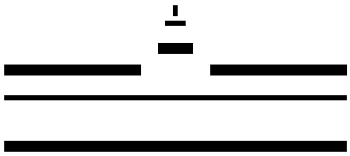


Title:

Application of Data Analytics in Failure Pattern Recognition

Seminar Thesis

In the context of the seminar "Application of Data Analytics in Spare Parts Supply Chain Management"
at the Chair for Information Systems and Supply Chain Management

Supervisors: Dr.-Ing.Christian Grimme
Dipl.-Inf. Jakob Bossek
Carolin Wagner MScIS

Presented by: Alexander Anokhin
Busso-Peus-Str. 14
Muenster 48149
Germany
+49 176 75-615943
a_anok01@uni-muenster.de

Joshua Peter Handali
Steinfurter Str. 81
Muenster 48149
Germany
+49 159 01-002038
j_hand02@uni-muenster.de

Date of submission: August 1, 2015

Contents

Figures

Tables

Abbreviations

ANN	Artificial Neural Networks
ANOVA RBF	ANOVA Radial Basis Function
CART	Classification and Regression Trees
CBM	Condition Based Maintenance
CNN	Condensed Nearest Neighbour
DET	Distance Evaluation Technique
ENN	Edited Nearest Neighbour
GA	Genetic Algorithm
GS	Grid Search
ICA	Independent Component Analysis
NCL	Neighbourhood Cleaning Rule
OSS	One-Sided Sampling
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
RFE	Recursive Feature Elimination
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine

Introduction

A successful failure pattern recognition system can go a long way in improving spare parts management in terms of allowing a good maintenance plan, fault detection manifests itself as a part of Condition Based Maintenance (CBM) [16]. Failure pattern recognition is aimed not only to differentiate between failure and normal operation signals, but also to distinguish between different failure patterns. Identifying patterns benefits in early failure detection and investigation of root cause of failure. Meanwhile, distinguishing different failure patterns may lead to effective fault management action plans, as failures can then be accurately located. These opportunities lead to efforts on automatic failure pattern recognition, one of which can be done by applying classification algorithms such as Support Vector Machine (SVM).

SVM is a relatively new computational learning method based on the statistical learning theory. Exhaustive overviews of SVM applicability in pattern recognition [3] and fault diagnosis [30] clearly show that SVM is a versatile and efficient technique for such problems. The concept of SVM is based on Vapnik-Chervonenkis theory [4, 26] that recently emerged as a general mathematical framework for estimating (learning) dependencies from finite samples [30, p. 2561]. The idea of SVM is to separate feature space by constructing linear boundary with the biggest margin between two classes. Moreover initial feature space can be enlarged with help of kernel transformations.

SVM approach is not only theoretically well-founded, but also superior in practice. Recent surveys show that tuned SVM classifiers has become more efficient in pattern recognition than other methods: Artificial Neural Networks (ANN) [7, 10, 11, 17, 21] and Classification and Regression Trees (CART) [15, 21]. In addition, SVMs have one significant advantage compared to conventional methods of pattern recognition such as ANNs. These methods struggle to solve problems with a small number of samples. For the reason that it is hard to obtain sufficient fault samples in practice, SVM is introduced into machines fault diagnosis due to its high accuracy and good generalization for a smaller number of samples [30, p. 2562].

An aspect of pattern recognition task typically needs to be addressed is the class imbalance problem. It occurs where one or more classes heavily outnumber other classes. This is usually the case when putting patterns from normal operations and failure occurrences together, which also presents in datasets available for this paper. Comprehensive reviews on this matter are available in literatures [14, 9] which also provide approaches to solve it. Sampling method is one of those approaches. It looks to train the classifier on a resampled dataset which involves reducing the size of majority classes and / or increasing the size of minority classes. Although sampling methods for binary classification tasks are more common, multi class sampling methods can also be found in researches [20, 27].

Despite its robustness, SVMs are still susceptible to a degradation of performance when applied in imbalanced datasets [24]. With that knowledge, basic sampling methods are pursued to explore their effects on this specific failure pattern recognition task using SVMs. Error rate is compared to investigate which sampling methods, if any, may improve the classification performance.

The first section of this paper describes the underlying problem, research methodology and provided experimental data. Section 2 explores data preparation step

which involves preprocessing the raw data and exploratory analysis. The analysis work starts with data balancing experiments in Section 3. This section covers both binary and multi class datasets. Section 4 provides a step-by-step SVM model building process, ranging from choice of kernel up to final model and further improvements. Additionally, dimensionality reduction and feature selection steps are also included in this section. Finally, Section 5 summarizes results of the analysis and outlines possibilities for future researches.

1 Problem Description

1.1 Research Methodology

Aim of analysis is to build a classifier, based on initially collected data, to detect failures. Such a diagnosis can highly benefit company in real life, since failures identified at early stage, future costly faults possibly prevented. The aim defines a number of research questions. First, it is unclear how the initially proposed data can be properly preprocessed. Second, since classes are possibly imbalanced another question is how they can be balanced and should they be balanced at all. Third, feature extraction and selection as well as SVM parametrization highly effect final results it will be also valuable to answer the question about optimal building procedure and parameters. Last question is how accuracy and efficiency of a classifier can vary along different settings, for example size of initial sample.

To accomplish the aim of a study research design is mainly based on optimization and simulation experiments, however literature research is also used in two ways: establishing a proper direction of analysis; identification of existing methods and approaches for further usage. Analysis is conducted with help of R programming language[1] which is used for statistical computing and graphics. Collaboration between authors is supported by usage of GitHub[2]. The whole analysis is fully reproducible and can be redone by sourcing scripts from public GitHub repository[3].

1.2 Data Description

The initial data is provided by a Brazilian manufacturer. The company produces different kinds of electric actuators to drive control values. Data are generated in a test environment under normal and overload conditions. Three sensors are installed near the gears that are able to record the incurring vibration. The test environment represents electric motor with three attached sensors. Sensor 1 is located in the bearing of the main spindle, Sensor 2 on the motor's bearing and Sensor 3 depicts a built-in torque sensor.

Tests perform valve aperture actions and are initiated under different conditions. New gears are considered with and without additional load. In addition, failure data is obtained using worn and broken gears therefore provided data comprises six different settings with three types of failures. For every second 2048 observations

[1] http://www.r-project.org/

[2] https://github.com/

[3] https://github.com/alexanderAnokhin/fault_diagnosis

of signal are collected. Each cycle lasts about 46-47 seconds. Table 1 summarizes generated data.

Class	Description	Cycles	Observations
Normal 1	Cycle without load on actuator	25	$2398 * 10^3$
Normal 2	Cycle with pressure equal 3.0 bar	25	$2572 * 10^3$
Normal 3	Cycle with pressure equal 1.0 bar	25	$2392 * 10^3$
Failure 1	Cycle with a worn gear without load	10	$930 * 10^3$
Failure 2	Cycle with two worn gears without load	2	$195 * 10^3$
Failure 3	Cycle with a broken gear without load	10	$946 * 10^3$

Table 1: Generated Data

Table 1 outlines three important aspects of further analysis. First, provided data are imbalanced along classes. The most imbalanced case is between Normal 2 and Failure 2 classes. The imbalance ratio here accounts for 75:1000. Second, generated data consist of about 10 millions observations in total which makes computationally impossible to use SVM directly. Third, slightly more emphasis in analysis is putted on comparison of Normal 1 and failure classes since they have been all obtained without load. That is premised in assumption of being able to measure load directly. Since load is measured we can explicitly exclude classes with or without load depending on estimated load.

2 Data Preparation

2.1 Data Preprocessing

To achieve appropriate performance and classification results data preprocessing step should be done beforehand. The aim of data preprocessing from one side is to reduce the noise in the data and from another side retain as much information as possible. Typically data preprocessing also requires feature extraction step where a set of reasonable features extracted from raw signals.

Generated data contain no missing data or obvious outliers which makes preprocessing step very straightforward. However one issue that affected performance at early stages is discovered. Every cycle has steady states in the beginning and in the end where signal does not change over about one second. That obviously distracts classifier since steady signal can be misclassified as failure which lead to full stop. To avoid such scenarios every second in a cycle is divided into 16 time windows and then for all of them standard deviation is calculated. In result if standard deviation for a certain time window is less than 5% quantile for a whole cycle then signal during that period of time is considered as stable and related observations removed from further consideration. This procedure results in accurate removals of steady states from every cycle while preserving the general structure of a signal behaviour.

Feature extraction step for vibration signal has been already widely explored in literature. Samanta [18] during this step extracted time domain features in order to identify gear faults. That approach was later improved by Soleimani et al. [23], vibration was considered not just as time varying but also as a physical signal, thus time domain and frequency domain features were extracted. In this analysis, based

on existing research conducted by Soleimani et al. [23, p. 2], vibration signal is considered in time interval of 5 seconds where time domain features are extracted. Table 2 presents extracted time domain features.

Feature	Formula
mean	$T_1 = \frac{1}{N}\sum_{i=1}^{N} x_i$
standard deviation	$T_2 = (\frac{1}{N}\sum_{i=1}^{N}(x_i - T_1)^2)^{0.5}$
peak	$T_3 = max\{x_i\}, i = 1, ..., N$
root mean square	$T_4 = (\frac{1}{N}\sum_{i=1}^{N} x_i^2)^{0.5}$
crest factor	$T_4 = \frac{T_3}{T_4}$
skewness	$T_4 = \frac{1}{T_2^6}\sum_{i=1}^{N}(x_i - T_1)^3$
kurtosis	$T_4 = \frac{1}{T_2^8}\sum_{i=1}^{N}(x_i - T_1)^4$

Table 2: Extracted Time Domain Features

After data preprocessing and feature extraction, size of data is significantly reduced even without feature selection step. Final training sample consist of about 920 observations in total that makes possible to apply SVM in order to classify gear failures. Every observation consists of 21 features: 7 extracted features for every sensor signal, and response variable which shows existence of failure.

2.2 Exploratory Analysis

Another important step in order to build classifier is exploratory analysis. The aim of exploratory analysis is to understand data and check possible underlying assumptions required for analysis. It helps to find incentives for further steps and omit impossible scenarios which saves time and resources. SVM approach is very versatile and does nor require data to follow specific distribution, that fact shortens exploratory step.

Since features are extracted from one signal there is a possibility that some of them are highly correlated and will create unnecessary computational overhead, during the process of training, without improvement of accuracy. It can be clearly seen from Table 3a that features correlate along one sensor, for example, standard deviation of signal from Sensor 1 functionally determines root mean square for the same signal. In addition, features from different sensors correlate as well. Standard deviation of signal from Sensor 2 correlates with standard deviation of signal but from Sensor 3, which is shown at Table 3b. These issues highlights the problem of dimensionality reduction which will be explored further in Section 4.2.

It was outlined before that SVM approach tries to find distinguishing hyperplane between two classes therefore during exploratory analysis it is also valuable to explore how separable observations from different classes. Figure 1 shows that some classes can be accurately separated using only two features. However it is not the common case for two-dimensional space where some classes have a lot of overlap. For example, classes with normal conditions are hardly separable in two-dimensional space. Of course, using more than two features can lead to accurate separation but it cannot be properly visualized.

Exploratory part here does not pretend to be exhaustive but the main goal is satisfied. Namely it helped understand more deeply data and provided further

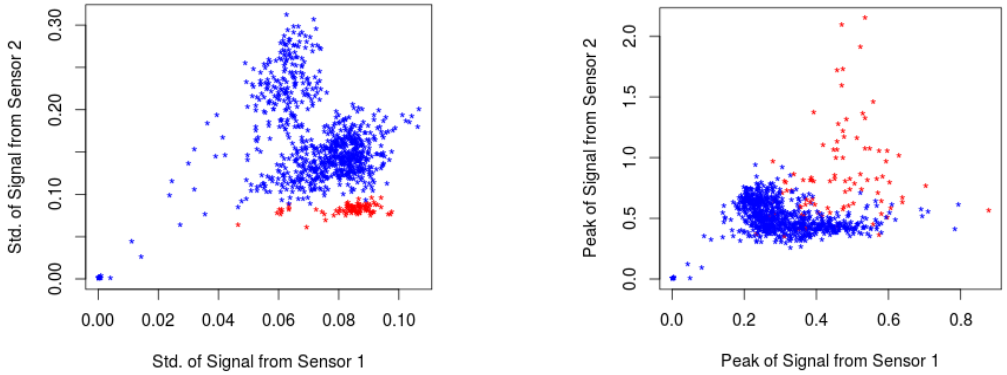
	1	2	3	4
1	1	0	.04	0
2		1	.57	1
3			1	.57
4				1

(a) Sensor 1

	5	6	7
5	1	.02	.4
6		1	.54
7			1

(b) All Sensors

Table 3: Correlation Matrices. 1 - mean; 2 - standard deviation; 3 - peak; 4 - root mean square; 5, 6, 7 - standard deviations for sensors 1, 2, 3 respectively.



(a) Failure 1

(b) Failure 3

Figure 1: Separation of Classes. Red stars represent certain failure classes, blue points represent the rest of observations.

incentives: extracted features are correlated; separation of classes is possible however there is overlap between some classes.

3 Data Balancing

3.1 Sampling Methods

There are eight different sampling methods plus a variant of one of those methods for the binary experiment. The original articles for each algorithms are cited accordingly. Furthermore, this paper used their implementations available in the *unbalanced*[4] package on R. The algorithm multi class sampling method, however, is directly implemented according to its source article while still utilizing a couple basic sampling algorithms from the *unbalanced* package.

Binary Sampling Methods. Nine sampling methods for binary datasets are composed of one oversampling method, six undersampling method, and 2 variants of a hybrid sampling method. Short descriptions of the methods are presented in Table 4. For the oversampling method random oversampling is chosen. Meanwhile, the six undersampling methods are random undersampling, Condensed Nearest Neighbour (CNN) [8], Edited Nearest Neighbour (ENN) [32], One-Sided Sampling (OSS) [12],

[4]https://cran.r-project.org/web/packages/unbalanced/unbalanced.pdf

Neighbourhood Cleaning Rule (NCL) [13], and Tomek link [1]. Synthetic Minority Oversampling Technique (SMOTE) becomes the representative for hybrid methods.

Further remarks regarding Tomek link, OSS, and SMOTE are as follows. Tomek link are pairs of instances with different classes which are each other's nearest neighbour. As for the implementation of OSS, a small change to the original function *ubOSS* from the *unbalanced* packaged was made to ensure the application of CNN is followed by Tomek links removal. Finally, SMOTE interpolates feature values between minority instances and one or more of its 5 minority class nearest neighbours to create new minority instances. This oversampling procedure is followed by undersampling the majority class, indicating the existence of adjustable oversampling and undersampling rates. For example, setting 200% on both resampling rates means two new minority instances are generated for every original minority instances and two majority instances are randomly included into the resampled training set for every new minority instances. In this paper SMOTE A and SMOTE B differ only in their oversampling rates with both variants have an undersampling rate of 200%. SMOTE A ensures a 50:50 class ratio and majority class reduction as the majority to minority ratio in the binary datasets are all larger than 2:1. SMOTE B has the default rate, which may lead to inflating instead of deflating the majority class. This occurs when the size of majority class is smaller than four times the size of minority class, which was the case for some of the binary datasets at hand.

Name	**Codename**	**Description**
Random Over-sampling	ros	Randomly replicating minority instances to reach 50:50 class ratio.
SMOTE A	smo50	SMOTE with 100% oversampling rate.
SMOTE B	smoD	SMOTE with 200% oversampling rate.
Random Under-sampling	rus	Randomly removing majority instances to reach 50:50 class ratio.
CNN	cnn	Selects a subset of majority class that correctly classify the original majority class with 1-nearest neighbour rule.
ENN	enn	Removes majority instances that differ from the majority of its 3-nearest neighbours.
OSS	oss	Application of a modified CNN followed by removing majority instances participating in Tomek links.
NCL	ncl	Removes majority instances different to their 3-nearest neighbours and the ones belonging to the 3-nearest-neighbours of minority instances.
Tomek link	tom	Majority instances which are Tomek links are removed.

Table 4: Binary Sampling Methods. Method names are followed by their codenames referred in Figure 2 and Figure 3, which are then followed by their descriptions.

Multi Class Sampling Methods. Two fairly straightforward methods to balance out sizes of multiple classes in one dataset are adapted from [20], namely Standard Mean (SMean) and Standard Median (SMedian). Both methods create a multi class dataset with equal class sizes. In order to decide this final class size SMean takes the arithmetic mean of all class sizes, while SMedian takes the median class size of those classes. If a class is larger than the final class size it would be randomly undersampled, otherwise random oversampling is applied to that class.

3.2 Experimental Setups

All experiments in this section utilize a basic SVM classifier with the Radial Basis Kernel. Furthermore, a 20-fold cross validation is performed on the training data to provide an evaluation measure which is the cross validation error rate.

Binary Datasets. A total of nine binary datasets are included in this initial resampling implementation experiment. Each dataset is a combination of one of the three failure types and one of the three normal types. The majority to minority ratio, or the class ratio, for each dataset are presented in Table 5. All nine binary sampling methods described in the previous subsections are applied to these datasets. Furthermore, every method and dataset combination is repeated ten times, yielding 90 cross validation error rates for each sampling method.

	Fail. 1	**Fail. 2**	**Fail. 3**
Norm. 1	2.6:1	2.8:1	2.6:1
Norm. 2	12.3:1	13.7:1	12.1:1
Norm. 3	2.6:1	2.8:1	2.6:1

Table 5: Class ratios of the binary datasets (Normal:Failure)

Multi Class Dataset. Three failure types and three normal types are all put together into a unified dataset. This 6-class dataset is then subjected to four different training setups: one is directly using the dataset for training and the other three incorporate balancing techniques. These techniques are the two multi-class sampling methods, SMean and SMedian, and a simple class weighting scheme. Equation 1 describes this weighting scheme, where c_i and n_i are the weight and size of class i respectively and k is the number of classes. The weights are normalized so that they all sum up to one. Each of these four SVM training setups is repeated a hundred times. One hundred cross validation error rates for every setups are used for comparison.

$$c_i = \frac{\sum_{i=1}^{k=6} n_i}{n_i} * (\sum_{i=1}^{k=6} \frac{\sum_{i=1}^{k=6} n_i}{n_i})^{-1} \tag{1}$$

3.3 Experimental Result Analysis

Both experiments are analysed with the same approach. It involves visual inspection of each sampling method's cross validation error rates distributions through boxplot

visualizations and statistical tests to compare the methods performance with each other. The Friedman test and post-hoc Nemenyi test are employed as suggested by [6]. These statistical tests are intended to tell if there are any significance difference between how each method ranks within the same iteration, across nine different datasets in the binary problem. Boxplots for the binary and multi class experiments are shown in Figure 2 and Figure 3 respectively. As Friedman tests on both experiment indicate statistical significance, important parts of the post-hoc Nemenyi tests are presented in Table 6 and Table 7 for the binary and multi class experiments respectively.

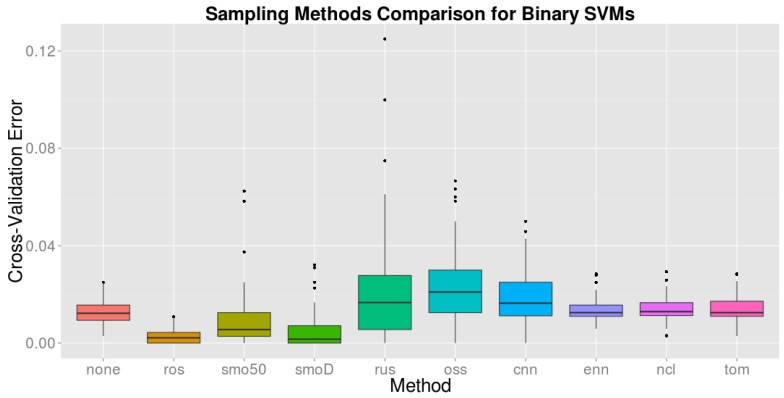


Figure 2: Sampling Methods for Binary SVMs. Each boxplot contains 90 observations, 10 observations for each binary datasets.

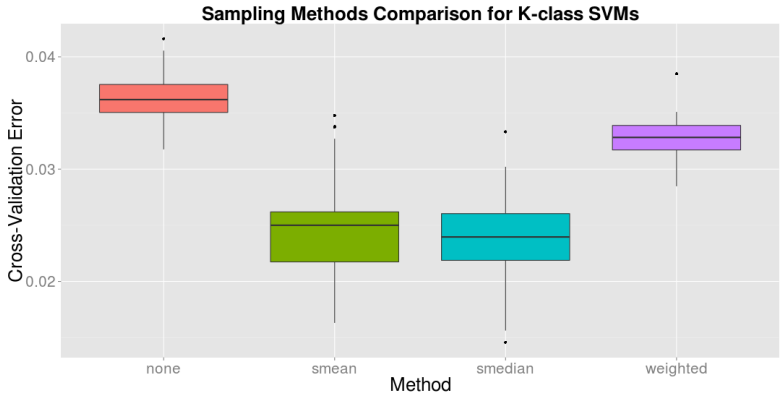


Figure 3: Sampling Methods for Multi Class SVMs. Each boxplot contains observations from 100 repeated runs on the 6-class dataset to train the SVM using the respective balancing method.

Binary Datasets. Table 6 suggests that all three oversampling based methods, the random oversampling and two SMOTE variants, performed significantly better than training without data balancing. However, Figure 2 provides further insight about the cross validation error rate distributions between those three sampling methods. Both random oversampling and the default SMOTE implementation (smoD) seem to have minimal overlap with training results on imbalanced datasets.

Method	p-value
Random Oversampling (ros)	$1.4 * 10^{-12}$
SMOTE A (smo50)	0.009
SMOTE B (smoD)	$6.6 * 10^{-11}$
Random Undersampling (rus)	0.318
One-Sided Sampling (oss)	$3.3 * 10^{-5}$
Condensed Nearest Neighbour (cnn)	0.048
Edited Nearest Neighbour (enn)	0.999
Neighbourhood Cleaning Rule (ncl)	0.594
Tomek link (tom)	0.697

Table 6: Statistical Significance of Sampling Methods for Binary Datasets. Reported p-value from Nemenyi test for the performances of nine sampling methods compared to training on the original binary datasets.

	none	**smean**	**smedian**
smean	$< 2 * 10^{-16}$	-	-
smedian	$< 2 * 10^{-16}$	0.96	-
weighted	$1.9 * 10^{-7}$	$2.7 * 10^{-13}$	$5.7 * 10^{-14}$

Table 7: Statistical Significance of Sampling Methods for Multi Class Datasets. Reported p-values from Nemenyi test for the performances of three balancing methods compared to training on the original multi class dataset (none).

Furthermore, SMOTE A which enforces majority class shrinkage (smo50) appears to vary more. A suggested cause here is that in SMOTE B (smoD) there are cases where the majority class are actually randomly oversampled instead of undersampled, potentially avoiding information loss from the undersampling step of SMOTE. The major take-away here is that in this particular case of data imbalance, inflating the size of minority classes, may lead to a significant improvement of SVM performance in terms of having low error rate. Undersampling, however, seems to cause important information loss of normal patterns.

Multi Class Dataset. Taking a look at Table 7, all four scenarios appear to be significantly different from each other except when comparing the two sampling methods SMean and SMedian. Moreover, by consulting Figure 3, all three data balancing approaches produced improved performances with sampling methods out performing the naive weighting scheme. As a definite sampling method is needed for the next section, a rather conservative choice of using SMean is taken. While both methods performed practically the same in terms of error rate, SMean provides a slightly smaller class size of 153 instances compared to SMedian's 160 and maintains roughly the same dataset size to that of the original's.

4 SVM Classifier

4.1 Choice of Kernel

Kernels make SVM able to map initial feature space into high dimensional one and find there decision boundary without additional computational overhead. Appropriate choice of such a transformation highly influence accuracy of classifier. It is an open question in almost any research in area of failure pattern recognition. There is no general rule for choosing a specific kernel. However in recent years there is a tendency to use Radial Basis Kernel as a benchmark.

In order to estimate the optimal transformation standard SVM classifiers are build on balanced training data with different kernels: Linear, Polynomial, Radial Basis, Laplacian and ANOVA Radial Basis Function (ANOVA RBF). Each kernel transformation was used with default parameters specified in function *ksvm* of package *kernlab*[5] built in R.

Kernel	**CV Error Rate (%)**	**Parameters**
Linear	1.96	—
Polynomial	2.07	$(d = 1, \alpha = 1, c = 1)$
Radial Basis	2.40	$(\sigma = 1)$
Laplacian	2.07	$(\sigma = 1)$
ANOVA RBF	1.42	$(\sigma = 1, d = 1)$

Table 8: Accuracy of Kernels. CV Error Rate represents 20-fold cross-validation error rate on balanced training dataset.

Table 8 shows that ANOVA RBF stably outperforms other kernels taking into account cross-validation error rate. However it has two parameters which have to be optimized later on. From this point of view Linear kernel seems to be reasonable to choose rather than ANOVA RBF, since it does not have parameters and brings second lowest error rate. Of course, it depends on analysis restrictions: if priority is given to accuracy then ANOVA RBF should be chosen, otherwise if time is prioritized then Linear kernel is the best transformation. In this analysis ANOVA RBF is used further as a base kernel transformation in assumption that accuracy is primary over time. Equation 2 represents this transformation.

$$k(x, y) = \sum_{k=1}^{n} exp\left(-\sigma\left(x^k - y^k\right)^2\right)^d \tag{2}$$

One can argue such a comparison since kernels are compared using default parameters but not the optimal ones. This fact can be explained in two ways. First, time and resource restrictions of analysis do not allow to find optimal parameters for every kernel, because it requires additional time and adequate computational resources. Second, it seems unrealistic that other kernels will outperform ANOVA RBF in terms of accuracy having relatively big gap in default settings.

[5]https://cran.r-project.org/web/packages/kernlab/kernlab.pdf

4.2 Dimensionality Reduction and Feature Selection

It is already explored in Section 2.2 that features are correlated that can potentially negatively affect classifier efficiency. One of methods to solve such a problem is to apply dimensionality reduction technique, it extracts only the optimal features and reduce the dimensionality. The idea behind that is to find independent subsets of features or combinations of features which explain the most variation in data. As number of dimensions is reduced accuracy increases and computing time can be shortened as well.

Many methods have been proposed to perform dimensionality reduction in failure pattern recognition problems. Most widespread methods are Principal Component Analysis (PCA) and Independent Component Analysis (ICA). These approaches and their nonlinear modifications were used, for instance, by Widodo et. al [29, 31] in faults diagnosis of induction motors. In this analysis PCA was used in order to reduce number of features. Using correlation matrix instead of covariance in computing principal components lead to 99% of variance explained by first 13 out of 21 principal components. However it does not mean what only first 13 principal components should be used. It is advisable in pattern recognition to use selection techniques which help to choose appropriate number of features without loses of important information. Here Genetic Algorithms (GA) and Distance Evaluation Techniques (DET) can be applied.

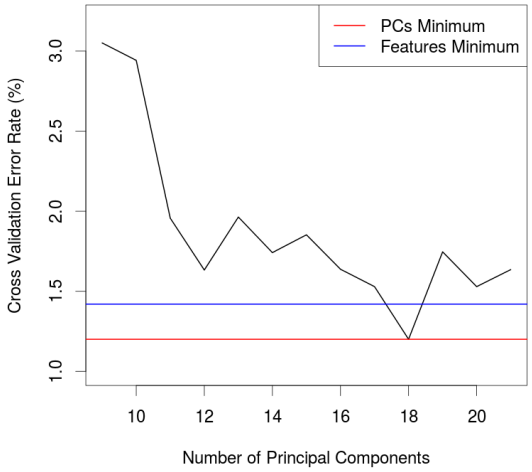


Figure 4: Effect of PCA. Black line represents error rate using a certain number of principal components. Red and blue lines show minimum error rates for fit using principal components and fit using extracted features, minimum error rates equal 1.42% and 1.2% respectively.

Another method to do feature selection, which is used in this analysis, is to iteratively fit classifiers for any reasonable number of principal components and track error rate over fits. Figure 4 shows that PCA improves not only error rate, which is decreased from 1.42% to 1.2%, but also removes unnecessary information since in optimal case only 18 out of 21 principal components are used. It is also reasonable to mention how fast error rate degrades at early stages. Starting from roughly a half of all components - 11 principal components, error rate is lower than 2% that makes predictions very accurate.

4.3 Multi Class SVM

Initially SVM technique was proposed as a binary classifier. However in this analysis underlying problem requires division between six classes: three normal and three failure ones. "One to Others" and "One Against One" modifications of SVM were proposed to deal with this limitation. In "One to Others" method there are basically two classes in each optimization process: one class and the others as the other class. The winner is defined as class with the maximum distance to decision boundary. "One Against One" approach uses $k(k-1)/2$ classifiers for k-class problem. Here each classifier is trained on data from two classes. After training process winner class can be deduced using voting scheme.

Instead of creating several binary classifiers, a more natural way is to distinguish all classes in one single optimization process. Initially Weston and Watkins proposed k-class SVM [28]. Further modification was proposed by Crammer and Singer [5]. These methods generalize optimization problem of binary case for k-classes, however final complexity impedes extensive use of these approaches in practice.

So far in this analysis "One Against One" approach was used. Model consisted of 15 binary SVM classifiers and the winner was deduced using voting procedure: class with the highest number of votes is a winner. However other approaches should be considered as well. Since "One to Others" approach can bring imbalance into an already balanced training data it is excluded from consideration. Table 9 represents comparison of different approaches.

Method	**Classifiers**	**CV Error Rate (%)**	**Elapsed Time (sec)**
"One Against One"	15	1.2	5.2
Crammer and Singer	1	2.28	6.1
Weston and Watkins	1	29.58	11.2

Table 9: Multi Class SVM. CV Error Rate represents 20-fold cross-validation error rate on balanced training dataset.

It can be seen from Table 9 that multi class modifications do not benefit computational performance and final accuracy of classifier at all. Moreover error rate using Weston and Watkins multi class method is dramatically higher. It can be possibly explained by the fact of inappropriate parametrization, since default parameters are used in experiments. In addition, computational time for this method more than twice longer than time to build 15 standard binary SVM classifiers.

4.4 Parameter Optimization

So far model was build under default parameters. More precisely, regularization strength parameter C of SVM classifier equals 1 and ANOVA RBF kernel has a vector of parameters $\alpha = (\sigma = 1, d = 1)$. Of course, proper parametrization is an open problem in failure pattern recognition. It requires time and expensive computational resources, however accuracy can significantly benefit from such optimization. In area of machine fault diagnosis a variety of methods were proposed to cope with that problem: GAs, Particle Swarm Optimization (PSO) and Grid Search (GS).

In this analysis in order to optimize parameters, PSO proposed by Kennedy, Eberhart and Shi [22] is used. This method is mainly based on behaviour of animals then they are trying to find the source of food. Optimization experiment used 25 iterations of algorithm with default settings specified for function *psoptim* of package *pso*[6] built in R.

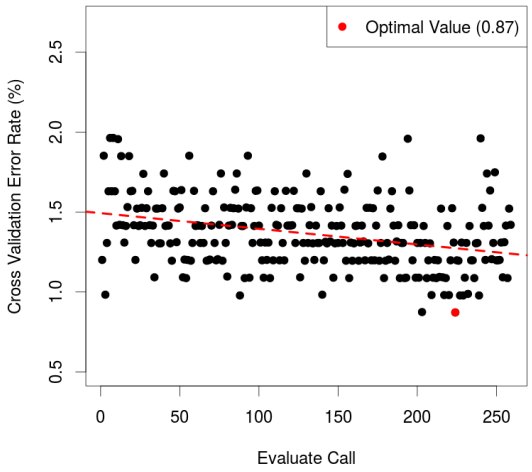


Figure 5: Parameter Optimization. Black points represent fitness of a single model evaluation, optimal fitness is 0.87% of 20-fold cross validation error rate.

Figure 5 represents search procedure of optimal parameters. It can be seen that general trend is decreasing over iterations. In the end of optimization, final 20-fold cross validation error decreased from 1.2% to 0.87%. The regularization strength parameter C changed from 1 to 20.65 that basically means that decision margin of classifier become thinner. Parameters of kernel changed as well, initial vector $\alpha = (\sigma = 1, d = 1)$ moved to $\alpha' = (\sigma = 0.77, d = 1.03)$. However optimization process was very computationally exhaustive and took about 72.5 minutes for 25 iterations, in comparison one fit lasts only 5.5 seconds.

Two remarks should be done here. First, found parameters are "pseudo" optimal since PSO is just a heuristic and does not provide global solutions. Second, estimated error rate is robust but can change, since division to folds in cross validation is stochastic. It can be fixed in two ways: increasing number of folds or setting up replication for evaluation of fitness. Replication means that error rate for a set of parameters will be calculated multiple times and then, for example, will be averaged. Both methods improve quality of optimization but increase complexity.

4.5 Final Model and Further Modifications

After all improvements and modifications final model incorporates "One Against One" approach as well as voting scheme to extend SVM for multi class problem. Therefore 15 standard "C" implementations of SVM with parameter $C = 20.65$ are build to distinguish between each pair of classes. Kernel transformation for every binary classifier is ANOVA RBF with a vector of parameters $\alpha = (\sigma = 0.77, d = 1.03)$.

[6]https://cran.r-project.org/web/packages/pso/pso.pdf

Model is fitted on first 18 principal components extracted from initial data. Training data are balanced with use of "SMean" technique explored in Section 3.1. Final 20-fold cross validation error rate equals 0.87% which makes model very accurate.

	Norm. 1	**Norm. 2**	**Norm. 3**	**Fail. 1**	**Fail. 2**	**Fail. 3**
Norm. 1	**229**	0	1	0	0	0
Norm. 2	0	**249**	1	0	0	0
Norm. 3	0	3	**228**	0	0	0
Fail. 1	0	0	0	**90**	0	0
Fail. 2	1	0	0	0	**19**	0
Fail. 3	2	0	0	0	0	**90**

Table 10: Confusion Matrix for Final Model. Norm. is short for Normal and Fail. is short for Failure respectively. Actual classes are arranged in columns, predicted classes in rows.

Table 10 shows final confusion matrix obtained testing model on initial imbalanced data. Overall accuracy of predictions is 99.1% with error rate 0.9% which is very close to obtained earlier cross validation error rate estimation of 0.87%. It is reasonable to mention that none of observations from failure classes were misclassified, however some observations from normal classes are recognized as failures. There is also some overlap between normal classes. These results were expected and can be explained that during balancing normal classes were undersampled and failure classes oversampled therefore model already seen all observations from failure classes and only a portion of observations from normal classes.

So far provided data were aggregated in time window of 5 seconds, it basically means what the answer for a question: "Under which conditions does system operate now?" can be answered in 5 seconds. However in some scenarios this response time is not enough. From one side, logically decrease of response time will negatively affect accuracy, since one observation is not that informative, but from the another side it will make model more sensitive to failures. Additionally, since time window is decreasing, size of training data is increasing and computational overhead may arise. These scenarios can be explored through simulation experiments using provided data.

Response Time (sec)	**CV Error Rate (%)**	**Elapsed Time (sec)**
0.5	4.97%	184.8
1	1.72%	60.6
3	0.66%	26
5	0.87%	21.5
10	0.81%	20.2
20	0.34%	17.5

Table 11: Affect of Response Time. CV Error Rate represents 20-fold cross-validation error rate on balanced training dataset. Elapsed time represents overall time to preprocess data, balance and fit classification model using SVM.

It can be seen from Table 11 that there is decreasing trend in error rate which

was expected and matches with logic. However there is one exclusion, accuracy for response time of 3 seconds is better than for 5 or even 10 seconds. It can be explained by the fact that error rate is stochastic and true error rate for that response time should be somewhere in range between 0.87% to 1.72%. To summarize this comparison it can be proposed to use response time equals 1 or 3 seconds. However, further decrease hardly can be done in practice since aggregation in time window of 0.5 second significantly affect computational time and accuracy.

One practical problem normally arises in area of failure pattern recognition. It is impossible to get all the failure data in advance: failures are not discovered yet or arriving very slowly as system is aimed to work without them. The straightforward way can be here is to wait for further failures and then retrain classifier incorporating arrived data. Of course, it is not advisory method in practice. Another way to cope with the problem is to use one-class SVM. This method was initially proposed by Scholkopf et al. [19]. The idea behind this method is to construct a representational model of normal training data. Then if newly arriving data are too different according to some measurements they labelled as out-of-class.

To explore performance of one-class SVM provided data are preprocessed and aggregated in time window of 5 seconds as before. Training data consist of 80% of observations from Normal 1 class, other normal classes were excluded as they were obtained under pressure conditions. Test data consist of the rest 20% from Normal 1 class and all observations from failure classes. Kernel transformation is ANOVA RBF with vector of parameters $\alpha = (\sigma = 1, d = 1)$

	Normal 1	**Failure 1**	**Failure 2**	**Failure 3**
Normal	**35**	3	1	9
Failure	12	**87**	**18**	**81**

Table 12: Confusion Matrix for One-Class SVM. Actual classes are arranged in columns, predicted classes in rows.

It can be seen from Table 12 that even without knowing actual failures one-class SVM can recognize them with high level of confidence. The final accuracy rate for that model is 89.8%. However about 25% of observations from class Normal 1 were misclassified, in practice it means that there will be a lot of false alarms. In spite of that fact, majority of failure observations were classified correctly which highlights applicability of one-class SVM.

5 Results and Conclusion

The first question that should be discussed here is feasibility of results. Final accuracy, obtained in this analysis, is extremely high and should be considered very cautiously. Good results can be explained by several reasons. First, crucial role here is purity of provided data. Data do not have outliers or missing values and can be even "artificial" since they were generated in test environment. Second, balancing techniques improve accuracy of predictions. Third, dimensionality reduction and feature selection as well as SVM tuning benefit results.

During the analysis following findings are explored. Removing steady states at data preprocessing stage improves further results. Aggregation of data in a cer-

tain time window is required and makes classification even possible. Also balancing mildly imbalanced data still can improve classification results. PCA slightly decreases needed amount of features and improves accuracy. This analysis shows that based on generated data ANOVA RBF transformation outperforms other kernels. Further optimization of classifier and kernel parameters can be done with help of PSO, but it is computationally expensive. Required response time can be decreased to 1 second, but it takes 3 times longer to train. From other hand, classifier, based on response time of 3 seconds, is slightly longer to build, but accuracy even better.

This analysis does not pretend to be exhaustive therefore a variety of improvements can be done in order to improve results. First, feature extraction can be extended, for example, frequency-domain features can be used to capture physical concept of a signal. This approach was explored by Soleimani et al. [23] and brought positive results. Second, at early stages more detailed preprocessing step can be done, for example data normalizing. Third, one can propose to apply more sophisticated k-class sampling methods. Fourth, more advanced dimensionality reduction techniques, such as ICA or nonlinear modifications of PCA and ICA, as well as advanced feature selection methods, for example, Distance Evaluation Techniques, can be applied. Another improvement is to try different methods of SVM, such as ν-SVM or ϵ-SVM, and different approaches of multi class SVM adaptations, for example"One-Against-Other" method. One can also propose to analyse relative importance of features, in other words find small subset of untransformed features which can be used to separate classes. This can be used in order to understand the causality effects of vibration on failure states. One technique that can be applied here is Recursive Feature Elimination (RFE). RFE is aimed to get a ranked list of features. Features are sorted relatively to contribution of SVM accuracy and therefore top of them can be used to solve above task. This method was applied by Tian et al. [25] in steel plates failure pattern recognition.

References

[1] Two modifications of cnn. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-6(11):769–772, Nov 1976.

[2] Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813, 2011.

[3] Hyeran Byun and Seong-Whan Lee. Applications of support vector machines for pattern recognition: A survey. In Seong-Whan Lee and Alessandro Verri, editors, *Pattern Recognition with Support Vector Machines*, volume 2388 of *Lecture Notes in Computer Science*, pages 213–236. Springer Berlin Heidelberg, 2002.

[4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[5] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[6] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, December 2006.

[7] S. Deng, Seng-Yi Lin, and We-Luan Chang. Application of multiclass support vector machines for fault diagnosis of field air defense gun. *Expert Systems with Applications*, 38(5):6007 – 6013, 2011.

[8] P. Hart. The condensed nearest neighbor rule (corresp.). *Information Theory, IEEE Transactions on*, 14(3):515–516, May 1968.

[9] Haibo He and E.A. Garcia. Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on*, 21(9):1263–1284, Sept 2009.

[10] Danilo S. Jodas, Norian Marranghello, Aledir S. Pereira, and Rodrigo C. Guido. Comparing support vector machines and artificial neural networks in the recognition of steering angle for driving of mobile robots through paths in plantations. *Procedia Computer Science*, 18(0):240 – 249, 2013. 2013 International Conference on Computational Science.

[11] J. Kamruzzaman and R.K. Begg. Support vector machines and other pattern recognition approaches to the diagnosis of cerebral palsy gait. *Biomedical Engineering, IEEE Transactions on*, 53(12):2479–2490, Dec 2006.

[12] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *In Proceedings of the Fourteenth International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann, 1997.

[13] Jorma Laurikkala. Improving identification of difficult small classes by balancing class distribution. In *Proceedings of the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine*, AIME '01, pages 63–66, London, UK, UK, 2001. Springer-Verlag.

[14] Victoria Lopez, Alberto Fernandez, Salvador GarcÃa, Vasile Palade, and Francisco Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113 – 141, 2013.

[15] AK Marnerides, S Malinowski, R Morla, and HS Kim. Fault diagnosis in dsl networks using support vector machines. *Computer Communications*, 62:72–84, 2015.

[16] K.K. McKee, G.L. Forbes, I. Mazhar, R. Entwistle, and I. Howard. A review of machinery diagnostics and prognostics implemented on a centrifugal pump. In Jay Lee, Jun Ni, Jagnathan Sarangapani, and Joseph Mathew, editors, *Engineering Asset Management 2011*, Lecture Notes in Mechanical Engineering, pages 593–614. Springer London, 2014.

[17] Song Pan, Serdar Iplikci, Kevin Warwick, and Tipu Z Aziz. Parkinsons disease tremor classification–a comparison between support vector machines and neural networks. *Expert Systems with Applications*, 39(12):10764–10771, 2012.

[18] B. Samanta. Gear fault detection using artificial neural networks and support vector machines with genetic algorithms. *Mechanical Systems and Signal Processing*, 18(3):625 – 644, 2004.

[19] Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support vector method for novelty detection. In *NIPS*, volume 12, pages 582–588, 1999.

[20] N. Seliya, Zhiwei Xu, and T.M. Khoshgoftaar. Addressing class imbalance in non-binary classification problems. In *Tools with Artificial Intelligence, 2008. ICTAI '08. 20th IEEE International Conference on*, volume 1, pages 460–466, Nov 2008.

[21] Yang Shao and Ross S. Lunetta. Comparison of support vector machine, neural network, and cart algorithms for the land-cover classification using limited training data points. *ISPRS Journal of Photogrammetry and Remote Sensing*, 70(0):78 – 87, 2012.

[22] Yuhui Shi and Russell C Eberhart. Empirical study of particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3. IEEE, 1999.

[23] Ali Soleimani, Mohammad J Mahjoob, and Masoud Shariatpanahi. Fault classification in gears using support vector machines (svms) and signal processing. In *Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, 2009. ICSCCW 2009. Fifth International Conference on*, pages 1–4. IEEE, 2009.

[24] YANMIN SUN, ANDREW K. C. WONG, and MOHAMED S. KAMEL. Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04):687–719, 2009.

[25] Yang Tian, Mengyu Fu, and Fang Wu. Steel plates fault diagnosis on the basis of support vector machines. *Neurocomputing*, 151, Part 1(0):296 – 303, 2015.

[26] Vladimir N Vapnik. The nature of statistical learning theory. 1995.

[27] Shuo Wang and Xin Yao. Multiclass imbalance problems: Analysis and potential solutions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(4):1119–1130, Aug 2012.

[28] Jason Weston and Chris Watkins. Multi-class support vector machines. Technical report, Citeseer, 1998.

[29] Achmad Widodo and Bo-Suk Yang. Application of nonlinear feature extraction and support vector machines for fault diagnosis of induction motors. *Expert Systems with Applications*, 33(1):241–250, 2007.

[30] Achmad Widodo and Bo-Suk Yang. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21(6):2560 – 2574, 2007.

[31] Achmad Widodo, Bo-Suk Yang, and Tian Han. Combination of independent component analysis and support vector machines for intelligent faults diagnosis of induction motors. *Expert Systems with Applications*, 32(2):299 – 312, 2007.

[32] Dennis L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-2(3):408–421, July 1972.

Declaration of Authorship

We hereby declare that, to the best of our knowledge and belief, this Seminar Thesis titled "Application of Data Analytics in Failure Pattern Recognition" is our own work. We confirm that each significant contribution to and quotation in this thesis that originates from the work or works of others is indicated by proper use of citation and references.

Münster, August 1, 2015