 6 time-domain features.

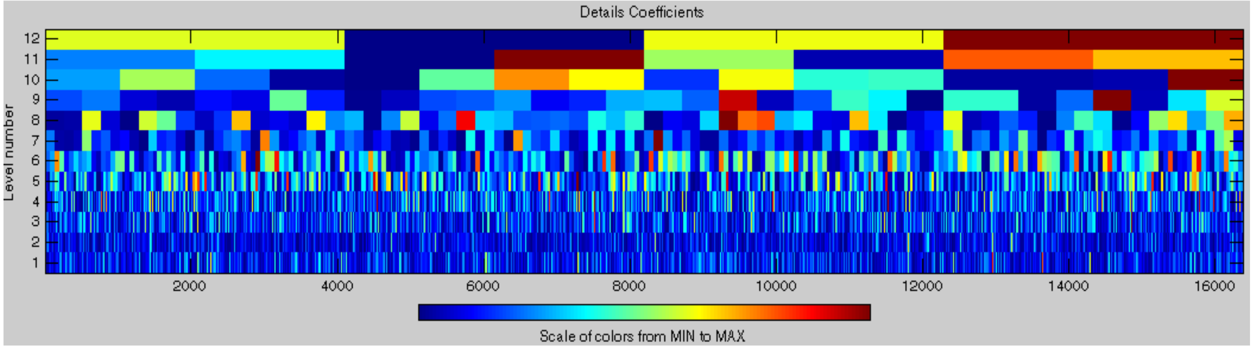

Figure 5-5: The 12 scale DWT of signal 2120 of bearing 3 as obtained using MATLAB. The 12 scales span a frequency range of 20 KHz whereas the horizontal axis is the time scale which spans 16384 data points representing 0.8 seconds.

The bearing diagnostics software, cbmLAD, is coded in C++ language using the free LPSOLVE 5.5 linear programming library [Berkelaar et al. 2004]. The program is first trained using a training data set, after which a decision model is created that can be tested using a testing set. Finally, the parameter k in the pattern generation algorithm, which is the user-defined number that specifies the minimum number of patterns required to cover each observation in the training data set, is also varied between 1 and 150 in order to study the impact of the modifications done on the LAD-based algorithm on the accuracy of the diagnosis.

5.7 Results

Each of the data sets shown in Tables 1 and 2 was used in different ways to train and test the LAD-based diagnostic software. The variables in each case are the type of feature set used and the user-defined parameter k. As mentioned in the previous section, two types of features sets are used on each data set so as to obtain 2 training and testing sets, one including all 18 time and wavelet-based features and one including the 12 wavelet based features exclusively. For each training and testing set the parameter k is varied six times to take the values: 1, 10, 25, 50, 75, and 100.

Table 5-1: 5 data sets are extracted from bearing 3 signal database. Each data set is composed of a training set and a testing set. Above is the size of each set and its composition.

Bearing 3 Data Sets		
Data Set Number	Training set Size	Testing Set Size

1	25	21 Normal 4 Faulty	2131	1713 Normal 418 Faulty
2	28	23 Normal 5 Faulty	2128	1711 Normal 417 Faulty
3	35	28 Normal 7 Faulty	2121	1706 Normal 415 Faulty
4	43	34 Normal 9 Faulty	2113	1700 Normal 413 Faulty
5	53	43 Normal 10 Faulty	2103	1691 Normal 412 Faulty

Table 5-2: 5 data sets are extracted from bearing 4 signal database. Each data set is composed of a training set and a testing set. Above is the size of each set and its composition.

Bearing 4 Data Sets				
Data Set Number	Training set Size		Testing Set Size	
1	30	20 Normal 10 Faulty	2126	1447 Normal 679 Faulty
2	34	23 Normal 11 Faulty	2122	1444 Normal 678 Faulty
3	39	27 Normal 12 Faulty	2117	1440 Normal 677 Faulty
4	47	32 Normal 15 Faulty	2109	1435 Normal 674 Faulty
5	53	36 Normal 17 Faulty	2103	1431 Normal 672 Faulty

To assess the performance of the diagnosis resulting from each trained model we calculate the following statistics: accuracy, true positive rate, true negative rate, false positive rate, false negative rate, and the quality of classification. The term positive refers to the detection of a defect (positive diagnosis) and negative refers to a normally functioning bearing (negative diagnosis). The accuracy measure gives the total number of correct classifications irrespective of the separate accuracy of normal and defective signal detection:

$$Accuracy = (A + B)/N$$

Where A is the total number of correctly diagnosed positive (defective) signals, B is the total number of correctly diagnosed negative (normal) signals, and N is the total number of signals in the testing set. Quality of Classification, on the other hand gives an assessment of the decision model based on the following formula:

$$Quality = \frac{(a+b)}{2} + \frac{(e+f)}{4}$$

Where a is the true positive rate, b is the true negative rate, and e and f are the respective proportions of non-classified positive and negative signals in the testing set.

The best results for the 5 data sets of bearing 3 are shown in Tables 3 and 4. These tables reveal a maximum accuracy ranging between 95.2% and 97.5% for each data set. In 9 out of 10 cases the accuracy levels increased with an increase in k. This demonstrates that the modified LAD-based decision model described in Section III results in increased accuracy 90% of the time. In 6 out of 9 cases the accuracy reached a maximum at a certain k value before decreasing again. This can be explained by the phenomenon that, after an ideal number of generated patterns is reached, all additional patterns generated may be too specific to certain observations in the training set and play a counterproductive role. This suggests that there is an ideal value of k for each data set at which maximum accuracy is attained.

Table 5-3: Best Classification results using the training sets of Table 1 using all features. The table shows training times, the number of patterns created by cbmLAD and some performance statistics.

	Results (Bearing 3) – All Features						
Training Set	1	2	3		4		5
Binary Attributes	76	114	138		146		177
k Factor	125	25	10	25	10	25	25
Training Time (s)	12	1	1	3	1	2	3
Normal Patterns	246	47	16	46	15	40	50
Defective Patterns	125	25	10	26	10	25	25
Accuracy (%)	**93.72**	**95.3**	**97.5**	97.1	96.9	**97.1**	**97.1**
True Negative (%)	98.4	98.4	97.9	97.2	970.	97.4	98.7
True Positive (%)	74.6	82.5	95.7	96.7	96.4	95.6	90.8
False Negative (%)	25.4	17.3	4.3	3.3	3.6	4.4	8.5
False Positive (%)	1.6	1.6	2.1	2.8	3.0	2.6	1.2
Classification Quality	**86.5**	**90.5**	96.9	**97.0**	**96.7**	96.5	**94.9**

Table 5-4: Best Classification results using the training sets of Table 1 using wavelet features. The table shows training times, the number of patterns created by cbmLAD and some performance statistics.

Results (Bearing 3) – Wavelet Features					
Training Set	1	2	3	4	5
Binary Attributes	54	88	101	111	134
k Factor	1	10	10	125	100
Training Time (s)	<1	<1	<1	34	31
Normal Patterns	1	15	19	242	200
Defective Patterns	1	10	12	200	145
Accuracy (%)	**97.1**	**95.2**	**97.2**	**97.2**	**96.0**
True Negative (%)	97.7	98.3	97.8	98.0	96.9
True Positive (%)	94.5	82.5	94.7	93.7	92.5
False Negative (%)	5.5	17.3	5.3	6.3	7.5
False Positive (%)	2.3	1.7	2.2	2.0	3.1
Classification Quality	**96.1**	**90.5**	**96.2**	**95.9**	**94.7**

The training time for the decision models increased with the increase in k and for larger training sets. The longest recorded training time was 57 seconds while the shortest was less than one second. An important statistic for monitoring the performance of the diagnosis is the true positive rate which reflects the ability of the diagnostic tool to detect defective vibration signals. The highest true positive rate achieved was 96.7%, obtained using training set 3 with all 18 features and at $k = 25$. It can be argued that the best classification result was obtained using data sets 3 and 4, from which the highest average classification quality levels per data set were obtained at 97.0% and 96.7%. The best overall decision model was obtained using data set 3 at $k = 10$ using the “all features” feature set type with the highest classification accuracy of 97.5% and a high classification quality of 96.9%.

The best results for the 5 data sets of bearing 4 are shown in Tables 5 and 6. These tables reveal a maximum accuracy ranging between 97.1% and 98.9% for each data set. Similarly to bearing 3 data sets, the accuracy levels increased with an increase in k in 80% of the cases. As with bearing 3, the accuracy reached a maximum at a certain k value before decreasing again in 8 out of the 9 cases where accuracy increased with k .

Table 5-5: Best Classification results using the training sets of Table 2 using all features. The table shows training times, the number of patterns created by cbmLAD and some performance statistics.

Results (Bearing 4) – All Features						
Training Set	1	2	3	4		5
Binary Attributes	113	132	127	174		198
k Factor	25	75	25	75	100	10
Training Time (s)	1	9	2	17	27	1
Normal Patterns	32	112	34	144	194	15
Defective Patterns	25	79	25	75	100	10
Accuracy (%)	**98.9**	**97.1**	**97.5**	**98.3**	**98.3**	**98.8**
True Negative (%)	99.0	99.0	99.9	99.5	99.5	99.0
True Positive (%)	98.7	93.2	92.2	95.7	95.7	100.0
False Negative (%)	1.3	6.8	7.8	4.3	4.3	0.0
False Positive (%)	1.0	1.0	0.1	0.5	0.5	1.0
Classification Quality (%)	**98.8**	**96.1**	**96.0**	**97.6**	**97.6**	**98.7**

Table 5-6: Best Classification results using the training sets of Table 2 using wavelet features. The table shows training times, the number of patterns created by cbmLAD and some performance statistics.

Results (Bearing 4) – Wavelet Features						
Training Set	1	2	3	4		5
Binary Attributes	81	90	87	123		131
k Factor	10	25	1	1	10	1
Training Time (s)	<1	1	<1	<1	1	<1
Normal Patterns	18	30	1	1	16	1
Defective Patterns	12	25	1	1	13	1
Accuracy (%)	**97.7**	**95.2**	**98.7**	97.0	**97.2**	**97.6**
True Negative (%)	98.8	96.8	99.9	96.2	99.3	99.7
True Positive (%)	95.1	91.6	96.2	98.7	92.6	96.3
False Negative (%)	4.9	8.4	3.8	1.3	7.4	0.0
False Positive (%)	1.2	3.2	0.1	3.8	0.7	0.3
Classification Quality (%)	**97.0**	**94.2**	**98.0**	**97.5**	95.9	**97.3**

The training time for the decision models increased with the increase in k and for larger training sets. The longest recorded training time was 67 seconds while shortest was less than one second. The highest true positive rate achieved was 100%, obtained using training set 5. A 100%

classification result means that 100% of defective signals in the test set were correctly identified. The best overall decision model was obtained using data set 1 at $k = 25$ using the "all features" feature set type with the highest classification accuracy of 98.9% and an equally high classification quality of 98.8%.

The overall results obtained from bearing 4 data sets are better than those obtained from bearing 3 in terms of both accuracy and classification quality. This can be explained by the fact that the data sets of bearing 3 contained signals recorded after the catastrophic failure of the bearing had occurred. These signals had been identified as defective signals during the training and testing process. Such signals no longer possess the properties of defective signals and may have, as such, misled the decision model in some minor cases.

For bearing 3 data sets, the difference in maximum accuracy between the decision models trained using all features and the models trained using wavelet-based features alone ranges between 0% and 3.5%; models using all the features had higher accuracy. The average classification quality achieved in each data set by each feature set type ranged between 86.5% and 97.0% when all features were used and between 90.5% and 96.2% when only wavelet features were used.

For bearing 4, the maximum accuracy per data set was obtained in 4 out of 5 cases using the feature set containing all features. The difference in maximum accuracy in these cases between the decision models trained using time and frequency features and the models trained using wavelet based features alone ranges between 1.2% and 1.9%; a narrower range than that found with bearing 3 data sets. The maximum accuracy was higher using the feature set containing wavelet features only in only 1 out of 5 cases. The average classification quality achieved in each data set by each feature set type ranged between 96.0% and 98.8% when all features are used and between 94.2% and 98.0% when only wavelet features were used. For both bearing 3 and 4 data sets, the difference in training time between all features and the wavelet-based features is not significant.

The relatively small difference in accuracy levels and average classification quality levels between those obtained using all features and those obtained using wavelet-based features alone suggests that the use of wavelet energies alone is sufficient to achieve a good diagnosis of bearing status after training. However the use of combined time and wavelet features results in most cases in higher accuracy as the patterns obtained from the decision models suggest. To further illustrate this point we look at the patterns generated by the decision models using data set 5 of bearing 4. Because of the transparency of the LAD-based decision modelling process, we can interpret and compare the patterns generated by the decision model. As an example we take the decision models created from training data set 5 of bearing 3 at $k = 10$ using all features and wavelet features alone. Tables 7 and 8 show the positive (defective) patterns that were found by the two decision models described above. The tables below show the classification power of each positive pattern found, obtained by calculating the percentage of positive observations covered by each positive pattern. The average classification power of the patterns found using all the features in the data set is 89.88% compared to 87.19% for those obtained using wavelet features alone.

Table 5-7: The list of positive (defective) patterns generated by the LAD decision model using bearing 4`s data set 5 at k=10 with wavelet features only.

	Positive (Defective) Patterns Found – Wavelet Features Only									
Feature	1	2	3	4	5	6	7	8	9	10
Wavelet Scale 12										
Wavelet Scale 11									>0.1985	>0.1985
Wavelet Scale 10								>0.221		
Wavelet Scale 9			>0.279	>0.279						
Wavelet Scale 8										
Wavelet Scale 7	>0.650	>0.650	>0.650	>0.650	>0.650	>0.650	>0.650	>0.650	>0.650	>0.650
Wavelet Scale 6					>1.357	>1.357	>1.357	>1.357		
Wavelet Scale 5					>6.873	>6.873			>6.873	>6.873
Wavelet Scale 4	>10.32									
Wavelet Scale 3		>26.56								
Wavelet Scale 2				>47.46		>47.46	>47.46			>47.46
Wavelet Scale 1			>37.44		>37.44			>37.44	>37.44	
Classification Pwr	90.63%	90.92%	84.97%	84.97%	89.58%	89.58%	90.18%	82.29%	84.38%	84.38%

Table 5-8: The list of positive (defective) patterns generated by the LAD decision model using bearing 4`s data set 5 at k=10 with all 18 features.

	Positive (Defective) Patterns Found - All Features									
Feature	1	2	3	4	5	6	7	8	9	10
Wavelet Scale 12										
Wavelet Scale 11						>0.198	>0.198	>0.198	>0.198	>0.198
Wavelet Scale 10										

Wavelet Scale 9						>0.279	>0.279	>0.279	>0.279	>0.279
Wavelet Scale 8										
Wavelet Scale 7						>0.650	>0.650	>0.650		
Wavelet Scale 6										>1.357
Wavelet Scale 5						>6.873	>6.873	>6.873	>6.873	>6.873
Wavelet Scale 4	>10.32									
Wavelet Scale 3		>26.56								
Wavelet Scale 2			>47.46							
Wavelet Scale 1							>37.44			>37.44
Peak Value								>0.585	>0.585	
RMS Value				>0.0011						
Kurtosis										
Skewness										
Std Deviation					>0.089					
Crest Factor						>469.3				
Classification Pwr	95.98%	98.36%	97.47%	97.77%	98.21%	77.23%	80.65%	78.13%	79.02%	95.98%

The patterns shown in Tables 7 and 8 demonstrate the advantage that LAD-based decision models have. The patterns generated by cbmLAD that form the decision rules of the model can be interpreted in terms of the features of the data set. For example, pattern 1 in Table 13 can be interpreted verbally as follows: the presence of an energy level higher than 0.65 at the 7th scale and higher than 10.32 at the 4th scale of the wavelet transform of a vibration signal of a bearing is an indicator that the concerned bearing is defective. This property can be very helpful to technicians as it helps understand the reasons that lead to the diagnosis and facilitates further investigation of the vibration signals of the bearing in light of the decision given by the LAD-based diagnostic tool.

5.8 Conclusion

This paper presented a new approach for automatic diagnosis of faults in rolling element bearings using a modified pattern generation approach based on MILP. The results obtained from tests done on the developed diagnostic software show the LAD-based decision modelling approach is a promising and reliable approach in early stage automatic detection of bearing faults. In addition to obtaining good accuracy and classification quality levels, the LAD-based approach gives decision models whose rules can be easily interpreted in terms of the input features of the model. Such a property is a great asset to technicians on the ground working on analyzing different signal processing tools individually.

The application demonstrated above can be extended to include more input features from both time and time-frequency domains. It can also be easily adapted to different components in rotating machinery on which vibration analysis can be applied. Additionally, the cbmLAD software can be further developed using a multi-class LAD-based decision model so as to identify different fault types in addition to fault diagnosis.

5.9 References

[1] Abbasion, S., Rafsanjani, A., Farshindianfar, A., and Irani, N., (2007), "Rolling element bearings multi-fault classification based on the wavelet denoising and support vector machine", Mechanical Systems and Signal Processing , 21, pp. 2933–2945.

[2] Abramson, S., Alexe, G., Hammer, P., and Kohn, J., (2005), "A computational approach to predicting cell growth on polymeric biomaterials", Wiley InterScience.

[3] Alexe, G., Alexe, S., Axelrod, D., Bonates, T., Lozina, I., and Reiss, M., (2006). "Breast cancer prognosis by combinatorial analysis of gene expression data". Paper obtained from Breast Cancer Research, available at: http://breast-cancer-research.com/content/8/4/R41 (Accessed on July 23, 2009)

[4] Alexe, G., Alexe, S., Axelrod, D., Hammer, P., and Weissmann, D., (2005), "Logical analysis of diffuse large B-cell lymphomas", Artificial Intelligence in Medicine , 34, pp. 235—267.

[5] Alexe, G., Alexe, S., Liotta, L., Petricoin, E., Reiss, M., and Hammer, P.L., (2004), "Ovarian cancer detection by logical analysis of proteomic data", Proteomics, 4,3, pp. 766-783.

Archambault, J., Archambault, R., and Thomas, M., (2002), "Time domain descriptors for rolling-element bearing fault detection", Proceedings of the 20th seminar on machinery vibration, CMVA, Québec, pp. 1-10.

[6] Baillie, D. C., and Mathew, J. (1996), "A Comparison of Autoregressive Modeling Techniques fir Fault Diagnosis of Rolling Element Bearings", Mechanical Systems and Signal Processing, 10,1, pp. 1-17.

[7] Berkelaar , M., Eikland, K., and Notebaert, P., (2004), lp_solve (Open source (Mixed-Integer) Linear Programming system). Available at: http://lpsolve.sourceforge.net/ (Accessed July 10 2009)

[8] Christian, K. N., Mureithi, N., Lakis, A., and Thomas, M., (2007), “On the use of time synchronous averaging, independent component analysis and support vectore machines for bearing fault diagnosis”. First International Conference on Industrial Risk Engineering, Montreal.

[9] Dyer D, and Stewart RM., (1978), “Detection of rolling element bearing damage by statistical vibration analysis”, Trans ASME, Journal of Mechanical Design, 100,2, pp. 229-235.

[10] Jardine, A. K., Lin, D., and Banjevic, D., (2006), “A review on machinery diagnostics and prognostics implementing condition-based maintenance”, Mechanical Systems and Signal Processing , 20, pp. 1483–1510.

[11] Lakis, A.A., (2007), “Rotating Machinery Fault Diagnosis Using Time-Frequency Methods”. 7th WSEAS International Conference on Electric Power Systems, High Voltages, Electric Machines. Venice, Italy.

[12] Lee, J., Qiu, H., Yu, G., Lin, J. and Rexnord Technical Services, (2007). 'Bearing Data Set', IMS, University of Cincinnati. NASA Ames Prognostics Data Repository. Available at: http://ti.arc.nasa.gov/project/prognostic-data-repository, (Accessed on March 12, 2010)

[13] Ma, J., and Li, C. J. (1995), “Detection of Localized Defects in Rolling Element Bearings Via Composite Hypothesis Test”, Mechanical Systems and Signal Processing , 9,1, pp. 63-75.

[14] Martin, H. R., and Honarvar, F., (1995), “Application of Statistical Moments to Bearing Failure Detection”, Applied Acoustics, 44, pp. 67-77.

[15] Mortada, M., Carroll T., and Yacout, S., (2009), “Rogue components: their effect and control using logical analysis of data”, Journal of Intelligent Manufacturing, 1-14.

[16] Qiu, H., Lee, J., and Lin, J., (2006), “Wavelet Filter-based Weak Signature Detection Method and its Application on Roller Bearing Prognostics”, Journal of Sound and Vibration, 289, pp. 1066-1090

[17] Ryoo, H. S., and Jang, I.-Y., (2009), “MILP approach to pattern generation in logical analysis of data”, Discrete Applied Mathematics , 157, pp. 749-761.

[18] Safizadeh, M.S., (1999), “Diagnostic des machines dans le plan temps-frequence”, Ph.D. Thesis Ecole Polytechnique de Montreal, Montreal.

[19] Salamanca, D., and Yacout, S., (2007), “Condition based maintenance with logical analysis of data”, 7e Congrès International de genie industriel, Québec, Canada.

[20] Saxena, A., and Saad, A., (2004), “Fault diagnosis in rotating mechanical systems using self-organizing maps”, Artificial Neural Networks in Engineering, St. Louis.

[21] Shahan, J.E., and Kamperman, G., 1976. (Chapter 8) “Machine element noise, Handbook of industrial noise control”, Industrial Press, New York.

[22] Subrahmanyam, M., and Sujatha, C., (1997), “Using neural networks for the diagnosis of localized defects in ball bearings” Tribology International , 30,10, pp. 739–752.

[23] Sun, W., Chen, J., and Li, J., (2007), “Decision tree and PCA-based fault diagnosis of rotating machinery”, Mechanical Systems and Signal Processing , 21, pp. 1300–1317.

[24] Williams, T., and Ribadeneira, X., (2001), “Rolling Element Bearing Diagnostics in Run-to-Failure Lifetime Testing”, Mechanical Systems and Signal Processing **15** (5), 979–993.

[25] Yacout, S., (2010), “Fault Detection and Diagnosis for Condition Based Maintenance Using Logical Analysis of Data”, **COMADEM International Conference, Nara, Japan.**

[26] Young, R., (1992), “Wavelet Theory and Its Application”, Kluwer Academic Publisher.

Chapter 6 Multi-Class Fault Diagnosis in Power Transformers using Logical Analysis of Data

Mohamad-Ali Mortada, Soumaya Yacout, Aouni Lakis

Submitted to: DISCRETE APPLIED MATHEMATICS
Manuscript Number: DA701

6.1 Abstract

This paper discusses the implementation of a novel multi-class decision model, based on an approach called Logical Analysis of Data (LAD), for the detection and identification of faults in Condition Based Maintenance (CBM). The resulting diagnostic tool is tested on several known machine learning data sets. The results of the test are compared to other classification approaches. To demonstrate its merit in CBM, the approach is tested on the detection and identification of faults in power transformers using Dissolved Gas Analysis (DGA) data. The paper reaches the conclusion that multi-class LAD based fault detection and identification, using a pattern generation technique based on mixed 0-1 integer and linear programming (MILP), is a promising diagnostic approach in CBM.

Keywords: Logical Analysis of Data, Multi-Class Decision Model, Fault Diagnosis, Mixed 0-1 Integer and Linear Programming.

6.2 Introduction

Condition based maintenance is defined as the continuous monitoring of an asset or equipment`s health in a bid to diagnose any faults and fix them before a catastrophic failure occurs. The diagnosis of equipment can be divided into two main parts: fault detection and fault identification. Fault detection classifies equipment as either normal or defective, whereas fault identification classifies the equipment under one of several possible defects that it might suffer from. Different types of classification models have been used for automatic diagnosis of faults in CBM. These models can be divided into two categories depending on their mathematical architecture: statistical approaches and artificial intelligence approaches [1].

A well known statistical approach is Hypothesis testing, implemented in Ma et al. [2] for bearing fault diagnosis. Another approach is Cluster analysis, applied in Staszewski et al. [3] and Jamaludin et al. [4] for fault diagnostics in mechanical systems. Support Vector Machines (SVM), have been used extensively for different diagnostic applications in Christian et al. [5], Abbasion et al. [6], and Widodo et al. [7]. Hidden Markov Models (HMM) is another type of statistical decision model. Ocak et al. [8], Li et al. [9], and Xu el al. [10] used HMM for fault diagnosis in mechanical applications. The common disadvantage of the above classification approaches is their dependence on statistical assumptions. For example, statistical classification models require as a precondition the use of independent and identically distributed (iid) input data. Otherwise, impractical assumptions have to be made that might make the quality of the decision model suffer. Alternatively, data processing techniques such as Independent Component Analysis (ICA) might be used to create new independent data decision variables. This leads to the loss of interpretability of the decision model as the relationship between the diagnostic decision and the input data it is based on is not known [11].

Methods based on artificial intelligence (AI) are reported to have an improved performance over conventional approaches [1]. These methods consist of two: steps training and testing. Training data consist of input observations, typically vectors made up of the values of *features* at certain instances. These *features* are obtained by processing data collected from sensors. AI approaches can be divided into two categories: supervised and unsupervised learning. If the classes of the observations in the data set used to train the model are known, then it is a supervised learning classification approach, otherwise it is an unsupervised learning approach.

The most popular AI approach is artificial neural networks (ANN), which have been used extensively in CBM. Subrahmanyam et al. [12], Spoerre [13], and Yam et al. [14] applied different neural network architectures and training algorithms to diagnostic applications. Unsupervised learning neural network algorithms have been discussed in CBM literature in Saxena et al. [15] and Wu et al. [16]. Many papers discuss combinations of different classification tools in building a fault diagnosis decision model. For example, Yang et al. [17] and Hu et al. [18] discussed the integration of fuzzy logic and expert systems with neural networks, respectively, in machine tools. The disadvantage of neural networks is the black box concept which characterizes its architecture. As such, it is difficult to have a physical explanation of how the trained model, and consequently the classification decision, came to be. Additionally, many ANN algorithms include statistical elements which lead to the same disadvantages as the statistical approaches discussed above.

Logical Analysis of Data (LAD) is a supervised learning classification approach that creates decision rules from Boolean patterns extracted from training data. The advantage of a LAD based decision model is its non-reliance on statistical analysis. As such, no assumptions are required for the type or nature of its input data. This is particularly important in CBM where often, data features are composed of dependent indicators, such as gas content ratios, or are statistical representations extracted from the same indicator data source, such as descriptive statistics extracted from vibration signals. In addition, LAD based decision models are fully interpretable; this means that any diagnostic decision taken by a LAD based decision model can be traced back to the input data features that led to the diagnosis. In more practical terms, LAD based decision models can help maintenance experts understand the patterns that lead to the diagnosis of a certain piece of equipment.

LAD decision models have been implemented in medical applications in Alexe G. et al. [19], Alexe G. et al. [20], Alexe G. et al. [21], and Abramson et al. [22]. Its use in the field of CBM was studied in Salamanca [23], Mortada et al. [24], and Yacout [25]. As LAD is originally a 2-class decision making approach, its use in these papers was limited to fault detection or to the identification of one fault versus another. This paper describes a new multi-class LAD approach that is capable of identifying multiple fault types. It starts by presenting the novel multi-class LAD decision model. The approach is then tested on well known datasets from a machine learning repository. Subsequently, the approach is implemented on a well known CBM application; the identification of faults in power transformers using dissolved gas analysis (DGA) data. Finally a discussion of the results and an evaluation of the contributions of the novel approach are conducted and conclusions are drawn.

6.3 Multi-Class LAD Approach

The main motivation behind using LAD in CBM is, in addition to its good performance, the power to interpret the patterns generated by it. A decision model that can generate patterns that are interpretable by experts is important in the field of equipment health management. For a maintenance expert, such patterns can be powerful tools that give insight on the causes of a certain fault. This knowledge can normally take years to build when relying exclusively on experts' knowledge. In other cases, these patterns can serve to confirm certain long held theories about certain failure causes or trends.

Previous research on the extension of LAD to multi-class applications can be narrowed down to two approaches presented in Moreira [26]. The first approach does not require the alteration of the structure of LAD, as it breaks down the multi-class problem into smaller two-class sub-problems. Each implementation of LAD results in a decision model that separates the observations into two distinct regions. Mayoraz & Moreira [27] , Moreira & Mayoraz [28], and Moreira [26] described different methods that break down a multi-class classification problem (Polychotomy) into two-class problems (Dichotomies). As such this approach could be called a multi-layered 2-class decision model rather than a true multi-class approach. The advantages of the multi-layered 2-class LAD approach are that (1) it permits the use of the conventional LAD methodology on a multi-class problem and (2) it breaks down the classification process into easier to model two-class classification processes which improves the accuracy.

The second approach noted in Moreira [26] for the adaptation of LAD to multi-class applications involves modifying the architecture of LAD. The aim here is to build a single multi-class decision model that classifies all observations into different classes all at once. The advantage of this method over the previous one is that it generates a less complex decision model that has a better execution time. Although it produces less accurate classification than the first approach, the decision rules it generates from Boolean patterns are more intuitive as they relate several classes at the same time.

This paper proposes a novel multi-class LAD approach inspired by the second approach discussed in Moreira [26]. The draw of a true multi-class LAD approach over a multi-layered 2-class one is precisely the intuitive nature of its generated patterns. This is an appealing property for CBM as the boundaries between different fault types are often not clear cut as will be demonstrated in the case of power transformer fault diagnosis. The new approach proposed in this paper introduces several new features that improve the accuracy of the decision model as will be detailed later.

As with conventional LAD, the multi-class LAD decision making approach proposed here is composed of 3 steps: Data Binarization, Pattern Generation, and Theory Formation. In what follows, we discuss in details each of those steps.

6.3.1 Data Binarization

The Data Binarization step involves the translation of data used to train the LAD decision model to binary data using a binarization technique that translates each numerical feature to a set of binary *attributes*. The technique used in this paper ranks the distinct values that a numerical feature u takes in a data set in ascending order. Then a cut-point α is inserted between each two values that belong to different classes. The cut-point is calculated as the average of the two values. A binary attribute b is then formed from each cut-point such that:

$$b(u) = \begin{cases} 1 & if \quad u \geq \alpha \\ 0 & if \quad u < \alpha \end{cases}$$

The number of binary attributes that make up the binarized training set at the end of the binarization process depends on the number of cut-points generated for each numerical feature.

6.3.2 Pattern Generation

The pattern generation step generates a set of Boolean rules called patterns that are distinct to the observations that belong to one class and not the other. A pattern $\boldsymbol{p}$ is composed of a conjunction of *literals*; a literal is a Boolean variable x or its negation $\bar{x}$ [29]. Each binary attribute b_i in the training set can be represented in a pattern by a literal x_i or its negation $\bar{x}_i$, where x_i is used for $b_i = 1$ and $\bar{x}_i$ for $b_i = 0$. In its strictest sense, a pattern $\boldsymbol{p}$ of degree δ is a conjunction of δ literals such that it is true for at least one observation of one class and not true for all the observations of the other class. A pattern that is true for a certain observation is said to *cover* that particular observation. Less strict definitions of pattern $\boldsymbol{p}$ allow for a large percentage of coverage for one class and a much smaller coverage for the other class.

The approach for pattern generation proposed in this paper is based on the formulation and solution of a mixed 0-1 integer and linear programming (MILP) problem. This approach offers two advantages over the one described in Moreira [26]. The first advantage is the ability to control the discriminating power between the different classes through a user defined parameter called the *discriminating factor*, l. The discriminating factor between a pair of classes is defined as the minimum number of patterns that must separate each observation belonging to class c_i from those belonging to class c_j. It can be deduced from Boros et al. [29] that the higher the number of patterns that separate one class of observations from another, the higher the discriminating power of the decision model that emerges from these patterns. As a result, a higher discriminating power between two classes is synonymous to higher classification power [29]. The second advantage of the proposed approach is that it sets no limit on the degree of the patterns generated by the pattern generation algorithm. This creates more possibilities for finding patterns, which in turn translates to more chances for generating useful knowledge. In what follows is a description of the proposed pattern generation step.

The pattern generation step requires a procedure that generates a pattern from the binarized training set, hence the single pattern generation algorithm, based on solving an MILP problem, which will be named *patt_gen*. In addition, it requires an iterative mechanism that calls and loops the *patt_gen* algorithm and ensures that enough patterns are generated to create a good decision model. This mechanism, which will be called the *Multi_Class_LAD* procedure, will be discussed first in the following subsection. The MILP *patt_gen* algorithm will be explained afterwards.

Multi_Class_LAD Procedure

The aim of this procedure is to build a shared set of multi-class patterns that can be used to create the decision model in the theory formation step of the LAD approach. This is a variation from the conventional structure of LAD which generates an entire set of patterns for a single dichotomy of two classes.

The *Multi_Class_LAD* procedure starts by creating an empty set of patterns P_{ij} for each pair of classes (c_i, c_j), where $i, j \in \{1,2,\ldots,K\}$ such that $i \neq j$, and K is the total number of classes. It should be noted here that any two sets P_{ij} and P_{ji} are not identical$(P_{ij} \neq P_{ji})$. For K number of classes, a total of $K(K-1)$ sets of patterns will be created. At the end of the *Multi_Class_LAD* procedure, a set P_{ij} may contain two types of patterns: those defined by the strict definition of a pattern which states that it should cover at least one observation belonging to class c_i but none in class c_j, and those that are less strictly defined, which require a certain percentage of observations in c_i to be covered but also allow for a small percentage of observations belonging to class c_j to be covered as well. These patterns are generated through multiple solutions of the single pattern generation MILP algorithm *patt_gen* which will be discussed in the following subsection.

The relationship between the pattern sets and the class pairs to which they belong is governed by the discriminating matrix $\mathbf{R}_l(K \times K)$ whose elements r_{ij} are calculated as follows:

$$r_{ij} = \frac{\text{disc}_l(c_i, c_j)}{m_i} \tag{1}$$

As mentioned above, the discriminating factor $l \geq 1$ represents the minimum number of patterns in set P_{ij} required to separate each observation belonging to class c_i from those in class c_j. The numerator $\text{disc}_l(c_i, c_j)$ refers to the number of observations in class c_i that have been covered at least l times by the patterns found in the set P_{ij}. The denominator m_i is the total number of observations that belong to class c_i. Initially, the sets P_{ij} are empty for every i and j such that $i \neq j$. Consequently, the matrix $\mathbf{R}_l$ is initialized as follows:

$$\mathbf{R}_l = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

The *Multi_Class_LAD* procedure generates patterns in an iterative manner by calling the MILP based single pattern algorithm *patt_gen* on each run. With the generation of a new pattern, the content of the pattern sets changes, and the elements of $\mathbf{R}_l$ are consequently updated. The *Multi_Class_LAD* procedure generates as many patterns as necessary until all elements of R are equal to 1 or reach a minimum value $r_{\min}$ specified by the user.

Upon each run, the MILP based single pattern generation algorithm, *patt_gen*, generates a pattern from the observations of the class pair that has the least discriminating rate, i.e. the class pair with the lowest value for r_{ij}. For a class pair (c_i, c_j) the observations belonging to c_i that have not yet been covered by l patterns are put in a set $S_{\text{uncovered},\, i}$ whereas all observations belonging to set c_j are placed in a set S_j. The *patt_gen* algorithm is then called to generate a pattern that covers observations in set $S_{\text{uncovered},\, i}$ but not those in set S_j. The generated pattern, $\boldsymbol{p}$, is then placed in the set of patterns P_{ij} corresponding to the pair of classes (c_i, c_j). In cases where a pattern can no longer be generated the corresponding value r_{ij} is set to $r_{\min}$ and the procedure continues.

The generated pattern $\boldsymbol{p}$ is then tested for all class pairs (c_a, c_b) where $a, b \neq i, j$, in order to decide on whether $\boldsymbol{p}$ can be placed in their corresponding pattern set P_{ab}. The rules for a pattern to be admitted to a set P_{ab} are governed by the two user defined values: the minimum positive coverage rate $d_{\min+}$ and the maximum negative coverage rate $d_{\max-}$. Here and throughout this section, "positive" refers to the class for which a pattern is being generated and "negative" for the opposite class. The generated pattern is added to each set of patterns P_{ab} that satisfies the following coverage rate thresholds:

$$d_a(\boldsymbol{p}) = \frac{\operatorname{cov}(\boldsymbol{p}, c_a)}{m_a} \geq d_{\min+} \qquad (2)$$

$$d_b(\boldsymbol{p}) = \frac{\operatorname{cov}(\boldsymbol{p}, c_b)}{m_b} \leq d_{\max-} \qquad (3)$$

The term $\operatorname{cov}(\boldsymbol{p}, c)$ refers to the number of observations belonging to a class c that are covered by pattern $\boldsymbol{p}$. The denominator m is the total number of observations that belong to class c. Consequently, a generated pattern $\boldsymbol{p}$ is placed in the set P_{ab} if its coverage rate is above $d_{\min+}$ for class c_a and below $d_{\max-}$ for class c_b.

With the content of the pattern sets updated, the elements of the discriminating matrix are consequently updated using equation (1). If the elements of the matrix are all above the user defined minimum discriminating rate $r_{\min}$, then the *Multi_Class_LAD* procedure stops. Otherwise, the class pair with the least discriminating rate is chosen and the procedure described above is repeated.

The output of the multi-class LAD procedure is $K(K-1)$ sets of patterns P_{ij}. These sets may have patterns in common as the purpose of this procedure is to obtain multi-class patterns. The pseudocode for this procedure is shown in Figure 1.

```
**Input** : Binarized Training Data Set $S$
```

```
                 $l$           (Discriminating Factor)
                 $r_{\min}$     (Minimum Discriminating Rate)
Parameters :     $d_{\min+}$    (Minimum  Positive Coverage Rate)
                 $d_{\max-}$    (Maximum  Negative Coverage Rate)
$r_{ij} \leftarrow \begin{cases} 0 & if \quad i \neq j \\ 1 & if \quad i = j \end{cases} \quad \forall i, j \in \{1,2,\ldots,K\}$   (Elements of the Discriminating Matrix $\mathbf{R}$)
$P_{ij} = \phi \quad \forall i, j \in \{1,2,\ldots,K\} \quad i \neq j$   (Pattern Set for each class pair $(c_i, c_j)$)
while exists an $r_{ij} < r_{\min}$ do
        $i, j \leftarrow \operatorname{argmin}_{i,j}(r_{ij})$   (Find the class pair with the least $r$ in $\mathbf{R}$)
        $S_{\text{uncovered},i} \leftarrow$ All Observations in class $c_i$ not covered by $l$ patterns in $P_{ij}$
        $S_j \leftarrow$ All Observations in class $c_j$
        $\boldsymbol{p} \leftarrow patt\_gen(S_{\text{uncovered},i}, S_j)$   (Generate a Single Pattern using the $patt\_gen$ algorithm)
        if $\boldsymbol{p}$ exists
                $P_{ij} \leftarrow P_{ij} \cup \boldsymbol{p}$   (Add Pattern to Set)
                for all $(c_a, c_b)$ where $a \neq b$ and $a, b \neq i, j$
                        if $\operatorname{cov}(\boldsymbol{p}, c_a)/m_a \geq d_{\min+} \quad AND \quad \operatorname{cov}(\boldsymbol{p}, c_b)/m_b \leq d_{\max-}$
                                $P_{ab} \leftarrow P_{ab} \cup \boldsymbol{p}$   (Add Pattern to Set)
                                Update $r_{ab}$
                        end if
                end for
        else $r_{ij} \leftarrow r_{\min}$
end while
Return $P_{ij} \quad \forall i, j \in \{1,2,\ldots,K\} \quad i \neq j$   (Return All Pattern Class Pair Sets)
```

Figure 6-1: Multi_Class_LAD Procedure

Generating a Single Pattern

As mentioned above, the *Multi_Class_LAD* procedure calls upon an MILP based single pattern generation algorithm called *patt_gen* to generate a pattern upon each run. In Moreira [26] a heuristic algorithm was used to find a pattern based on preset constraints, namely the maximum pattern degree $\delta_{\max}$, the minimum positive coverage rate $d_{\min+}$ and maximum negative coverage rate $d_{\max-}$. The disadvantage of this procedure is that there is no way of knowing the optimal values that need to be assigned to these constraints in order to obtain the most optimal patterns that will lead to a good decision model. The algorithm proposed in this paper is inspired from Ryoo and Jang [30]. The algorithm presented in Ryoo and Jang [30] is based on MILP and was shown to have short training times [31]. In addition to that, it is a malleable algorithm that can be easily modified to generate different types of patterns. The algorithm presented in this paper adds a key adjustment to the original algorithm of Ryoo and Jang [30] in order to accommodate the requirement that more than one pattern should cover each observation due to the presence of the discriminating factor $l \geq 1$ introduced in the previous sub-section. The MILP based pattern generation algorithm described here puts no limit on the maximum degree that a pattern should have. The algorithm instead generates a pattern that is optimal with respect to the degree and the coverage rate. The resulting pattern has the least possible degree and the highest positive coverage rate in order to increase its generalization power. We illustrate this algorithm below.

The single pattern generation algorithm involves finding the optimal values for a set of decision variables that minimize a certain objective function subject to a set of constraints. The constraints impose restrictions on the values that these decision variables can take. The decision variables involved in this algorithm are: the pattern degree δ, the Boolean pattern vector $\mathbf{w}$ describing the composition of the pattern found, and the coverage vector $\mathbf{y}$.

The Boolean *pattern vector* $\mathbf{w}(w_1, w_2, \ldots, w_{2q})$ has a size n that is double the number of binary attributes q that make up the binarized training data set, i.e. $n = 2q$. The elements $w_1, w_2, \ldots, w_q$ of $\mathbf{w}$ are such that if $w_j = 1$ then the literal x_j is included in pattern $\boldsymbol{p}$. Similarly, the elements $w_{q+1}, w_{q+2}, \ldots, w_{2q}$ are such that if $w_{q+j} = 1$ then literal $\overline{x}_j$ is included in pattern $\boldsymbol{p}$. Naturally, a pattern cannot include both the literal x_j and its negation $\bar{x}_j$ at the same time, hence the constraint:

$$w_j + w_{q+j} \leq 1 \quad j = 1,2,\ldots,q \tag{4}$$

Note that when both $w_j = 0$ and $w_{q+j} = 0$, then the binary attribute b_j is not represented in the pattern. $\mathbf{y}$ is a Boolean vector whose number of elements equals the number of positive observations in the set $S_{\text{uncovered},\,i}$ obtained in sub-section 2.2.1, and which will be called S^+ here. As such, the binarized data used to generate a pattern is divided into two subsets: S^+ includes all the observations of class c_i that have not yet been covered by at least l patterns and S^- includes all the observations of class c_j. The elements y_i of vector $\mathbf{y}$ are the variables to minimize in the set covering problem such that $y_i = 0$ when observation $i \in S^+$ is covered by pattern $\boldsymbol{p}$ and 1 otherwise.

The solution to the MILP problem is a pattern $\boldsymbol{p}$ of degree δ. The composition of the pattern can be deduced from vector $\mathbf{w}$ and its coverage from vector $\mathbf{y}$. The optimal solution for the above variables is found by minimizing the following objective function:

$$\min_{\mathbf{w},\mathbf{y},\delta} \sum_{i \in S^+} y_i + g\delta$$

where g is a user provided parameter. If $g > 0$, then the above objective function maximizes the number of observations covered by the generated pattern $\boldsymbol{p}$ in the set S^+ while simultaneously minimizing the degree of the pattern by penalizing the objective function with $g\delta$. As such the generated pattern will be optimal with respect to degree and coverage. Patterns that are optimal with respect to degree and coverage have been proven to reduce the number of unclassified observations [32]. Moreover, such patterns provide high explanatory power because of their small degree and the simplicity of the decision model built by them [33]. As such, they are more likely to cover observations belonging to several classes. This is a desired property for the multi-class LAD approach discussed here as the patterns generated using the *patt_gen* algorithm are tested for their coverage of other classes as seen in the *Multi_Class_LAD* procedure. The generalization power of these patterns leads to an accelerated training time for the decision model as the generated patterns are likely to separate more class pairs. However, the degree optimality insured by the addend $g\delta$ at $g > 0$ comes at the expense of having to search for the optimal value of g. Moreover the MILP problem risks having a feasible solution that is not a pattern at all; hence the step in the *Multi_Class_LAD* procedure that verifies whether the output of the *patt_gen* algorithm is indeed a pattern. These disadvantages can be avoided if g is set to zero, as the output of the MILP problem is then guaranteed to be a coverage optimal pattern as proven in Ryoo and Jang [30] and Mortada et al. [31]. The disadvantage in that case is that the pattern generated by the *patt_gen* algorithm is not degree optimal. Such patterns are more conservative and will consequently slow the training process of the decision model and have a lower explanatory power.

The above objective function must be subject to constraints in order for the resulting MILP problem to generate a degree and coverage optimal pattern. Three major sets of constraints need to be satisfied for generating a pattern from the optimization of the objective function:

(1) The resulting pattern must be able to cover a positive observation $i \in S^+$, however it is not required to cover all the observations in S^+. This condition can be described as:

$$\sum_{j=1}^{2q} a_{ij} w_j + q y_i \geq \delta \quad \forall i \in S^+ \tag{5}$$

Where a_{ij} are the elements of a vector $\mathbf{a_i}$ associated with observation $i \in S^+$. The Boolean observation vector $\mathbf{a_i}\left(a_{i,1}, a_{i,2}, \ldots, a_{i,q}, \ldots a_{i,2q}\right)$ is such that $a_{ij} = 1 \quad (j = 1,2,\ldots,q)$ if the value of attribute b_j in i is 1, and $a_{i(j+q)} = 1 \quad (j = 1,2,\ldots,q)$ if $b_j = 0$ in i. a_{ij} and $a_{i(j+q)}$ are mutually exclusive since b_j cannot be 1 and 0 at the same time for the same observation.

(2) A positive pattern should not cover any negative observations. For this reason the dot product of the pattern vector $\mathbf{w}$ and the observation vector $\mathbf{a_i}$ of each negative observation $i \in S^-$ must be less than the degree δ of $\boldsymbol{p}$:

$$\sum_{j=1}^{2q} a_{i,j} w_j \leq \delta - 1 \quad \forall i \in S^- \tag{6}$$

(3) The MILP problem must not generate a pattern that has been generated in previous runs. Additionally, in order to increase the diversity of the patterns, the newly-generated pattern must not be a subset of any of the patterns that have already been generated. The set of constraints that prevents this from happening is:

$$\sum_{j=1}^{2q} v_{k,j} w_j \leq \delta_k - 1 \quad \forall \mathbf{v}_k \in \mathbf{V}_{ij} \tag{7}$$

Where $\mathbf{V}_{ij}$ is the set containing the Boolean pattern vectors of all the generated patterns in the set P_{ij}. Initially, the set $\mathbf{V}_{ij}$ is empty and this constraint is not considered. However, with the progress of the pattern generation procedure, a pattern $\boldsymbol{p}_k$ of degree δ_k and Boolean pattern vector $\mathbf{w}_k$ is added to the set P_{ij}. The vector $\mathbf{w}_k$ obtained for that pattern is renamed as vector $\mathbf{v}_k$ and added to the set $\mathbf{V}_{ij}$. As such, a new constraint is added for each pattern already found in the set P_{ij}.

The resulting MILP problem can be solved using linear programming software. In this paper, a C++ linear programming library called LPSOLVE 5.5 [34] was used for this purpose. The MILP problem is shown below:

$$\min_{\mathbf{w},\mathbf{y},\delta} \sum_{i \in S^+} y_i + g\delta$$

$$s.t. \quad \begin{cases} (4),(5),(6),(7) \\ \sum_{j=1}^{2q} w_j = \delta \quad (8) \\ 1 \leq \delta \leq q \quad (9) \\ \mathbf{w} \in \{0,1\}^{2q} \quad (10) \\ \mathbf{y} \in \{0,1\}^{m} \quad (11) \end{cases}$$

where m is the number of observations in S^+. The constraints in equations (8) to (11) can be justified intuitively. A solution of the above MILP problem generates a single pattern from the observations of class pair c_i and c_j. The generated pattern is placed in set P_{ij} and then tested for all class pairs and added to the sets P_{ab} for every $a,b \in \{1,2,\ldots,K\}$, where $i,j \neq a,b$, subject to the conditions defined by $d_{\min +}$ and $d_{\max -}$ discussed in the *Multi_Class_LAD* procedure. The set of constraints in (7) and the varying content of the sets S^+ and S^- insure that a new pattern is generated upon solution of the above problem.

6.3.3 Theory Formation

The final step in the formation of a LAD decision model is the theory formation step where the patterns generated in the previous step are used to create a decision function called the discriminant. For a classical two-class LAD decision model, where one class is labelled positive and the other negative, the output of a discriminant function is a value that is positive when the tested observation belongs to the positive class and negative otherwise. A value of zero means no classification is possible. However, the discriminant function used in the multi-class LAD approach described here is significantly different. We describe the new function below.

As there are more than two classes in a multi-class LAD decision model, the sign of a single output of a discriminant is no longer sufficient to classify an observation. Instead, the discriminant used here creates a score for each class. An observation therefore belongs to the class with the highest score. In order to calculate these scores we first create a pattern/class relationship matrix $\mathbf{D}_{ij}(N_{ij} \times 1)$ for each class pair pattern set P_{ij}, where N_{ij} is the number of patterns in the set P_{ij}. Each element $d_{ij,n}$, where $n \in \{1,2,\ldots,N_{ij}\}$, of the matrix is calculated as the coverage rate of the pattern $\boldsymbol{p}_n$ in P_{ij} with respect to the observations of class c_i, normalized by the sum of coverage rates of all the patterns in set P_{ij}:

$$d_{ij,n} = \frac{\operatorname{cov}(\boldsymbol{p}_n, c_i)/m_i}{\sum_{n=1}^{N_{ij}} \left(\operatorname{cov}(\boldsymbol{p}_n, c_i)/m_i\right)} \qquad (12)$$

As a result, every pattern $\boldsymbol{p}_n$ in a set P_{ij} is associated with an element $d_{ij,n}$ of the matrix $\mathbf{D}_{ij}$. These elements act as normalized weights for the patterns. In order to calculate the score for a certain class $k \in \{1,2,\ldots,K\}$, we group all the class pair pattern sets that separate class k and all the remaining $(K-1)$ classes. As such, for each class k we group $(K-1)$ class pair sets: $P_{kj} \quad \forall j \in \{1,2,\ldots,K\}$ where $j \neq k$. For a new observation O, the score for a class $k \in \{1,2,\ldots,K\}$ is calculated by adding the elements $d_{kj,n}$ of matrix $\mathbf{D}_{kj}$ whose corresponding pattern $\boldsymbol{p}_n$ in set P_{kj} covers observation O, for all $(K-1)$ class pair sets. The maximum score obtained for one class is equal to $(K-1)$, which is obtained when all the patterns in the sets P_{kj} $\forall j \in \{1,2,\ldots,K\}$, where $j \neq k$, cover the observation O. The resulting discriminant function therefore takes the following shape:

$$\hat{F}(O) = \underset{k=1,\ldots,K}{\arg\max} \sum_{\substack{j\in\{1,2,\ldots,K\} \\ j\neq k}} \sum_{p_n \in P_{kj}} \boldsymbol{p}_n(O) \cdot d_{kj,n} \tag{13}$$

A pattern $\boldsymbol{p}_n(O) = 1$ if it covers observation O and zero otherwise. The output of the above decision function for a new unclassified observation O is the highest scoring class for that observation.

6.4 Experiments

The above multi-class LAD approach was implemented in C++ as the multi-class component of the CBM software cbmLAD. In order to examine its performance, we tested it on 3 well studied data sets, shown in table 1, obtained from a machine learning data repository [35]. The experiment was conducted by using 50% of the data sets for training the LAD decision model and the rest for testing it. The training data sets were binarized using the technique described in section 2. To demonstrate the ability of the proposed multi-class LAD approach to handle large data sets, the experiment was first conducted without any set reduction mechanisms. The results were then compared with those obtained using data sets reduced by a generalized simple greedy set reduction algorithm proposed in Almuallim and Dietterich [36]. To test the impact of the discriminating factor l on the performance of the resulting decision model, we trained the LAD decision model at three values of l (1, 5, and 10) and compared the results of the three resulting decision models. Finally, in order to insure a fair assessment, we repeated the experiment 30 times on each of the 3 data sets, each time on new, randomly and independently generated, training and testing sets.

Table 6-1: Data Sets obtained from the machine learning repository [35].

Data Set	Number of		
	Classes	Attributes	Observations
Wine	3	13	178
Glass	6	10	214

Ecoli	8	8	336

The results of the experiments described above are shown in tables 2-4. Table 2 shows the arithmetic mean and the standard deviation of the number of patterns generated from the 30 experiments using the non-reduced and reduced training data and by varying the discriminating factor between 1, 5, and 10. Table 3 shows the arithmetic mean of the time (in seconds) it took to train the LAD decision model using the two different data set configurations and the three discriminating factor values. Table 4 shows the arithmetic mean and standard deviation of the classification accuracy as well as the percentage of unclassified observations obtained in the 30 trained LAD decision models. The best accuracy for each experiment is shown in bold.

Table 6-2: Number of Attributes and Number of Patterns Generated.

Data Set	Attributes	Number of Patterns			Attributes	Number of Patterns		
	NonReduced	$l = 1$	$l = 5$	$l = 10$	Reduced	$l = 1$	$l = 5$	$l = 10$
Wine	445.4 ±14.3	6.1±0.8	31.5±3.1	62.5±4.9	4.3±0.7	7.2±2.0	8.9±3.2	8.9±3.0
Glass	416.57±10.5	25.0±2.5	120.5±8.6	236.5±17.4	11.1±09	39.6±4.6	120.5±22.3	156.1±40.7
Ecoli	240.6±4.8	32.5±2.9	151.9±15.5	304.7±30.7	13.9±1.4	39.6±4.1	87.2±25.7	89.7±32.8

Table 6-3: Training Time in seconds of the Decision Models using an ordinary computer.

Data Set	Training Time using Non-Reduced Sets			Training Time using Reduced Sets		
	$l = 1$	$l = 5$	$l = 10$	$l = 1$	$l = 5$	$l = 10$
Wine	2s	9s	23s	<1s	<1s	<1s
Glass	494s	1436s	2928s	1s	16s	39s
Ecoli	171s	985s	1838s	1s	19s	49s

Table 6-4: Accuracy Levels Achieved using the Multi-Class LAD decision models.

Data Set		Performance of Non-Reduced Sets at			Performance of Reduced Sets at		
		$l = 1$	$l = 5$	$l = 10$	$l = 1$	$l = 5$	$l = 10$
Wine	Accuracy (%)	90.1±4.3	92.4±3.2	**93.1±3.2**	85.8±1.6	87.8±4.2	87.8±4.2
	Unclassified (%)	1.3±1.9	0.0±0.0	0.0±0.0	1.9±3.8	2.3±4.1	2.3±4.1
Glass	Accuracy (%)	62.1±5.0	64.6±5.0	**65.0±5.4**	54.9±5.0	54.8±5.8	54.2±6.0
	Unclassified (%)	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
Ecoli	Accuracy (%)	76.8±4.1	**79.2±4.2**	78.9±3.2	71.4±4.7	69.3±6.4	67.1±6.5
	Unclassified (%)	0.04±0.2	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0

The results show that a higher discriminating factor results in an increase in the number of patterns generated. In the 3 data sets a higher accuracy level was achieved at $l > 1$ when the non-reduced binarized data sets were used. The reduced data sets produce less accurate decision models and the increase in discriminating factor does not have a positive effect on the accuracy in 2 of the 3 data sets. The loss in accuracy in comparison to the non-reduced data sets can be justified by the loss in information that occurs when the binarized training data sets are reduced. The simple greedy technique used to reduce the number of binary attributes results in severe data loss as it achieves up to 99% set reduction as in the case of the wine data sets. A more intelligent set reduction algorithm is required to achieve better accuracy results. However, as the main purpose of this paper is to highlight the performance of the proposed multi-class LAD approach and its pattern generation algorithm, the use of the set reduction algorithm was simply to demonstrate the fast training times that can be achieved using the procedures described in section 2. Indeed, the training times achieved using the reduced data sets are significantly lower than the ones using the non-reduced data sets, and compare favourably with other classification techniques in the literature.

A comparison of the classification accuracy obtained in this paper with those reported in Moreira [26] is shown in Table 5. The results obtained here are compared to those of four other approaches: Moreira, CN2, TABATA, and C4.5. The Moreira approach is the multi-class LAD approach studied in Moreira [2000]. CN2 is a popular algorithm based on sequential covering rule induction. TABATA is another sequential covering rule induction algorithm that uses the Tabu search approach for generating patterns. C4.5 is a decision tree learning algorithm [37] that is capable of solving multi-class problems without the need for parameter tuning. This algorithm is widely cited in the literature for its balancing of short execution time and good classification accuracy in different applications [38]. The comparison in Table 5 reveals that the proposed approach improves on the accuracy levels of the model described by Moreira [26] in all three data sets. Moreover, the proposed approach gives the best accuracy among all four other approaches for the wine data set. For the Glass and Ecoli data sets, the proposed approach results in the second best accuracy. Out of all the methods shown in Table 5, the proposed approach has the best overall average accuracy for the 3 data sets. The number of patterns generated from these models is significantly higher than that obtained using the other classification techniques.

Table 6-5: Comparison of Classification Accuracy with the results stated in Moreira [26].

Data Set	Best Classification Accuracy	Achieved at	Comparison with [Moreira 2000]			
			Moreira	CN2	TABATA	C4.5
Wine	**93.1±3.2**	$l = 10$	92.7±2.54	90.2±2.7	70.6±10.4	89.9±3.1
Glass	**65.0±5.4**	$l = 10$	62.4±5.9	**65.5±5.3**	58.6±5.6	62.8±4.4
Ecoli	**79.2±4.2**	$l = 5$	78.3±3.4	77.0±3.4	42.7±0.4	**80.6±4.1**

6.5 Implementation in CBM

The multi-class LAD decision making approach introduced in section 2 was implemented on a fault diagnosis CBM application: the detection and identification of faults in power transformers using dissolved gas analysis (DGA) data. Power transformers are high-priced electromechanical equipment that is monitored continuously in order to detect faults in operation prior to the occurrence of potentially costly and hazardous failures. DGA data has been discovered as an important indicator of the state of a power transformer [39]. DGA data has been analyzed extensively over the decades and many experts devised standardized rules and patterns that relate gas indicator values with fault types using expert knowledge [39, 40]. The advantage of using LAD in this application is that the decision model automatically generates interpretable patterns that can help maintenance experts and manufacturers understand the causes of a certain type of failure.

To demonstrate the potential contributions of the proposed approach to this type of application, we implement it on a data set of historical samples of 5 characteristic gases obtained from a 500kV transformer located in a substation of the South China Electric Power Company [41]. From a total of 75 samples in the database, 9 are reported to be normal, 38 suffer from thermal heating, 21 are faulty due to high energy discharge, and 7 due to low energy discharge [41]. The characteristic gases are processed using the formulas in Table 6 to get a data set composed of 6 numerical features: the relative content of the gases H_2, CH_4, C_2H_6, C_2H_4, and C_2H_2 respectively, as well as the logarithm of the highest gas content in the sample. The data set is split into training and testing sets that are identical to those presented in Lv et al. [41]. The multi-class LAD decision making approach discussed in section 2 is then implemented while varying the discriminating factor l between 1 and 10. The best accuracy achieved was 84% at $l = 2$. This is in contrast to the 100% accuracy level declared in Lv et al. [41] using support vector machines. Training time for the decision model took less than 1 second on an ordinary computer.

Table 6-6: Power Transformer Data Set Feature[12] [41]

$y_1 = \dfrac{c_1}{\underset{i=1\to 5}{\max}(c_i)}$	$y_2 = \dfrac{c_2}{\underset{i=1\to 5}{\max}(c_i)}$	$y_3 = \dfrac{c_3}{\underset{i=1\to 5}{\max}(c_i)}$

$y_4 = \dfrac{c_4}{\max\limits_{i=1\to5}(c_i)}$	$y_5 = \dfrac{c_5}{\max\limits_{i=1\to5}(c_i)}$	$y_6 = \log_{10}\left(\max\limits_{i=1\to5} c_i\right)$

[1] c_i is the content of one of the 5 respective characteristic gases: H_2, CH_4, C_2H_6, C_2H_4, and C_2H_2.

[2] y_i is one of the 6 numerical features extracted from the DGA data.

In comparing the results achieved using this method to that obtained using support vector machines [41], it is obvious that multi-class LAD does not improve on the accuracy. However, the true merit of using multi-class LAD is in the interpretability of the generated patterns. These patterns can be translated into a set of rules similar to those obtained using expert knowledge. The 18 patterns obtained using the multi-class LAD decision model at $l = 2$ are translated to the rules shown in Table 7. Each rule may belong to more than one class where class 1 refers to a fault with High Energy discharge, class 2 to a fault with Low Energy discharge, class 3 to a fault due to Thermal Heating, and class 4 to a normal state.

Table 6-7: Rules Generated By LAD patterns.

Rule	y_1	y_2	y_3	y_4	y_5	y_6	Class Type
1					>0.037	>1.577	1,3
2					>0.163	>1.577	1
3					>0.163		1,2,4
4					<0.163		2,3,4
5					<0.037		2,3,4
6				<0.697		<3.353	1,2,4
7				<0.697		>1.147	1,2
8				<0.862		>1.147	1,2,3
9			>0.053			>2.283	1,3
10					<0.497	<1.727	1,4
11					<0.568	<1.727	1,4
12			>0.053		<0.035		3,4
13						>1.727	1,2,3
14				<0.107			1,2,4
15						<1.531	2,4
16						<1.577	2,4
17				>0.614		>2.123	3
18			<0.001				2

The table above reveals rules for fault detection and identification that were generated without reliance on expert knowledge. These rules can be used by maintenance experts for verification purposes as well as to study the root causes of certain faults. For example, Rule 13 in the table above covers all 3 fault types but not the normal state which means that we have found a unique pattern that covers the 3 faulty classes and separates them from the normal class. This rule can be interpreted as follows: when the logarithm of the absolute content of one of the five characteristic gases is higher than 1.727, then this is an indicator that a fault has occurred. Similarly, rule 17 reveals that a fault due to thermal heating occurs when the relative content of C_2H_4 is greater than 0.614 and the absolute gas content measure is greater than 2.123. The multi-class nature of the rules generated from the patterns validates the argument that multi-class patterns offer a more intuitive and natural relationship between the classes and the monitored indicator data. An example is rules 8 and 13 which relate the monitored DGA indicators to all failure classes and rule 7 which relates the indicator data to the Energy type faults exclusively. The boundaries between different classes are often difficult to separate by two-class rules because of the different relationships between the classes; hence, the usefulness of multi-class LAD pattern generation. The above rules can be used by maintenance personnel to validate knowledge obtained from expert rules such as the ones described in Heathcote [39], and Duval and DePablo [40], but also to investigate the new relationships that arise between the faults and the DGA indicators from the multi-class patterns.

6.6 Conclusion

This paper discusses a novel approach to detection and identification of faults in CBM using multi-class Logical Analysis of Data. A new multi-class LAD fault diagnosis approach was designed that relies on a newly introduced discriminating factor and an MILP based pattern generation algorithm. The resulting diagnostic tool was tested on several known machine learning data sets. The results revealed that the proposed approach fairs favourably in terms of accuracy levels in comparison with other classification approaches. The paper then tested the designed tool on a CBM application for the detection and identification of faults in power transformers using DGA data. The results reveal moderately good classification accuracy; however the true merit of the approach is in the interpretability of the generated patterns. These results lead to the conclusion that multi-class LAD based fault detection and identification using an MILP based pattern generation algorithm is a promising diagnostic approach in CBM due to the explanatory power of the resulting decision model through the generated patterns it is constructed from.

6.7 References

[1] A. Jardine, D. Lin, D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, Mechanical Systems and Signal Processing, 20 (2006) 1483-1510.

[2] J. Ma, J. Li, Detection of localised defects in rolling element bearings via composite hypothesis test, Mechanical Systems and Signal Processing, 9 (1995) 63-75.

[3] W. Staszewski, K. Worden, G. Tomlinson, Time-frequency analysis in gearbox fault detection using the Wigner-Ville distribution and pattern recognition, Mechanical Systems and Signal Processing, 11 (1997) 673-692.

[4] N. Jamaludin, D. Mba, R. Bannister, Condition monitoring of slow-speed rolling element bearings using stress waves, Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering, 215 (2001) 245-271.

[5] K. Christian, N. Mureithi, A. Lakis, M. Thomas, ON THE USE OF TIME SYNCHRONOUS AVERAGING, INDEPENDENT COMPONENT ANALYSIS AND SUPPORT VECTOR MACHINES FOR BEARING FAULT DIAGNOSIS, in: First International Conference on Industrial Risk Engineering, Montreal, 2007.

[6] S. Abbasion, A. Rafsanjani, A. Farshidianfar, N. Irani, Rolling element bearings multi-fault classification based on the wavelet denoising and support vector machine, Mechanical Systems and Signal Processing, 21 (2007) 2933-2945.

[7] A. Widodo, B. Yang, T. Han, Combination of independent component analysis and support vector machines for intelligent faults diagnosis of induction motors, Expert Systems with Applications, 32 (2007) 299-312.

[8] H. Ocak, K. Loparo, A new bearing fault detection and diagnosis scheme based on hidden Markov modeling of vibration signals, in: IEEE INTERNATIONAL CONFERENCE ON ACOUSTIC SPEECH SIGNAL PROCESSING Citeseer, 2001, pp. 3141-3144.

[9] Z. Li, Z. Wu, Y. He, C. Fulei, Hidden Markov model-based fault diagnostics method in speed-up and speed-down process for rotating machinery, Mechanical Systems and Signal Processing, 19 (2005) 329-339.

[10] Y. Xu, M. Ge, Hidden Markov model-based process monitoring system, Journal of Intelligent Manufacturing, 15 (2004) 337-350.

[11] S. Saitta, B. Raphael, I. Smith, Data mining techniques for improving the reliability of system identification, Advanced Engineering Informatics, 19 (2005) 289-298.

[12] M. Subrahmanyam, C. Sujatha, Using neural networks for the diagnosis of localized defects in ball bearings, Tribology International, 30 (1997) 739-752.

[13] J. Spoerre, Application of the cascade correlation algorithm (CCA) to bearing fault classification problems, Computers in Industry, 32 (1997) 295-304.

[14] R. Yam, P. Tse, L. Li, P. Tu, Intelligent predictive decision support system for condition-based maintenance, The International Journal of Advanced Manufacturing Technology, 17 (2001) 383-391.

[15] A. Saxena, A. Saad, Fault diagnosis in rotating mechanical systems using self-organizing maps, Artificial Neural Networks in Engineering (ANNIE04), (2004).

[16] S. Wu, T. Chow, Induction machine fault detection using SOM-based RBF neural networks, IEEE Transactions on Industrial Electronics, 51 (2004) 183-194.

[17] P. Yang, Q. Wang, Fault Diagnosis System for Turbo-Generator Set Based on Fuzzy Neural Network, in, 2006, pp. 228-231.

[18] W. Hu, A. Starr, Z. Zhou, A. Leung, An intelligent integrated system scheme for machine tool diagnostics, The International Journal of Advanced Manufacturing Technology, 18 (2001) 836-841.

[19] G. Alexe, S. Alexe, L. Liotta, E. Petricoin, M. Reiss, P. Hammer, Ovarian cancer detection by logical analysis of proteomic data, Proteomics, 4 (2004) 766-783.

[20] G. Alexe, S. Alexe, D. Axelrod, P. Hammer, D. Weissmann, Logical analysis of diffuse large B-cell lymphomas, Artificial Intelligence in Medicine, 34 (2005) 235-267.

[21] G. Alexe, S. Alexe, D. Axelrod, T. Bonates, I. Lozina, M. Reiss, P. Hammer, Breast cancer prognosis by combinatorial analysis of gene expression data, Breast Cancer Research, 8 (2006) R41.

[22] S. Abramson, G. Alexe, P. Hammer, J. Kohn, A computational approach to predicting cell growth on polymeric biomaterials, Journal of Biomedical Materials Research Part A, 73 (2005) 116-124.

[23] D. Salamanca, S. Yacout, Condition based maintenance with logical analysis of data, in: 7e Congrès International de genie industriel, Quebec, 2007.

[24] M. Mortada, T. Carroll, S. Yacout, A. Lakis, Rogue components: their effect and control using logical analysis of data, Journal of Intelligent Manufacturing, (2009) 1-14.

[25] S. Yacout, Fault Detection and Diagnosis for Condition Based Maintenance Using Logical Analysis of Data, in: COMADEM International Conference, Nara, 2010.

[26] L. MOREIRA, The use of Boolean concepts in general classification contexts, in, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2000.

[27] E. Mayoraz, M. Moreira, On the decomposition of polychotomies into dichotomies, (1996).

[28] M. Moreira, E. Mayoraz, Improved pairwise coupling classification with correcting classifiers, Machine Learning: ECML-98, (1998) 160-171.

[29] E. Boros, P. Hammer, T. Ibaraki, A. Kogan, E. Mayoraz, I. Muchnik, An implementation of logical analysis of data, Knowledge and Data Engineering, IEEE Transactions on, 12 (2000) 292-306.

[30] H. Ryoo, I. Jang, Milp approach to pattern generation in logical analysis of data, Discrete Applied Mathematics, 157 (2009) 749-761.

[31] M. Mortada, S. Yacout, A. Lakis, Fault Diagnosis of Power Transformers Using Logical Analysis of Data, Discrete Applied Mathematics, (Unpublished results).

[32] P. Hammer, A. Kogan, B. Simeone, S. Szedmák, Pareto-optimal patterns in logical analysis of data, Discrete Applied Mathematics, 144 (2004) 79-102.

[33] G. Alexe, S. Alexe, T. Bonates, A. Kogan, Logical analysis of data–the vision of Peter L. Hammer, Annals of Mathematics and Artificial Intelligence, 49 (2007) 265-312.

[34] M. Berkelaar, K. Eikland, P. Notebaert, lp solve, open source (mixed-integer) linear programming system, in: (GNU LGPL (Lesser General Public Licence) Version 5.5, 2004.

[35] A. Frank, A. Asuncion, UCI machine learning repository, 2010, URL http://archive. ics. uci. edu/ml.

[36] H. Almuallim, T. Dietterich, Learning boolean concepts in the presence of many irrelevant features, Artificial Intelligence, 69 (1994) 279-305.

[37] J. Quinlan, C4. 5: programs for machine learning, Morgan Kaufmann, 1993.

[38] T. Lim, W. Loh, Y. Shih, A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms, Machine learning, 40 (2000) 203-228.

[39] M. Heathcote, The J & P transformer book: a practical technology of the power transformer, Elsevier, 2007.

[40] M. Duval, A. DePablo, Interpretation of gas-in-oil analysis using new IEC publication 60599 and IEC TC 10 databases, IEEE Electrical Insulation Magazine, 17 (2001) 31-41.

[41] G. Lv, H. Cheng, H. Zhai, L. Dong, Fault diagnosis of power transformer based on multi-layer SVM classifier, Electric power systems research, 75 (2005) 9-15.

Chapter 7 GENERAL DISCUSSION

In this chapter, we discuss the contribution of the articles presented in the previous chapters towards the objectives that this thesis set out to accomplish. The first objective is to study the applicability of LAD in different CBM situations requiring diverse considerations regarding the type of input data and output decisions. This was accomplished by applying LAD to three distinct CBM applications.

The first application discussed in Chapter 3 involves the use of historical event data to detect rogue components within an inventory of serviceable parts. The resulting features which were obtained from the historical records of 2 indicators, the RRC (Reason-for-Removal Codes) and TTR (Time-To-Removal) codes of the repairable components, have been traditionally used by maintenance personnel to detect and isolate rogue components. A LAD decision model was built using a conventional bottom-up pattern generation algorithm. The accuracy results obtained using the LAD decision model implemented in Chapter 3 reveal a maximum quality measure of 99.65%. The true positive rate indicating the number of rogue components that were correctly detected during the classification is 100%. Additionally, for all three models built, no rogue components were misclassified as non-rogue and vice versa. More importantly, an examination of the patterns generated by the LAD decision models revealed that they are similar to those found through expert analysis of rogue component data. Testing results therefore showed that LAD is capable of detecting rogue components automatically using the components` performance history. This demonstrated the potential for LAD to, not only automate the decision process thus saving a lot of resources, but also to potentially generate new patterns if condition monitoring data were to be made available to it. The new patterns that could be generated based on the condition monitoring data can aid in understanding the root causes of rogue components, thus aiding in updating the PM measures put in place to avoid their occurrence.

The second application discussed in Chapters 4 and 6 involves the detection and isolation of faults in power transformers using level type data obtained from the analysis of dissolved gas in power transformers. The designed decision models classify data into multiple classes in different ways. While in chapter 4 a series of two-class multilayered LAD decision models were implemented, chapter 6 discussed the use of a multi-class LAD decision model on the same application. The multilayered approach discussed in chapter 4 was tested in two experiments on two different data sets. In one experiment LAD performed well against neural networks and fuzzy logic based approaches and was outperformed only when a novel feature selection approach was paired with an integrated neural fuzzy approach. In a second experiment the accuracy achieved was similar to ANN, ES, and Fuzzy Logic approaches but was evidently less than that achieved by SVM. The real advantage of LAD, as with the previous application, was the generation of patterns similar to the rules of standardised expert based procedures for the detection of faults. In Chapter 4, we interpret some of the patterns generated and compare them to the rules of the Rogers Ratios Method. The multi-class approach discussed in chapter 6 was tested first on data sets obtained from a machine learning data base in order to compare it to other multi-class approaches such as the LAD approach discussed in [Moreira 2000] and the famous clustering approach called C4.5 [Quinlan 1993]. The results of the comparison revealed that the multi-class LAD approach has the best overall average performance, albeit at the expense of slower training times. The multi-class approach was then tested on a data set obtained from a power transformer [Lv et al. 2005]. The results showed moderately good accuracy.

The third application discussed in Chapter 5 is different from the first two in its reliance on vibration data for the diagnosis of faults in rotor bearings. Here, unlike the first two applications, we separate the vibration signals in a signal database into normal and faulty signals using expert knowledge such as visual inspection of processed signal features. From the separated database, we extract the training and testing data sets of the decision model. The LAD decision model is then trained and tested in two experiments on two rotor bearings exhibiting different failure conditions to find out if it is capable of automating the decision process. The results of the decision models show a maximum accuracy ranging between 95.2% and 97.5% for the first experiment and between 97.1% and 98.9% for the second. The patterns generated by the LAD decision model provide new insight on how to classify vibration signals. The patterns found had a classification power of more than 80% in some cases, which means that they covered more than 80% of normal or faulty signals.

The second objective set out by this thesis is to adapt the LAD methodology to suit the particular requirement of each CBM application. These particular requirements were demonstrated in Chapters 4 and 6 when a multi-class model was needed to classify the data, and again in Chapters 4, 5, and 6 when the need to generate more patterns was the motivation behind the modification of the pattern generation procedure. The innovative modifications led to the fulfillment of the third objective: to improve the LAD methodology in order to increase diagnosis accuracy and result interpretability.

The need to generate more patterns was motivated by the desire to increase the differentiability of the discriminant function obtained at the end of the LAD decision model and by the wish to increase the amount of useful knowledge generated by the decision model. This has been achieved by introducing a new set of conditions to the MILP based pattern generation methodology and by modifying the mechanism that loops the MILP pattern generation algorithm. Chapter 4 introduced the following set of conditions to the methodology:

$$\sum_{j=1}^{2q} r_{k,j} w_j \le d_k - 1 \quad \forall \mathbf{r}_k \in \mathbf{R}$$

$$r_{k,j} = \begin{cases} 1 & if \quad v_{k,j} = 1 \\ -1 & if \quad v_{k,j} = 0 \end{cases} \quad j = \{1,2,\ldots,2q\}$$

where $\mathbf{r}_k$ is the vector corresponding to each found pattern $\mathbf{v}_k \in \mathbf{V}$ of degree d_k and $\mathbf{w}$ is the new candidate pattern vector. The new set of conditions and looping scheme allow for the generation of more than one pattern per single observation, such that the next generated pattern is also the "strongest". This however leads to the generation of patterns that are subsets of previously generated ones, which created some redundancy that affected the pattern interpretation process. To alleviate that problem the set of conditions were modified in Chapter 5 to be:

$$\sum_{j=1}^{2q} v_{k,j} w_j \le d_k - 1 \quad \forall \mathbf{v}_k \in \mathbf{V}$$

$$v_{k,j} = \begin{cases} 1 & if \quad w_{k,j} = 1 \\ 0 & if \quad w_{k,j} = 0 \end{cases} \quad j = \{1,2,\ldots,2q\}$$

This condition does not allow patterns that are subsets of previously found patterns to be generated. This consequently increases the amount of new knowledge that can be created. Due to the introduction of the above modifications, the average accuracy of the decision models increased when compared to accuracy levels obtained without the above modifications.

The need for a multi-class decision model in the case of power transformer fault diagnosis led to the two LAD multi-class configurations discussed in Chapter 4 and 6. The first paper takes a simple approach by cascading the two-class LAD decision models in a multilayer configuration in order to achieve a multi-class decision model. This approach is similar to the first approach studied in [Mayoraz & Moreira 1996, Moreira & Mayoraz 1999, Moreira 2000]. This configuration achieved good accuracy results; however the decision model obtained was too complex. Additionally, although each two-class decision model had a good average training time, the need for training each model independently took a lot of time. This was addressed in Chapter 6 with a novel multi-class LAD approach based on MILP based pattern generation. When compared to the multi-class decision models presented in the literature, the new multi-class LAD approach performed comparably to the best classifier and was superior to the multi-class LAD presented in [Moreira 2000] in terms of classification accuracy. This however came at the expense of increased training time. Additionally, the obtained patterns were more intuitive and meaningful as they related more than 2 classes at the same time. The advantage of this feature in CBM was explained in chapter 6.

So far we discussed how the objectives of the thesis have been fulfilled through the research presented in Chapters 3 to 6 of this thesis. Next we discuss the role of LAD in solving the problems that this thesis has set out to solve: inappropriate PM actions that lack rational explanation to maintenance personnel.

As mentioned on many occasions, in addition to automating the decision process, LAD creates interpretable patterns as demonstrated in the three applications studied throughout this research. These patterns can play a big role in solving the problems mentioned above. The generated patterns can:

1- Help maintenance personnel confirm expert opinions and rules used in the industry. This was demonstrated in the rogue component detection problem and power transformer fault identification problem when some of the generated patterns were identical to rules devised using expert knowledge.

2- Help maintenance personnel better understand the behaviour of the equipment they are monitoring through the new information they provide. As shown in the power transformers application LAD also generates new patterns that provide new information to maintenance personnel that have the potential to help them better understand the reasons that lead to the failure of certain equipment and the conditions under which they fail.

3- Aid in upgrading PM actions and updating maintenance strategies. Maintenance personnel can use the confirmed observed patterns and the new generated knowledge, to analyze their maintenance strategies and update their PM strategies in order to avoid unnecessary system outages.

4- Help assess the use of expensive monitoring technologies. The patterns generated through LAD are functions of the data features used to monitor the condition of equipment or assets. The generated patterns provide insight on which data features are most effective in monitoring the equipment for a certain fault type. Based on this information, maintenance personnel can better manage their data acquisition strategies to obtain more pertinent information.

The use of LAD in CBM is therefore justified by:

1- Its good performance in terms of accuracy and training time.

2- The adaptability of the approach to the different requirements in terms of the type of application being addressed, the nature of the input data, and the types of maintenance decisions needed.

3- The advantage of the pattern interpretability property that LAD possesses over other decision making approaches and the role that pattern interpretability plays in aiding maintenance personnel solve the problems mentioned in the beginning of this thesis.

CONCLUSION

This thesis studied the implementation of a new decision making approach in CBM based on LAD. The approach possesses advantages over other conventional decision making models that have been proven throughout this research to be an asset in the field of maintenance. LAD was tested on three CBM applications for the first time: the detection of rogue components in an inventory of reparable parts, the diagnosis of power transformer faults, and the detection of faults in rotor bearings.

In the first application, it was demonstrated that a LAD based decision model can automate the detection of rogue components, using event data collected from maintenance records, with high accuracy. In addition, the patterns generated by LAD where found to confirm the patterns observed by maintenance experts.

In the second application, a LAD based decision model was implemented for the diagnosis of multiple fault types in power transformers using DGA condition monitoring data. The LAD decision model was therefore capable of achieving a higher accuracy than expert based decision models and comparable to automatic decision models based on statistical processes and artificial intelligence. However, thanks to the interpretability property of LAD, the applied decision model was capable of generating a set of interpretable decision rules, similar to those on which expert based decision models are based. The rules generated from the LAD patterns can be regarded as new knowledge due to their distinctiveness from any of those used in the expert decision models.

In the third application, the ability of LAD to diagnose bearing faults using vibration signal analysis was demonstrated. The LAD based decision model was capable of diagnosing bearing faults at a high accuracy rate. In addition, the interpretability property of LAD allowed for an analysis of the data features used in generating the LAD patterns. This in turn gave insight on the type of signal processing technique that is best suited for the detection of bearing faults, as it was shown that the data features obtained from time-frequency based signal processing techniques were more useful than those obtained from time based signal processing techniques.

The success with which LAD was applied on three CBM applications with different requirements in terms of the type of data they admit and the type of maintenance decisions required of the designed CBM program lead to the conclusion, by inductive generalization, that

LAD is a promising approach that is applicable in the field of CBM due to its good performance and its adaptability. In addition, thanks to the interpretability of the patterns that it generates, LAD can provide insight and added knowledge to the maintenance organization as demonstrated in the three applications described in this thesis. In addition, the interpretability of the generated patterns provides feedback on the data features, and hence condition monitoring technologies, that are best suited for a certain application.

The original contributions of this thesis are the introduction of innovative modifications to the pattern generation procedure as well as a novel procedure that transforms LAD to a multi-class classifier. The modifications to the MILP based pattern generation algorithm resulted in improved performance in the case of power transformer fault diagnosis and bearing fault detection. The novel multi-class LAD procedure achieved higher accuracy than the existing multi-class LAD methodology. In addition, the new procedure achieved the best overall accuracy when tested on machine learning databases and compared to other popular rule induction based algorithms. The new algorithms devised in this thesis were demonstrated on three diagnostic CBM with success. However, these algorithms can easily be adapted to other diagnostic applications in CBM as well as other applications requiring classification.

As a result of the research conducted on LAD throughout this thesis, we have identified two limitations of the methodology. The first limitation is the inapplicability of the LAD based decision making approaches described in this thesis on prognostic CBM applications. Prognosis involves the estimation of the probability that a certain piece of equipment will fail as well as the time left until this failure occurs. Although this research has demonstrated the applicability of LAD in diagnostic applications esclusively, some initial tests on prognostic applications have been conducted with mixed results. The potential benefits from adapting LAD to prognostics are high. As such, it is suggested that future research on LAD focus on the development of new pattern interpretation procedures that would help apply LAD on prognostic CBM applications.

The second limitation of LAD, in comparison to other automatic decision models, is that it is a supervised learning approach. This means that the applicability of LAD depends on the existence of preclassified data, needed to train the decision model. This problem can be circumvented with the availability of expert knowledge that can be used in conjunction with LAD in unsupervised learning environments. This has been demonstrated successfully in the first article of this thesis with the use of LAD with the help of the expert knowledge of maintenance personnel for rogue component detection. However, an unsupervised learning LAD decision model would help bypass the expert elicitation procedure. The transformation of LAD from a supervised learning to an unsupervised learning approach is therefore suggested as an important area of focus in any future research on LAD. The. This step is particularly challenging as it requires changing binarization and pattern generation procedures, however the potential consequences of its achievement are promising.

In the end, some final words are in order to emphasize the potential of LAD in solving the unique rogue component detection problem addressed in the first article of this thesis. The ability of LAD to detect rogue components with high accuracy was clearly demonstrated in chapter 3. However the real advantage, as has been pointed out, is the ability of the patterns to automate the expert knowledge of maintenance personnel. As the data features used for the detection of rogue components were identical to those used by maintenance experts, the generated patterns were able to duplicate and confirm the experts' observed patterns. However, the potential for the use of LAD in rogue component detection is much greater than mere confirmation of already existing knowledge. If given access to data collected from the standard tests applied on components at the repair shop level, the LAD methodology is capable of generating new knowledge that can potentially help discover the reasons behind the phenomenon of rogue component development. Unfortunately, this type of data was not made available to us in our research on LAD decision models. As such, we leave this matter to future research endeavours that might have better access to this type of data.

References

Abbasion, S., A. Rafsanjani, et al. (2007). "Rolling element bearings multi-fault classification based on the wavelet denoising and support vector machine." Mechanical Systems and Signal Processing 21(7): 2933-2945.

Abramson, S., G. Alexe, et al. (2005). "A computational approach to predicting cell growth on polymeric biomaterials." Journal of Biomedical Materials Research Part A 73(1): 116-124.

Alexe, G., S. Alexe, et al. (2006). "Breast cancer prognosis by combinatorial analysis of gene expression data." Breast Cancer Research 8(4): R41.

Alexe, G., S. Alexe, et al. (2005). "Logical analysis of diffuse large B-cell lymphomas." Artificial Intelligence in Medicine 34(3): 235-267.

Alexe, G., S. Alexe, et al. (2007). "Logical analysis of data–the vision of Peter L. Hammer." Annals of Mathematics and Artificial Intelligence 49(1): 265-312.

Alexe, G., S. Alexe, et al. (2008). "Comprehensive vs. comprehensible classifiers in logical analysis of data." Discrete Applied Mathematics 156(6): 870-882.

Alexe, G., S. Alexe, et al. (2004). "Ovarian cancer detection by logical analysis of proteomic data." Proteomics 4(3): 766-783.

Alexe, G. and P. Hammer (2006). "Spanned patterns for the logical analysis of data." Discrete Applied Mathematics 154(7): 1039-1049.

Alexe, S. and P. Hammer (2006). "Accelerated algorithm for pattern detection in logical analysis of data." Discrete Applied Mathematics 154(7): 1050-1063.

Almuallim, H. and T. Dietterich (1994). "Learning boolean concepts in the presence of many irrelevant features." Artificial Intelligence 69(1-2): 279-305.

Baillie, D. and J. Mathew (1996). "A comparison of autoregressive modeling techniques for fault diagnosis of rolling element bearings." Mechanical Systems and Signal Processing 10(1): 1-17.

Berkelaar, M., K. Eikland, et al. (2004). lp solve, open source (mixed-integer) linear programming system. (GNU LGPL (Lesser General Public Licence) Version 5.5.

Bloch, H. and F. Geitner (1983). Machinery failure analysis and troubleshooting, Gulf Publishing Company.

Bonates, T., P. Hammer, et al. (2008). "Maximum patterns in datasets." Discrete Applied Mathematics 156(6): 846-861.

Boros, E., P. Hammer, et al. (1997). "Logical analysis of numerical data." Mathematical Programming 79(1): 163-190.

Boros, E., P. Hammer, et al. (2000). "An implementation of logical analysis of data." IEEE Transactions on Knowledge and Data Engineering 12(2): 292-306.

Butcher, S. (2000). Assessment of Condition-Based Maintenance in the Department of Defense. McLean, VA, LOGISTICS MANAGEMENT INSTITUTE: 1-70.

Christian, K., N. Mureithi, et al. (2007). On the Use of Time Synchronous Averaging, Independent Component Analysis and Support Vector Machines for Bearing Fault Diagnosis. First International Conference on Industrial Risk Engineering. Montreal, Canada.

Chvatal, V. (1979). "A greedy heuristic for the set-covering problem." Mathematics of operations research 4(3): 233-235.

Chvatal, V. (1979). "A greedy heuristic for the set-covering problem." Mathematics of operations research 4(3): 233-235.

COIN-OR (2004). CLP, Computational Infrastructure for Operational Research.

Crama, Y., P. Hammer, et al. (1988). "Cause-effect relationships and partially defined Boolean functions." Annals of Operations Research 16(1): 299-325.

Dhillon, B. (2002). Engineering maintenance: a modern approach, CRC.

Dhillon, B. (2006). Maintainability, maintenance, and reliability for engineers, CRC press.

Diederich, J. (2008). "Rule extraction from support vector machines: An introduction." Rule Extraction from Support Vector Machines: 3-31.

Evans, J. and W. Lindsay (2004). An introduction to Six Sigma & process improvement, Thomson/South-Western.

Frank, P., S. Ding, et al. (2000). "Model-based fault diagnosis in technical processes." TRANSACTIONS-INSTITUTE OF MEASUREMENT AND CONTROL 22(1): 45-102.

Fugate, M., H. Sohn, et al. (2001). "Vibration-based damage detection using statistical process control." Mechanical Systems and Signal Processing 15(4): 707-721.

Fukunaga, K. (1990). Introduction to statistical pattern recognition, Academic Pr.

Garamone, J. (2008). "Bush Delivers $515.4 Billion Defense Budget Request to Congress." from http://www.defense.gov/news/newsarticle.aspx?id=48860.

Hamdy, N. (2008). "Applied Signal Processing: Concepts, Circuits, and Systems."

Hammer, P. (1986). Partially defined Boolean functions and cause-effect relationships. Internationall Conference on Multi-attribute decision making via OR-Based expert systems.

Hammer, P. and T. Bonates (2006). "Logical analysis of data—An overview: from combinatorial optimization to medical applications." Annals of Operations Research 148(1): 203-225.

Hammer, P., A. Kogan, et al. (2004). "Pareto-optimal patterns in logical analysis of data." Discrete Applied Mathematics 144(1-2): 79-102.

Heng, A., S. Zhang, et al. (2009). "Rotating machinery prognostics: State of the art, challenges and opportunities." Mechanical Systems and Signal Processing 23(3): 724-739.

Howard, I. (1994). A Review of rolling element bearing vibration" detection, diagnosis and prognosis". Melbourne, Aeronautical and Maritime Research Laboratory: 94.

Hu, W., A. Starr, et al. (2001). "An intelligent integrated system scheme for machine tool diagnostics." The International Journal of Advanced Manufacturing Technology 18(11): 836-841.

ILOG, I. (2003). "CPLEX 9.0 Reference Manual." ILOG CPLEX Division.

Isermann, R. (2005). "Model-based fault-detection and diagnosis-status and applications." Annual Reviews in control 29(1): 71-85.

Isermann, R. and P. Balle (1997). "Trends in the application of model-based fault detection and diagnosis of technical processes." Control Engineering Practice 5(5): 709-719.

Jamaludin, N., D. Mba, et al. (2001). "Condition monitoring of slow-speed rolling element bearings using stress waves." Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering 215(4): 245-271.

Jardine, A., D. Lin, et al. (2006). "A review on machinery diagnostics and prognostics implementing condition-based maintenance." Mechanical Systems and Signal Processing 20(7): 1483-1510.

Kecman, V. (2005). "Support Vector Machines–An Introduction." Support vector machines: theory and applications: 1-47.

Korbicz, J., J. Koscielny, et al. (2004). Fault diagnosis: models, artificial intelligence, applications, Springer Verlag.

Lee, H., D. Park, et al. (2000). "A fuzzy expert system for the integrated fault diagnosis." IEEE Transactions on Power Delivery 15(2): 833.

Levitt, J. (2003). Complete guide to preventive and predictive maintenance, Industrial Press Inc.

Li, Z., Z. Wu, et al. (2005). "Hidden Markov model-based fault diagnostics method in speed-up and speed-down process for rotating machinery." Mechanical Systems and Signal Processing 19(2): 329-339.

Liu, T., J. Singonahalli, et al. (1996). "Detection of roller bearing defects using expert system and fuzzy logic." Mechanical Systems and Signal Processing 10(5): 595-614.

Loparo, K., M. Adams, et al. (2000). "Fault detection and diagnosis of rotating machinery." IEEE Transactions on Industrial Electronics 47(5): 1005-1014.

Lv, G., H. Cheng, et al. (2005). "Fault diagnosis of power transformer based on multi-layer SVM classifier." Electric power systems research 75(1): 9-15.

Ma, J. and J. Li (1995). "Detection of localised defects in rolling element bearings via composite hypothesis test." Mechanical Systems and Signal Processing 9(1): 63-75.

Mandal, M. and A. Asif (2007). Continuous and discrete time signals and systems, Cambridge University Press.

Márquez, A. (2007). The maintenance management framework: Models and methods for complex systems maintenance, Springer Verlag.

Mayoraz, E. and M. Moreira (1996). "On the decomposition of polychotomies into dichotomies."

Mobley, R. (2002). An introduction to predictive maintenance, Butterworth-Heinemann.

Monsef, H., A. Ranjbar, et al. (1997). "Fuzzy rule-based expert system for power system fault diagnosis." IEE Proceedings-Generation, Transmission and Distribution 144: 186.

Moreira, M. (2000). The use of Boolean concepts in general classification contexts. Lausanne, Ecole Polytechnique Federale de Lausanne. PhD.

Moreira, M. and E. Mayoraz (1998). "Improved pairwise coupling classification with correcting classifiers." Machine Learning: ECML-98: 160-171.

Ocak, H. and K. Loparo (2001). A new bearing fault detection and diagnosis scheme based on hidden Markov modeling of vibration signals, Citeseer.

Quinlan, J. (1993). C4. 5: programs for machine learning, Morgan Kaufmann.

Ryoo, H. and I. Jang (2009). "Milp approach to pattern generation in logical analysis of data." Discrete Applied Mathematics 157(4): 749-761.

Safizadeh, M., A. Lakis, et al. (1999). Application of short-time fourier transform in machine fault detection, Ecole Polytechnique de Montreal.

Saitta, S., B. Raphael, et al. (2005). "Data mining techniques for improving the reliability of system identification." Advanced Engineering Informatics 19(4): 289-298.

Salamanca, D. and S. Yacout (2007). Condition based maintenance with logical analysis of data. 7e Congrès International de genie industriel. Quebec.

Samanta, B. (2004). "Gear fault detection using artificial neural networks and support vector machines with genetic algorithms." Mechanical Systems and Signal Processing 18(3): 625-644.

Saxena, A. and A. Saad (2004). "Fault diagnosis in rotating mechanical systems using self-organizing maps." Artificial Neural Networks in Engineering (ANNIE04).

Saxena, A. and A. Saad (2007). "Evolving an Artificial Neural Network Classifier for Condition Monitoring of Rotating Mechanical Systems." Applied Soft Computing 7: 441–454.

Simani, S., C. Fantuzzi, et al. (2003). Model-based fault diagnosis in dynamic systems using identification techniques, Springer London.

Smith, A. and G. Hinchcliffe (2004). RCM: gateway to world class maintenance, Butterworth-Heinemann.

Spoerre, J. (1997). "Application of the cascade correlation algorithm (CCA) to bearing fault classification problems." Computers in Industry 32(3): 295-304.

Staszewski, W., K. Worden, et al. (1997). "Time-frequency analysis in gearbox fault detection using the Wigner-Ville distribution and pattern recognition." Mechanical Systems and Signal Processing 11(5): 673-692.

Steinwart, I. and A. Christmann (2008). Support vector machines, Springer Verlag.

Subrahmanyam, M. and C. Sujatha (1997). "Using neural networks for the diagnosis of localized defects in ball bearings." Tribology International 30(10): 739-752.

Sun, Q., P. Chen, et al. (2004). "Pattern recognition for automatic machinery fault diagnosis." Journal of vibration and acoustics 126: 307.

Sun, W., J. Chen, et al. (2007). "Decision tree and PCA-based fault diagnosis of rotating machinery." Mechanical Systems and Signal Processing 21(3): 1300-1317.

Tu, J. V. (1996). "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes." Journal of Clinical Epidemiology 49(11): 1225-1231.

Widodo, A., B. Yang, et al. (2007). "Combination of independent component analysis and support vector machines for intelligent faults diagnosis of induction motors." Expert Systems with Applications 32(2): 299-312.

Wu, S. and T. Chow (2004). "Induction machine fault detection using SOM-based RBF neural networks." IEEE Transactions on Industrial Electronics 51(1): 183-194.

Xu, Y. and M. Ge (2004). "Hidden Markov model-based process monitoring system." Journal of Intelligent Manufacturing 15(3): 337-350.

Yacout, S. (2010). Fault Detection and Diagnosis for Condition Based Maintenance Using Logical Analysis of Data. COMADEM International Conference. Nara, Japan.

Yam, R., P. Tse, et al. (2001). "Intelligent predictive decision support system for condition-based maintenance." The International Journal of Advanced Manufacturing Technology 17(5): 383-391.

Yang, P. and Q. Wang (2006). Fault Diagnosis System for Turbo-Generator Set Based on Fuzzy Neural Network.