Western Australian School of Mines

Characterisation of Dynamic Process Systems by Use of Recurrence Texture Analysis

Jason Bardinas

This thesis is presented for the Degree of

Doctor of Philosophy

of

Curtin University

May 2018

DECLARATION

To the best of my knowledge and belief, this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Jason Bardinas

07/05/2018

Date

ABSTRACT

Many dynamic processes exhibit recurrent behaviour. This is an important feature that led to the idea that the recurrence of the states is a key element in the comprehensive understanding of dynamical systems. Although the recurrence has been known for a long time, its prevalence has just seen recently due to the advancement of computational technology especially in the visualisation of the recurrence of the states.

The recurrence plot is the most common method in studying the recurrence, which are often quantified by using Recurrence Quantification Analysis. Although proven effective, these methods only work when the distance matrix is compared to the threshold value, a significant parameter that can only be defined by the user. This threshold value actually dictate the features of the recurrence inside the plot. Although the so-called 'unthresholded recurrence plot' which is a mere distance matrix has already been proposed, its use is still limited due to the absence of reliable method that could extract useful features of the recurrence of the states from these plots. This research gap has been the motivation in the development of the proposed method which is termed as "Recurrence Texture Analysis".

Recurrence Texture Analysis was stimulated in the hypothesis that the recurrence plots of dynamical systems, whether thresholded or not, are governed by a well-defined texture descriptors that can be extracted by some algorithms used for texture analysis. The validation of this premise and the evaluation of the method, along with the exploration of the use of deep neural networks and other texture extraction algorithms to quantify textures and hence the behaviour or dynamics of the time series data, were accomplished in this research. More specifically, the applicability of the method to capture the structures of complex dynamic process systems represented by time-series and characterise its dynamic behaviour were carried out systematically. These were done by transforming the recurrence patterns into sets of feature matrices containing the texture features. Conversely, the structures of time series are captured via the extraction of variables representing the distances between measurements over time. This is achieved through analysis of the structure of the data during discrete periods of process observation. These joint distributions of the elements in these distance matrices are subsequently captured by descriptors or features most often used in the analysis of textures from image data.

Six texture algorithms are considered in this research including the two state-of-the-art pretrained deep convolutional neural networks. The texture features extracted using grey-level co-occurrence matrix, wavelet transforms, local binary pattern, textons, CNN-AlexNet and CNN-VGG16 were analysed via cluster and classification analysis of the features. To verify the robustness of the method in dealing real process datasets, RTA was also applied in some datasets of grinding circuits and powder flow to characterise their dynamic behaviour. Furthermore, the method was also used as a data pre-processing technique in time-series classification task. The method was also extended as a core in the development of statistical process monitoring system.

Over the course of this research, RTA showed its applicability in capturing the textural features of recurrence that are relevant in the understanding of the dynamical behaviour of dynamic process systems. It has showed substantial advantage in the characterisation of the complex and dynamic properties of several processes over its counterparts. Specifically, the VGG16 provided that most discriminative textural features as demonstrated by its outstanding performance, outperforming other RTA methods and even to other published methods. Throughout this thesis, it consistently achieved highly competitive results against any other considered similar approaches. Furthermore, AlexNet and texton also displayed reliable performance in which they satisfactorily performed well over GLCM, wavelet and LBP and even to RQA. With this, it can also be said that these are also good candidates for capturing the structures of process systems.

It is also demonstrated in this thesis the applicability and effectiveness of pretrained CNNs (VGG16 and AlexNet) in capturing the dynamic behaviour of time series data eventhough the distance matrix plots are not included in the training stage of these algorithms. This further confirmed that these algorithms can be extended to other applications and used to other types of datasets. Most importantly, the thesis clearly showed the promising application and the pioneering work of recurrence texture analysis to minerals processing including grinding circuits, which would open more researches with the method and other related methods for minerals engineering application. In this work, it proved that the method could form the basis for more development of models for applications in minerals processing, which could in principle be used and implemented online once calibrated.

NOMENCLATURE

LIST OF ABBREVIATIONS AND ACRONYMS

2D	two-dimensional
3D	three-dimensional
AANN	Autoassociative neural network
ACF	Autocorrelation function
AIC	Akaike information criterion
AMI	Average Mutual Information
ANN	Artificial neural network
AUC	Area-under-the-curve
CF	Cake Flour
CNN	Convolutional neural network
CON	Contrast
COR	Correlation
CPV	Cumulative Percent Variance
CWT	Continuous Wavelet Transforms
DPCA	Dynamic Principal Component Analysis
DWT	Discrete Wavelet Transforms
EMI	Eigenvalue-based Mutual Information
ENE	Energy
ENTR	Entropy
EOG	Electrooculogram
FAR	False alarm rate
FD	Feed Disturbance
FL	Feed Limited
FNN	False Nearest Neighbour
GAF	Gramian Angular Field

GADF	Gramian Angular Difference Field
GASF	Gramian Angular Summation Field
GLCM	Grey Level Co-occurrence Matrix
GMM	Gaussian Mixture Model
HOM	Homogeneity
ILSCRC	ImageNet Large-Scale Visual Recognition Challenge
k-NN	k-Nearest Neighbour
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LM	Leung-Malik filter bank
LoG	Laplacian of Gaussian
LOI	Line of Interest
LTM	Lagged Trajectory Matrix
LVPP	Lotka-Volterra Predator-prey System
MAE	Mean-absolute-error
MAR	Missing Alarm Rate
Max	Maximum
MF	Maize Flour
Min	Minimum
MISSQ	Minimal Increase of Sum-of-squares
MRP	Multi-level Recurrence Plot
MSE	Mean-square-error
MTF	Markov Transition Field
NLPCA	Nonlinear Principal Component Analysis
NOC	Normal Operating Condition
PC	Portland Cement
PCA	Principal Component Analysis
PLS	Partial Least Squares

Q	Quartz
R^2	Coefficient of Determination
RF	Random Forest
RFS	Root Filter Set
RI	Rotation Invariant
RL	Run Length
RMSE	Root-mean-square-error
ROC	Receiving Operating Characteristics
RP	Recurrence Plot
RPCD	Recurrence Patterns Compression Distance
RQA	Recurrence Quantification Analysis
RR	Recurrence Rate
RTA	Recurrence Texture Analysis
SFTA	Segmentation-based Fractal Texture Analysis
SOM	Self-organising Maps
SSA	Singular Spectrum Analysis
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TE	Tennessee Eastman Problem
TFRP	Texture features from Recurrence Patterns
TS	Table Salt
UTRP	Unthresholded Recurrence Plot

LIST OF SYMBOLS

a, b, c	Scaled time ratios of the autocatalytic reaction
b	Window width
B	Bias term
C	Clustering coefficient

$\boldsymbol{c}$	Convolutional output (subscript: H for horizontal, A for approximation, D for diagonal, and V for vertical)
c	Number of dimensions (column) in time series $\boldsymbol{Y}$
C_c	Cophenetic correlation function
$\hat{C}$	Reaction rate or the frequency of the contact of the prey and predator
Cp	Controlling parameter
d	Distance
$\boldsymbol{D}$	Distance matrix
$\widehat{D}$	Conversion efficiency or the efficiency of predators in converting food into offspring
db4	Daubechies wavelet family
E	Error in Sammon map
$\boldsymbol{E}$	Residual matrix
F	Number of features
f	Lower dimensional latent variable in LDA
$F_0(\dot{t},\sigma)$	Zero direct current component guarantor
g	Image pixel (subscript: c for center, p for neighbouring pixel)
G	Number of grey level
h	Histogram bin
I	Image
I_G	Greyscale image
I_{sc}	Greyscale image (scaled)
J	All-ones matrix
K	Predator-prey rate (subscript: 1 for prey population growth, 2 for predator mortality)
k	Number of principal component retained

$K\ (x_i, x_j)$	Kernel function
L	Diagonal line length
$\boldsymbol{L}$	Segmented time series
L	Primal Lagrangian
L_2- norm	Euclidean norm
L_∞-norm	Maximum or Supremum norm
m	Sliding step
$\boldsymbol{m}_d$	Embedding dimension
n	Number of observations
N	Number of segments (distance matrix)
N_n	Average number of neighbours
N_P	Number of significant peaks in density plot
$\boldsymbol{P}$	Loading matrix
$p(\bullet)$	Probability function
$p(\bullet,\bullet)$	Joint probability function
$p(x\|\theta_i)$	Probability density function
$\hat{p}_{i,j}$	Pixel pair entry in (i,j)
$\hat{p}(\boldsymbol{z}\|\omega_k)$	Conditional density of the number of samples N_k within the class ω_k
Q-res	Q-residuals
r	Number of classes
$\mathbf{R}$	Recurrence matrix
$R(\boldsymbol{z})$	Hypersphere with volume $V(z)$ with $\boldsymbol{z}$ as the center of $R(\boldsymbol{z})$
$\boldsymbol{R}_{i,j}\ (unthres)$	Unthresholded recurrence plot
s	Scale parameter of the signal
$\boldsymbol{s}$	LBP operator threshold value
S	Schmid Filter Set

T	Texton channel
$\boldsymbol{T}$	Score matrix in PCA
T_c	Transitivity coefficient
T_n	Recurrence time
T^2	Hotelling’s T-squared
$\dot{t}$	Cycle count in the harmonic function
u	Discrete time series
U	Uniformity measure
$\bar{u}$	Mean of time series
V	Vertical line
W	Transformed function
w	Scaling filter (subscript: **H** for high, **L** for low)
$\dot{\boldsymbol{w}}$	LDA transformation
x, y, z	Feed concentration variables involved in autocatalytic reaction
$\boldsymbol{X}$	Feature matrix or data matrix
$\dot{x}$	Number of prey
$\vec{x}$	Trajectory point
$\mathbf{Y}$	Parent time series
$\dot{y}$	Number of predator
γ	Scaled feed concentration ratios
$\boldsymbol{\sigma}$	Gaussian noise
σ^2	Standard deviation
$\boldsymbol{\varepsilon}$	Threshold distance or neighbourhood size
θ	Angle of grey levels
$\boldsymbol{\Theta}$	Heaveside function
λ	Eigenvalues
$\boldsymbol{\tau}$	Time delay

τ	Translation parameter
ξ_i	Slack variable in SVM
ψ	Wavelet
$\psi\ (\bullet)$	Pre-defined function mapping
$\parallel \bullet \parallel$	Norm in the phase space

ACKNOWLEDGMENTS

My PhD journey is indeed a roller-coaster ride with many ups and downs. But as a wise man once said, "There is always light at the end of the tunnel". Of course, the journey towards this tunnel is made up of several challenges in many forms, but these challenges have transcended into triumphs because of the people mentioned here who supported me along the way.

First of all, I would like to express my deepest gratitude to my supervisors, Prof. Chris Aldrich and Dr. Boris Albijanic, for their guidance throughout this journey. I have learned more about the world of data, and the world in general, from them in the past three years of my life than I would ever have imagined was possible. Special mention to Chris for his patience he has given on me during the times when the light of the tunnel is nowhere to be found.

I extend many thanks to WASM and to Curtin International Postgraduate Research Scholarship (CIPRS) for the financial support and scholarships. Without this grant, this PhD journey would most likely not become possible.

I would also like to mention the support of my friends who have supported me during the periods where I felt so down and empty. Special shout out to France for the help and support especially in producing this manuscript.

Most especially, I would like to give my deepest gratitude to my family, particularly to my loving parents. Thank you for their encouraging words of wisdom and for always believing in me despite everything I have been through. Ultimately, I would like to thank the Almighty God in heaven. Thank you for the gifts of life and the power of believing in my works and in myself in general. I could never have this PhD without the faith I have in You.

PUBLICATIONS

This thesis includes the following works that have been submitted for publication over the course of my PhD study:

Bardinas, J., Aldrich, C., & Albijanic, B. (2016). Identification of the dynamic behaviour of an autogenous mill by use of time series cluster analysis. *Proceedings of the XXVIII International Mineral Processing Congress (IMPC 2016)*. ISBN: 978-1-926872-29-2, September 11 – 15, 2016, Québec City, Canadian Institute of Mining, Metallurgy and Petroleum. **[Chapter 7]**

Bardinas, J., Aldrich, C., & Napier, L. (2018). Predicting the operating states of grinding circuits by use of recurrence texture analysis of time series data. *Processes* 6, 17; doi: 10.3390/pr6020017 **[Chapter 7]**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Complexity in Dynamical Systems

Many natural and man-made systems show complex and nonlinear behaviour. In process engineering, this includes grinding circuits (Aldrich, Burchell, de V. Groenewald, & Yzelle, 2014), fluidized beds (Johnsson, Zijerveld, Schouten, van Den Bleek, & Leckner, 2000; Llop, Gascons, & Llauró, 2015; Pain, Mansoorzadeh, Gomes, & de Oliveira, 2002), multiphase flow (Keska, Smith, & Williams, 1999), multistage evaporation of liquor in the Bayer process used for alumina production by Alcoa (Kam, 2000) etc. – a cultivating idea that encouraged the emergence of the concept of complex systems, an area that is concerned with the understanding of the complexity and properties of the systems. Atay (2010) describes this subject as a study of any process or system that is comprised of numerous parts that produce macroscopic behaviour with a manifestation of forming distinctive temporal, spatial or functional structures.

While complex systems can be considered through different metaphors, such as bio-inspired algorithms (Cotta & Schaefer, 2017), cellular automata (Kotyrba, Volna, & Bujok, 2015), complex network theory (Mei, Zarrabi, Lees, & Sloot, 2015), chaos theory (e.g. (Sivakumar, 2004)) and others, the nonlinear time series analysis theory developed over the last four decades (e.g. (Kantz, 2004; Sprott, 2003)) is arguably one of the most important.

1.2 Phase Space Approach to Time Series Analysis

The basic point of departure in nonlinear time series analysis is that it contains repetitive patterns that can be analysed in a so-called phase space, which is simply a projection of the observed time series data onto a new coordinate system consisting of number of lagged time series variables. That is,

$$y(t) \rightarrow [x(t), x(t-k), \dots x(t-(m-1)k)] \tag{1-1}$$

Two parameters define the projection, namely the time series lag, k and the embedding dimension, m of the time series. Most of the time, the time series lag k is estimated by either the use of autocorrelation function of the average mutual information (Jemwa, 2003). The embedding dimension, m, on the other hand, can be assessed by several methods including singular value decomposition, and false nearest neighbours.

The repetition of patterns in the time series data are often referred to as "recurrence", which is considered a fundamental property in many dynamical systems. The recurrence is commonly visualised using recurrence plots, which are derived from a phase space embedding of the time series data, which are constructed based on distance matrix. Although embedding of the time series data is commonly done prior to construction of recurrence plots, this is not strictly necessary, as was shown by Iwanski and Bradley (1998) and March, Chapman, and Dendy (2005). In general, the recurrence point observed in the plot is a distance matrix element having a distance less than or equal to a specified neighbourhood size or threshold value, ε. This plot generally gives recurrence patterns, which are normally quantified using recurrence quantification analysis (RQA).

Although RQA is a highly versatile tool and has seen considerable success in various applications, (Hou, Aldrich, Lepkova, Machuca, & Kinsella, 2016; Souza, Silva, & Batista, 2014; Webber Jr, Marwan, Facchini, & Giuliani, 2009), it suffers from two drawbacks when used as a means to generate features of predictors for generalized time series analysis. First, thresholding the plot results in a loss of information, which may be important in subsequent analysis. Second, powerful as they are, RQA features are essentially hand crafted and may not be the optimal features required for a given analysis.

One of the logical thoughts in addressing these limitations in RQA is to eliminate the incorporation of the threshold value in constructing recurrence plots. Although the use of so-called "unthresholded recurrence plots (UTRP)" has recently been proposed (Sipers, Borm, & Peeters, 2011, 2017), the utilisation and full implementation is limited as yet, as there is still no well-established quantification method that can be employed to describe the features of the recurrences contained within these plots In fact, some researchers like Acuña-González, García-Ochoa, and González-Sánchez (2008) for example, mention the lack of analytical means explicitly as a reason for not using unthresholded recurrence plots. This challenge has motivated the development of the proposed method in this thesis.

1.3 Recurrence Texture Analysis: A Novel Method in Characterising the Recurrence

One could say the UTRP is basically just a distance matrix since each element is not compared to a threshold value nor transformed into binary form. In fact, the idea of UTRP has been known for quite some time, especially in the context of nonlinear data analysis.

UTRP or distance matrix plots commonly provide noticeable elements even to the small variations in the distances between the recurrence points (Savari et al., 2017) that sometimes are not reflected in the thresholded recurrence plots. While this becomes a significant advantage of UTRP, this is not commonly exploited, mainly due to its difficulty in quantifying the recurrence features contained in the plot. There is at present no established method that can be employed to extract useful features in UTRPs, since RQA is only applicable to thresholded RPs.

A basic idea that motivated the development of this thesis is that the recurrence plot, whether thresholded or not, can also be viewed as a texture, the nature of which is determined by the recurrence or repetition of patterns in the time series data. This means that a time series, as represented by a distance matrix, can be characterised by examining the texture of recurrences. For it to make sense, these textures could be extracted for further analysis, which could be possible through the use of several texture modelling techniques.

Texture modelling, as commonly used in image analysis and computer vision, is considered a novel approach on studying dynamic systems. Literature reviews suggested that the incorporation of texture in analysing the distance matrix of the time series, and hence of the structure of dynamical systems is not well established yet.

Moreover, the encoding of time series into images allows one to take advantage of recent developments in state-of-the-art algorithms in image analysis. This includes, in particular, the deep convolutional neural networks that are currently enjoying massive interest among researchers and developers, owing to their proven ability to quantify discriminative textures are also considered an innovative approach for time series analysis.

The use of recurrence texture analysis explored in this thesis therefore aims to take advantage of the fact that the behaviour of time series data can be expressed as textures in distance matrices derived from these data. In tandem with this, it also aims to explore the use of deep neural networks to quantify these textures and hence the behaviour or dynamics of the time series data.

1.4 Objectives

The general aim of this study is to evaluate the applicability of the recurrence texture analysis in characterising dynamic process systems and subsequent analysis of their dynamic behaviour.

This main objective will be achieved by the accomplishment of the following tasks:

- The analysis of the method and its performance in analysing time series by:
    - extraction of texture features of distance matrix plots using several textural extraction algorithms, including those based on the use of pretrained deep convolutional neural networks
    - evaluation of the influence of parameters of the methodology, specifically the window length and type of distance metric used on the overall performance of the method.
    - comparison of results to other approaches, particularly the use of RQA features, from which this approach is derived.
- The application of the method in dealing with time series analysis typically associated with applications in the process industries, namely:
    - time series classification of benchmark data sets
    - application of the proposed method to real datasets such as in grinding circuits and solids processing time series data
- Assessment of the performance of dynamic process monitoring systems derived from the method, via:
    - application of the method on simulated data and a benchmark process engineering dataset

- comparison of the results to other related approaches.

1.5 Scope

The thesis, as a whole, is a proof-of-concept study of the proposed method, which is mostly concentrated on the evaluation of the method as a viable tool in analysing the structures of dynamical systems. In particular, although other benchmark datasets are used, the thesis is comprehensively focused on the evaluation of the method to dynamic process systems.

This thesis involves applying the method to time series classification employing RTA as data preprocessing method, and dynamic process monitoring or fault detection, using actual data sets, simulated and benchmarked datasets and public data sets. The thesis primarily focuses on applications to minerals engineering time series and consequently on the understanding of the nonlinear and dynamic behaviour of minerals processing datasets, such as in grinding circuit. In terms of texture modelling, the algorithms used to extract textures are limited to texture classification algorithms, hence, this specifically excludes spectral feature extraction and regression.

Ultimately, this thesis does not constitute an end-product that is all set for application in a real scheme. Rather, this work contributes towards the long-term goal of developing more effective methods that deal with nonlinear and dynamic behaviour of systems.

1.6 Outline of Thesis

This thesis is organised as follows: In Chapter 2, literature review on the unthresholded and thresholded recurrence plots and the time series analysis based on these plots are presented. Following this, in Chapter 3, the analytical methodology is formalised and explained, including the general approach and the texture analytical algorithms considered in the investigation. Subsequent chapters deal with the application of the methodology, i.e. Chapter 4 details the results of the preliminary study of the method using simulated time series. The influence of parameters involved in employing the proposed method are studied in this chapter and the results are compared to RQA. In Chapter 5, the method is applied to time series classification on publicly available and simulated time series data sets. In Chapter 6, the dynamic behaviour of powder flow and autogenous mills are characterised using the method. In Chapter 7, the method is used as a framework for dynamic process monitoring based on the use of principal component models. Finally, Chapter 8 closes with the most important conclusions and recommendations from the study.

2. RECURRENCE IN DYNAMICAL SYSTEMS

2.1 Recurrence Plots

The recurrence plot (RP) is a graphical representation of recurrent patterns in a time series. First introduced by Eckmann (1987), the RP can visually describe the recurrence characteristic of many dynamical systems. It is a two-dimensional squared binary matrix wherein both axes represent time which is based on the distance measured. Formally, the recurrence matrix **R** of state $\vec{x}$ can be obtained using two time i and j along a phase space trajectory such that:

$$\boldsymbol{R_{i,j}} = \boldsymbol{\Theta}(\boldsymbol{\varepsilon} - \|\vec{\boldsymbol{x}}_{\boldsymbol{i}} - \vec{\boldsymbol{x}}_{\boldsymbol{j}}\|), \qquad i, j = 1, 2, 3, \ldots N \qquad (2\text{-}1)$$

where N is number of states under consideration, $\boldsymbol{\varepsilon}$ is the threshold distance, $\|\cdot\|$ is the norm in the phase space and $\boldsymbol{\Theta}$ is the Heaviside step function (Marwan, Kurths, & Foerster, 2015; Schultz, Spiegel, Marwan, & Albayrak, 2015). Since the recurrence matrix **R** is a binarised matrix, the RP is then plot using two different colours (i.e. black and white). That is, at coordinates *(i,j)*, if $\boldsymbol{R_{i,j}} = 1$, a black dot is drawn as opposed to a white dot if $\boldsymbol{R_{i,j}} = 0$. Further, the RP always contains a black main diagonal line that correspond to $\boldsymbol{R}_{i,j} = 1|_{i=1}^{N}$. The diagonal line is commonly referred to as the line of identity (LOI). It is important to note that RP generally exhibits symmetry with respect to LOI ($\boldsymbol{R}_{i,j} = \boldsymbol{R}_{j,i}$). In some cases, although it is not required, a phase space reconstruction is carried out in constructing RPs, which requires embedding the data to other phase space using embedding parameters (e.g. time lag, embedding dimension).

Based on eqn (2-1) , RP is dependent on the type of norm $\|\cdot\|$ and threshold distance $\boldsymbol{\varepsilon}$. These factors and parameters should always be taken into account when generating RP. In the next subsections, the emphasis on the effect of these parameters are discussed in detail.

2.1.1 The Choice of Norm

From a theoretical point of view of RPs, the choice of norm is deemed to be insignificant. However, it is not the case when it comes to practical purposes as the visual characteristic of RPs could change for different norms (Bradley & Mantilla, 2002). This is attributed to the fact that the choice of norm ($\|\cdot\|$) dictates its structure brought about by the different shapes of neighbourhood.

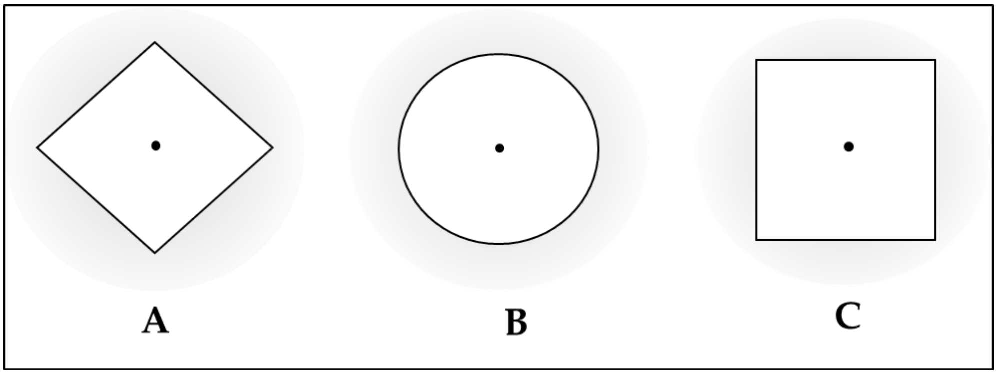


Figure 2-1. Three norms for the neighbourhood with same radius: L_1-norm (A), L_2-norm (B), L_∞-norm (C).

In the comprehensive review of RP made by Marwan, Carmen, Thiel, and Kurths (2007), three commonly used norms are considered. These are the L1- norm, L2- norm (more commonly known as Euclidean norm) and the L∞-norm (also known as the Maximum or Supremum norm). As an example, for fixed ε, these norms generate different shapes of neighbourhood. The L∞-norm gave the most number of neighbours, followed by the Euclidean norm (L2- norm), and then L1-norm. With this outcome, it is emphasised that L∞-norm is the best option in constructing RP, mostly because of its computational efficiency. At some point, the Euclidean norm was also seen as a good norm in many studies involving RPs (Hou et al., 2016; Javorka, Turianikova, Tonhajzerova, Javorka, & Baumert, 2009; Zbilut, Zaldivar-Comenges, & Strozzi, 2002).

2.1.2 The Choice of Threshold Distance

The neighbourhood size, more commonly known as the threshold distance, $\boldsymbol{\varepsilon}$ is another parameter of consideration when constructing RPs. A good value of $\boldsymbol{\varepsilon}$ has the capability of retaining more unique dynamically useful information while minimising the redundant information that could trigger misinterpretation of the features of recurrence (Schultz, Zou, Marwan, & Turvey, 2011). Normally, this is a user pre-defined parameter.

The structures of RPs generally depend on how large or small the chosen $\boldsymbol{\varepsilon}$ is. For instance, if $\boldsymbol{\varepsilon}$ is too small, recurrence structure becomes futile since no recurrence points are observed. On the contrary, $\boldsymbol{\varepsilon}$ being too large would result to a lot of artefacts and noise since the space considered is unreasonably vase such that every point is considered a neighbour of every other points (Marwan et al., 2007). With this, careful consideration is of utmost importance in identifying the optimal value of $\boldsymbol{\varepsilon}$.

A number of researchers have dedicated their efforts to studying this parameter and some of them established several "rules of thumb" in selecting $\boldsymbol{\varepsilon}$. For most studies, the $\boldsymbol{\varepsilon}$ is selected based on known information obtained from RP and phase space. This can be grouped into three: use of phase space diameter, use of the standard deviation of the noise present, and use of RP structures (i.e. recurrence point density, diagonal line).

For instance, in the work of Mindlin and Gilmore (1992), they used an estimate of $\boldsymbol{\varepsilon}$ by determining a few percent of the diameter of the attractor using its minimum and maximum value ($\boldsymbol{\varepsilon \sim 10^{-2} x \{max[x(i)] - min[x\{i)]\}}$**).** Similarly, Zbilut and Webber (1992) used a small value of $\boldsymbol{\varepsilon}$

relative to the noise level. In addition, they noted that the value is generally not be greater than 10% of the normalised mean phase space diameter. In the work of Schinkel, Dimigen, and Marwan (2008), they inferred that, in the context of signal classification and discrimination, the most acceptable threshold value $\boldsymbol{\varepsilon}$ is about 5% of the maximum phase space diameter.

In the study of Thiel et al. (2002) on the effect of observational noise on the RPs, they found out that noise could significantly change the features and properties of RPs, thus observational noise should be taken into account when constructing RPs, particularly in threshold value. They proposed that $\boldsymbol{\varepsilon}$ should be at least five times the standard deviation of the observational Gaussian noise, σ ($\boldsymbol{\varepsilon} > \mathbf{5}\boldsymbol{\sigma}$).

The $\boldsymbol{\varepsilon}$ can also be estimated using features obtained in RPs. For instance, the use of recurrence point density could approximate the value of $\boldsymbol{\varepsilon}$**.** A good value of $\boldsymbol{\varepsilon}$ is obtained if the recurrence point density is about 1% (Zbilut et al., 2002). For (quasi-)periodic processes, the information on the diagonal structures of RP can also be used to estimate $\boldsymbol{\varepsilon}$**.** (Marwan et al., 2007) claimed that a good value is $\boldsymbol{\varepsilon}$ is one that could minimize the quantity $\beta(\varepsilon)$.

$$\beta(\varepsilon) = \frac{|N_n(\varepsilon) - N_p(\varepsilon)|}{N_n(\varepsilon)} \tag{2-2}$$

where N_p is the number of significant peaks in a certain density plots, N_n is the average number of neighbours. In other words, $\boldsymbol{\varepsilon}$ is said to be optimised when N_p is maximised and N_n approaches N_p. While this estimation works well especially for de-noising applications, the significant distribution of the diagonal lines in RP could be compromised if observational noise is present in the signal. To address this, the use of fixed recurrence point density and the use of fixed number of neighbours for every point were proposed. With the fixed recurrence point density, more information is preserved, which allows for comparison of RPs even without undertaking time series normalisation prior to analysis.

Overall, even though there are already methods and guidelines available in determining $\boldsymbol{\varepsilon}$**,** there is still no strict rule that governs in determining optimal $\boldsymbol{\varepsilon}$**.** It is still largely dependent on the user/s and the type of systems being considered. In this sense, $\boldsymbol{\varepsilon}$ is a drawback in RPs, and therefore eliminating the use of this value is a good topic to research on as it has not been widely explored yet.

2.2 Unthresholded Recurrence Plots

The so-called "unthresholded RPs" is one of the variations of RPs. Generally speaking, it is referred to the distance plot since these RPs does not require any threshold distances and did not undergo binarisation (using Heaviside function) (Iwanski & Bradley, 1998). To some, it is also referred to as global recurrence plots (Webber Jr & Zbilut, 2003). Formally, the unthresholded RPs $\boldsymbol{R}_{i,j}\ (unthres)$ is determined using eqn (2-3), which is identical to the equation used in calculating distance matrix.

$$\boldsymbol{R}_{i,j}(unthres) = \left\|\vec{\boldsymbol{x}}_{\boldsymbol{i}} - \vec{\boldsymbol{x}}_{\boldsymbol{j}}\right\| \quad for\ i = 1, 2, 3 \dots N \qquad (2\text{-}3)$$

where || · || is the norm in the phase space.

Some researchers have directed their attention to the examination of the unthresholded RPs on the basis of possessing and providing as much information and explanation on the mechanism of the considered dynamic system (or signal) in diverse areas, including the interpretation of financial time series (Addo, Billio, & Guégan, 2013), unemployment data (Caraiani & Haven, 2013; W.-S. Chen, 2011), electrochemical signals (Acuña-González et al., 2008; Cazares-Ibáñez, Vázquez-Coutiño, & García-Ochoa, 2005), simulated stochastic signals (Rohde, Nichols, Dissinger, & Bucholtz, 2008) and the monitoring of liquid sprayed spouted beds (Savari et al., 2017). In these studies, interpretation of the unthresholded recurrence plots or distance plots of the time series data was based on visual inspection of the plots. Sipers et al. (2017) conducted a more analytical approach, wherein they studied the information content of the unthresholded recurrence plots (Sipers et al., 2011), as well as investigating the variation of the information when these plots are changed or distorted (Sipers et al., 2017). They have concluded that the information of the parent signal can always be represented by an unthresholded RPs up to an affine isometry.

Moreover, the choice of embedding parameters, and the amount of frequency exhibited by the original signal dictate the extent of information that can be recovered from unthresholded RPs. As the re-constructability of a signal is dependent of the embedding parameters, they also noted that the issue is resolved when the embedding dimension m_d is equal to 1. In other words, the information which can be offered by thresholded RPs is identical to that of unthresholded ones. Additionally, in the state of reconstruction distortion, the information obtained from RP is in principle, different from that of unthresholded. To address this, they proposed the idea of multi-level recurrence plot (MRP) along with the assurance of high data compression rate. This idea sprouted from the probe of the possible phenomena when a certain unthresholded RP is discretised using multiple thresholds while under reconstruction disturbance.

In general, critical literature review suggest that the idea of unthresholded RPs, and its variations, are not yet fully established. A number of researchers were able to provide some mathematical equations explaining it about it. However, none of them managed to successfully employ this concept in some applications. This is mainly due to the difficulty in getting enough information that could holistically represent the parent signal and in turn be used for possible applications. This inference is therefore one of the areas of research that require in-depth focus from the perspective of nonlinear analysis.

2.3 Analysis of Recurrence Plots

2.3.1 Visual Interpretation of Recurrence Plots

The form and visual features of RP can provide a representation of the time evolution of trajectories by a certain dynamic system. For example, as seen in Figure 2-2, the characteristic typology of a homogenous (or uniformly distributed noise) time series differs appreciably from that of periodic, drift, or disrupted ones.

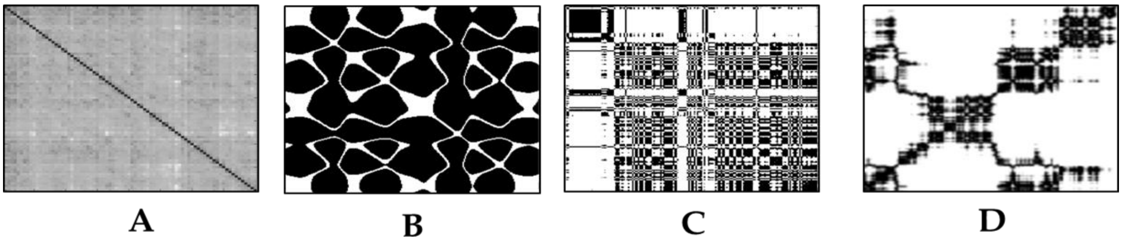


Figure 2-2. Sample RPs showing different typology: (A) homogeneous (uniformly distributed noise), (B) periodic (super-positioned harmonic oscillations), (C) drift (logistic map corrupted with a linearly increasing term), and (D) disrupted (Brownian motion).

In the comprehensive work of Marwan et al. (2007), a list of typical visual features of RPs were presented, as summarised in Table 2-1. They also defined that RPs contain patterns both in large scale (commonly referred as typology) and small scale (commonly referred as texture). The typology gives global impression of the system, while the textures collectively refer to the single dot, diagonal, vertical, and horizontal lines present in the plot.

Table 2-1. Visual Interpretation of recurrence plots

Type	Pattern	Interpretation
Typology	Homogeneity	The system is stationary
	Periodic / quasi-periodic	Cyclic system; the time distance between periodic patterns correspond to the period; different distances between long diagonal lines reveals quasi-periodic system
	Drift (fading to the upper left and lower right corners)	Non-stationary; the system contains a trend or a drift
	Disruptions (white bands)	Non-stationary; some states are rare of far from the normal; transitions may have occurred
Textures	Single isolated points	Strong fluctuation in the system; if only single isolated points occur, the system may be an uncorrelated random or even anti-correlated system
	Diagonal lines (parallel to LOI)	The evolution of states is similar at different epochs; the process could be deterministic; if occurred beside single isolated points, the system could be chaotic
	Diagonal lines (orthogonal to LOI)	The evolution of states is similar at different times but with reverse time, sometimes an indication for an insufficient embedding
	Vertical and horizontal lines/clusters	Some states are constant or are changing slowly over time; an indication of laminar states
	Long bowed line structures	The evolution of states is similar at different epochs; with different velocity the dynamics of the system could be changing

2.3.2 Recurrence Quantification Analysis

Most of the time, it is difficult to visually examine the structures of RPs. Furthermore, most of the applications require numerical interpretation of the recurrence and statistics of the key features of RPs. Thus, the recurrence quantification analysis (RQA) is developed. RQA is a collective term for the features and statistics that can be extracted and computed from RPs. The features generally describe and measure the complexity of the RP which are mostly based on the information derived from the recurrence density (i.e. recurrence rate), on the diagonal lines (i.e. determinism, average diagonal line lengths, entropy), and on the vertical line (i.e. laminarity, trapping time) (Gao & Cai, 2000; Marwan et al., 2007; Webber Jr et al., 2009). Some of the RQA features, along with their descriptions, are listed in Table 2-2.

Table 2-2. The RQA features, and its corresponding descriptions

RQA Features	Description
Recurrence Rate	Percentage of the recurrence points in the recurrence plot
Determinism	Fraction of recurrence points that form diagonal lines (measurement for predictability of the system)
Entropy	Shannon entropy of the probability distribution of the diagonal line length p(l) (measurement of the complexity of the recurrence plot with regards to diagonal lines)
Averaged diagonal line length	The mean of the lengths of the diagonal lines in RPs (often referred to as mean prediction time)
Longest diagonal line	Length of the longest diagonal line in RPs
Longest vertical lines	Maximal length of the vertical lines in RP (provides the degree of complexity of a dynamical system)
Transitivity Coefficient	Quantify the geometric properties of the attractor in the RPs
Recurrence Time Entropy	Quantify the extent of recurrence and is related to Persin dimension

Laminarity	The frequency distribution of the lengths l of vertical structures
Clustering Coefficient	Measures the probability that two neighbours of any given state are also neighbours
Trapping Time	Average length of vertical structures which estimates the mean time at which a particular system will follow a certain state
Recurrence Time	Poincaré recurrence time 1 and 2 generally detects non-stationarity

2.3.3 Application of texture in recurrence plots

Even though the use of RQA has proven reliable to a wide range of applications (e.g. (Hou et al., 2016; Li et al., 2004; Terrill, Wilson, Suresh, Cooper, & Dakin, 2013), there are still studies that explore other approaches to describe and analyse the structures of recurrence plots. It is mostly associated with the characterisation of time series. One of the approaches is using the concept of fractal dimension in the analysis, as it is believed that fractals have a natural relationship to recurrence. Fractals, as initially proposed by Mandelbrot (1967) and generally defined as "self-similar structures observed repeatedly at different scales of magnitude"(Holden, Riley, Gao, & Torre, 2013), is a mathematical concept that can be used to describe the structures of objects. In the study of Babinec, Kucera, and Babincova (2005), this concept is used to characterise the recurrence plots of both regular and chaotic systems, and is particularly applied in the analysis of human electrocardiogram. In their analysis, the recurrence plots are treated as two-dimensional images so that the fractal dimensions can be calculated.

Another contemporary method is the incorporation of the concept of texture in the analysis of RP structure. This is motivated by the idea that RPs of a certain system have distinct visual texture patterns that can be used to analyse its structural changes and thus be used to distinguish from other RPs of different systems. In essence, the method involves understanding of its textures, which requires extraction of textural features using several textural extraction algorithms that are commonly used in image analysis. Based from literature review, it is argued that this idea is quite novel in the research community as limited studies relating to it have been presented.

In the work of Yanhua, Carmona, and Murphy (2006), the co-occurrence based temporal textures are extracted from the time series fluorescence microscope images and are used as predictors in the classification of subcellular location patterns. They regarded the co-occurrence based temporal textures as robust features as these give both temporal and spatial information which became the basis of attaining high classification accuracy. Similar work was done by Singha, Wu, and Zhang (2017) wherein they used a combination of temporal features extracted from coarse resolution time series data and spectral features of fine resolution data for object-based paddy rice mapping application. The temporal features are extracted on the Moderate resolution imaging spectroradiometer (MODIS) of the remote sensing of paddy rice.

In terms of utilizing texture algorithms in studying the structures of time series, the study of Souza et al. (2014) can be considered the closest one. Coined as 'Texture Features from Recurrence Patterns' (TFRP), the method used textural algorithms to extract the features in the recurrence plots. The combination of all extracted features from Local Binary Pattern (LBP), Grey-Level Co-occurrence Matrix (GLCM), Gabor filters, and Segmentation-based Fractal Texture Analysis (SFTA) were employed as predictors in classification of the UCR Time Series Archive (Bagnall, Batista, Begum, Chen, Keogh, Hu, & Mueen, 2015). Furthermore, TFRP was also compared to other methods they have previously developed. One of which is the 'Recurrence Patterns Compression Distance' (RPCD) Silva, Souza, and Batista (2013), which also use recurrence plots, and with the incorporation of 1-NN algorithm to estimate the similarity of the two recurrence plots via employing a video compression based distance measure (CK-1). From there, the comparison of the texture similarity between two images could be possible in RPCD using the Kolmogorov complexity.

2.3.4 Application of convolutional neural network in recurrence plots

Due to the promising results, the use of CNN has gained popularity in the research community. However, its application to time series analysis, specifically its use in texture analysis in time series images, is still in the infancy period as only few papers were found in this area. One of which is the study of Z. Wang and Oates (2015) where they proposed a framework for encoding time series as different types of images, i.e. Gramian Angular Summation/Difference Fields (GASF/GADF) and Markov Transition Fields (MTF), and used Tiled CNN to learn these time series images. In their study, the time series were represented in 2 images: the first is in polar coordinates transformed into Gramian matrix to form Gramian Angular Field (GAF) images, and the second is in Markov Transition Field (MTF) built by discretised Markov matrix of quantile bins. More importantly, their study explored the use of Tiled CNN, which uses tiles that are parameterised by a tile size k to control the distance over shared weights, and successfully achieved competitive results in terms of time series classification against other published methods.

Another related study is the work of Guangliang et al. (2016) where they transformed physiological signals such as single-channel Electrooculogram (EOG) data into RPs and used CNN to extract its features. The CNN architecture consists of 102 x 45 size input layer, 2 convolutional layers, 1 max pooling layer, 2 dropout layers and 1 fully connected layers. Results showed that their approach attained higher accuracy against other methods.

Lastly, the study of Hatami, Gavet, and Debayle (2017) is perhaps the closest research work by far. In their paper, the time series are transformed into grey-level texture images which is equivalent to unthresholded recurrence plots, and eventually used as inputs for texture extraction using 2-stage CNN. The CNN architecture has 1-channel input of size 28 x 28 and the output layer with *c* neurons. Their proposed approach was applied in time series classification and achieved competitive accuracy among advanced algorithms.

2.3.5 Application of texture analysis to minerals processing time series

While the application of texture analysis in minerals processing including grinding and comminution circuits is not a novel idea, the use of texture analysis to analyse the time series data of any minerals engineering is a novel methodology. As far as the authors are concerned, there are still no published related literature that deal with the application of texture analysis to any minerals processing time series. Moreover, limited literature were found on the characterisation of nonlinear behaviour of any minerals related process systems that uses texture analysis and the state-of-the-art convolutional neural networks.

2.4 Research Gap

As discussed in section 2.2, unthresholded recurrence plots have mostly been used qualitatively in the interpretation of time series data. Nonetheless, in principle these plots contain more information than their binarized versions and if this can be quantified, it should provide a parallel and possibly more powerful approach to the recurrence quantification analysis. This is essentially what will be explored in this thesis, focusing mostly on time series applications in process engineering.

A rich framework of analytical methods can be accessed by considering the unthresholded recurrence or distance plots as textural images and in this investigation, multivariate image analysis based both on engineered features via the use of grey level co-occurrence matrices, wavelets, local binary patterns and textons, as well as learned features via the use of convolutional neural networks are considered.

The steps involved in this methodology, along with the state-of-the-art textural feature extraction algorithms that are considered in the study, are described in detail in Chapter 3.

3. RECURRENCE TEXTURE ANALYSIS: THE METHODOLOGY

3.1 Introduction

Dynamic process systems, as usually represented by time series, are ubiquitous in nature. As such, it is recognized that the time series provides substantial information about the underlying mechanisms of these systems. This led to the growth of studying this kind of data, mainly on its properties that could be used for further characterisation. As a particular, the use of the patterns of recurrence inside the time series has attracted a lot of interest among researchers for the past years. As seen in the literature review, the use of this characteristic resulted to the massive studies related to it, with emphasis on the development of methods that could describe the data in a deeper sense, and on its practical applications to several scientific domains. Among these methods is the Recurrence plot, which is a graphical representation of the recurrence, and the Recurrence Quantification Analysis, which is a quantified counterpart of the plots. Although these have achieved considerable success, the methods are still encountering a number of challenges (e.g. choice of threshold distance, embedding parameters) for full implementation. In this sense, exploring alternative methods to study the recurrence patterns and subsequently characterise the structures of time series data is deemed necessary.

In this chapter, an alternative method in capturing the dynamic behaviour of time series data is introduced. This is termed as Recurrence Texture Analysis. The general framework of this method, with detailed explanation of the steps involved are thoroughly discussed here.

3.2 Motivation

In the research standpoint, as mentioned in the introduction, the complete utilisation of recurrence plots in the analysis of time series is still a challenge owing to several issues, particularly in the selection of threshold distance. One could say that the said issues would be addressed by using unthresholded recurrence plots (or the distance plots/ global recurrence plots) (Sipers et al., 2011) which is referred to the distance matrix. From now on, the word "distance matrix" is a collective term for unthresholded recurrence plots, distance plots or global recurrence plots. However, the use of distance matrix alone in the analysis of nonlinear time series is still in the initiation stage, mainly due to the insufficient methods that could be used to capture relevant information of the time series which are believed to contain in the matrix.

In another note, it is acknowledged that recurrence plots (or distance matrix if unthresholded) contains visual patterns of recurrence, both in large and small scales. These patterns, especially the small scale patterns, are very important in analysing the structure of the data. These small scale pattern are also called textures. In Chapter 2, it is revealed that the concept of texture in the context of dynamic system is quite a novel theme in research community especially the convolutional neural network. It is in this view that this thesis is inspired: to incorporate the concept of texture analysis in analysing dynamic system, particularly by use of distance matrix to represent time series. In other words, this thesis believes that the time series data, as transformed into distance matrix, contain recurrent behaviour that are governed by a well-defined texture descriptors which can be extracted by use of texture analysis algorithms. As a consequence to this, the recurrence texture analysis is then developed and proposed here.

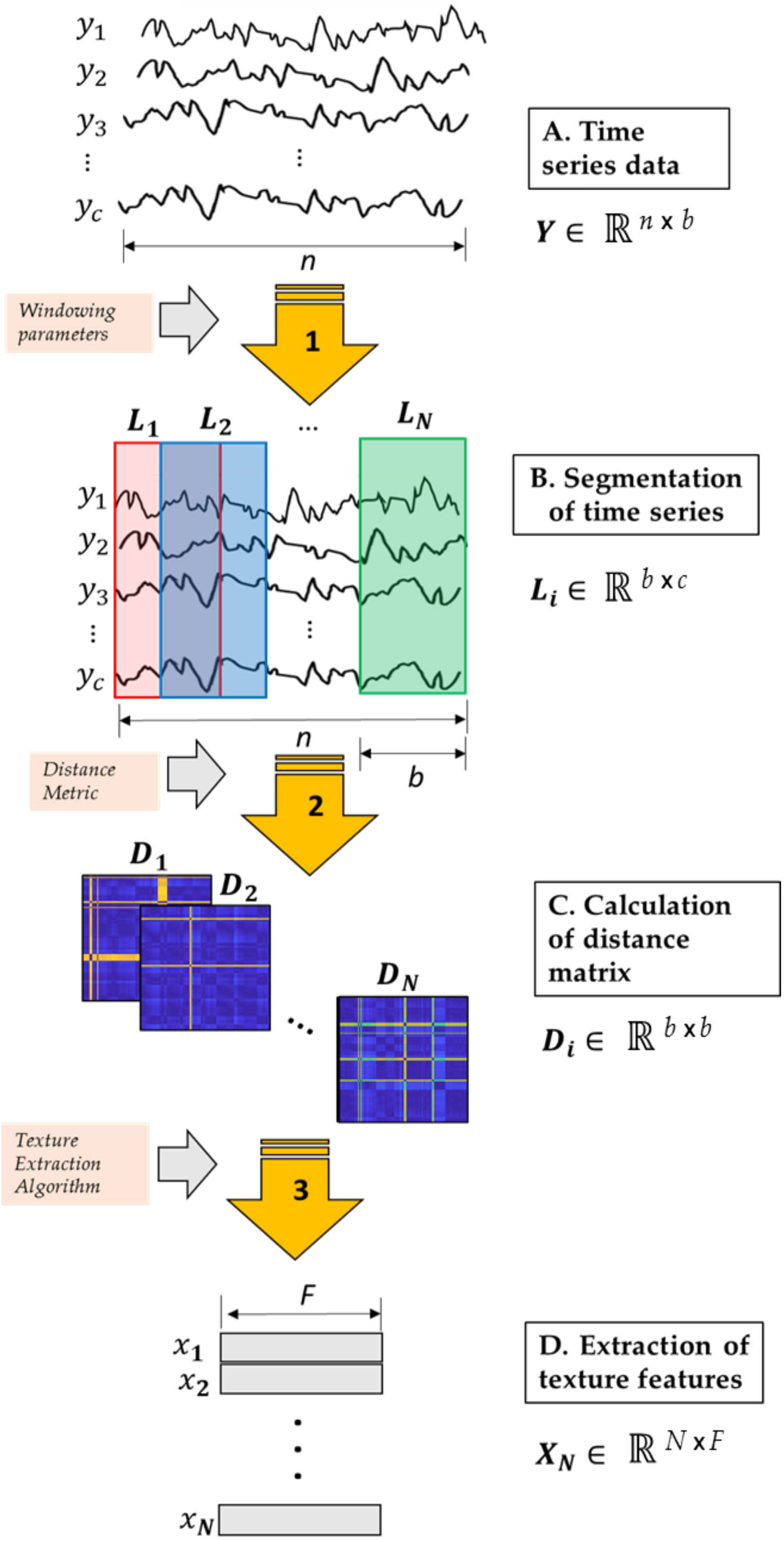
Another important motivation in this research is the consideration of deep convolutional neural network in application to process engineering, and to time series analysis in general. Specifically, the use of pretrained CNNs, i.e. AlexNet and VGG16, are explored here as contemporary feature extraction algorithms to distance matrices that were not initially part of its trained images. Lastly, this is also motivated by extending the application of RTA to dynamic process monitoring and to its real case application such as in grinding circuit and solids processing.

3.3 Recurrence Texture Analysis

Recurrence texture analysis (RTA) is a synthesis of two entirely different fields: textural analysis and non-linear time series analysis. The concept of textural analysis is widely used in machine vision, particularly in image analysis whereas the concept of non-linear time series analysis is a broad field of study that is originated from the theory of dynamical systems. The use of the idea of recurrence inside the data, which triggered the conception of recurrence plot analysis, is just a part of a bigger field. This is where the proposed method belong.

This method focuses on capturing the structure of the time series data by means of extracting texture features from the distances between measurements over time, commonly exemplified by distance matrices. This is achieved through analysis of the structure of the data during discrete periods of process observation. The general overview of this method is presented in **Figure 3-1**.

From the definition, RTA requires segmentation of the parent time series data into several discrete time series using the windowing parameters. It should be noted that the embedding of data can also be incorporated in the method. However, for ease of evaluation, the embedding parameters are set always to unity ***(τ*** *, m_d=1)* throughout the analysis.



y_1
y_2
y_3
y_c
n
A. Time series data
$Y \in \mathbb{R}^{n \times b}$
Windowing parameters
1
L_1
L_2
...
L_N
B. Segmentation of time series
$L_i \in \mathbb{R}^{b \times c}$
n
b
Distance Metric
2
D_1
D_2
D_N
C. Calculation of distance matrix
$D_i \in \mathbb{R}^{b \times b}$
Texture Extraction Algorithm
3
F
x_1
x_2
x_N
D. Extraction of texture features
$X_N \in \mathbb{R}^{N \times F}$

Figure 3-1. Schematic workflow of the study, comprising the time series data (A), segmentation of the data (B), calculation of distance matrix for each time series segment (C), and extraction of texture features from each matrix (D).

The method also requires calculation of distance matrix from these discrete time series data. The distance matrix uses distance metric (distance measure or similarity measure). The most common distance metric (also used here frequently) is the Euclidean distance, although other distance metrics could also be considered (e.g Chebychev, Mahalanobis) as part of the optimisation study of the method. Then, the distance matrices are then visualised and the textures of the recurrence patterns are analysed. In the textural analysis, the information of the textures are extracted using RTA algorithms namely GLCM, wavelet, LBP, textons, CNN-AlexNet, and CNN-VGG16, which are all discussed in this chapter. The extracted information are then termed as "feature matrix", which consequently be used as predictors for analysis and other applications.

3.3.1 Segmentation of the time series

In this paper, the term "segmentation of time series" refers to the process of splitting-up or partitioning the parent time series into several discrete time series (termed as segments). The segmentation entails windowing parameters, viz. window width *(b)* and sliding step *(m)*, which would define the dimension of the segments. For the window width b, this parameter is estimated using the autocorrelation function. In essence, each segment of discrete time series having a width of *b* should contain the dynamic behaviour of the parent time series. More formally, given a time series $\mathbf{Y}(t)$ and its lagged counterpart $\mathbf{Y}(t + b)$, the *b* is obtained using the eqn (3-1).

$$min_b(ACF) = \frac{1}{(N-b)\sigma^2}\sum_{b=1}^{\mathrm{N-b}}\{(\mathbf{Y}(t)-\overline{\mathbf{Y}})(\mathbf{Y}(t+b)-\overline{\mathbf{Y}})\} \quad (3\text{-}1)$$

where $\overline{\boldsymbol{Y}}$ and σ^2 are the mean, and standard deviation of the time series $\mathbf{Y}(t)$, respectively, and *N* is the number of observations in the time series.

It should be noted that the value of b should be chosen carefully as it is usually dependent on the kind of structural changes the researchers are being investigated. Large value of b would correspond to inefficient detection of the change in the system. On the contrary, too small value of b would normally cause incorporation of outliers which subsequently detected as a structural change, thus giving false interpretation of the data (Auret & Aldrich, 2010).

The sliding step m, on the other hand, dictates the steps on the movement of the window. Generally, there are three kinds of windowing approach, viz., moving, stationary or fixed, and progressive. Moving window is performed when the sliding step is less than window width (*m<b).* When *m=b*, fixed windowing is done. It is where the segments do not overlap with each other. Progressive windowing, on the other hand, is a multiscale type of windowing wherein the sliding step is progressively increasing. This computation is very expensive, thus it is less preferred. Normally, the choice of the sliding step m should be as small as possible. However, it should be noted that the selection of m is a trade-off between the information being explained and the computational efficiency. Using too small m is computationally expensive. Moskvina and Zhigljavsky (2003) remarked that a reasonable value of this parameter should be at half of the window width (*m=b/2)* to give a reliable capture on the information of the time series.

Once the windowing parameters (*b,* and *m)* are defined, the parent time series are then segmented into *N* number of segments. Each of this segment contain b elements. In general, given a time series $\in \mathbb{R}^{nxc}$, where n and c are the number of rows (sample size or observations) and the number of columns (variables or dimensions), respectively and using the defined windowing parameters, b and m, a set of N segments, denoted by $\mathbf{L}_i \in \mathbb{R}^{bxc}$ for $i = 1,2, \ldots N$, is obtained. The value of N is can be computed according to eqn (3-2).

$$N = Round\ down(\frac{n-b}{m} + 1) \tag{3-2}$$

In this step, it should be noted that the time series is preferred to be in normalised form to avoid misinterpretation of the results. Moreover, since the method requires equally divided segments, the remainder of the segments that do not have lengths equal to b are removed and disregarded in the analysis.

3.3.2 Calculation of distance matrix

The distance metric plays an integral part in Recurrence Texture Analysis since this will give some knowledge on the shape of the data and will quantify the distances of any points in the time series, which eventually led to the establishment of recurrence patterns. The calculated distance measurements are transformed into matrix called distance matrix and is plotted for analysis.

To recall, RTA is a variation of an analysis of recurrence plot, which is constructed using threshold distance ε, and a norm $\|\cdot\|$. With RTA, the use of threshold distance is omitted, which makes it only dependent on the norm $\|\cdot\|$. This norm is now referred to as distance metric. In the review paper of Marwan et al. (2007) where they considered three commonly used norms (i.e. L1- norm, L2- norm or the Euclidean norm and the L∞-norm or the Maximum norm), it is noted that the L∞-norm is preferred due to its high computational efficiency. Interestingly, the L2- norm (Euclidean norm) is seen to be a popular alternative and has gotten a number of successful applications using this.

It is noted that in mathematics, the terms "norm" and "distance" are different and should not be used interchangeably. A norm refers to a function of one element which is the length of a vector, while distance refers to a function of two elements which is the length of the line segments connecting these two elements. As a particular, the Euclidean distance uses Euclidean norm (L_2 norm) to calculate the proximity of a given point x_i from another point x_j to form a distance square d_{ij} using the equation below.

$$d_{ij} = \left\|x_i - x_j\right\|^2 \tag{3-3}$$

In contrast, Chebyshev distance uses Maximum norm (L_∞-norm) to define two points (i.e. x_i, x_j) where their distance is the maximum of their differences (refer to eqn (*3-4*)).

$$d_{ij} = \max\{\left\|x_i - x_j\right\|\} \tag{3-4}$$

Generally, the Euclidean distance and Chebyshev distance are both derived from Minkowski distance. As seen in eqn (*3-5*), the formula is equal to Euclidean distance when Minkowski distance has $\boldsymbol{p}$=*2*. Similarly, Minkowski distance takes form the Chebyshev distance when $\boldsymbol{p}$=∞.

$$d_{ij} = \sqrt[p]{\sum_{i,j=1}^{n} \left\|x_i - x_j\right\|^p} \tag{3-5}$$

Considered as vital component in the method, the selection of distance metric should always be taken into consideration. Thus, studying the effect of distance metrics as part of optimisation of the method should be made. A dedicated discussion of this, along with the results, are presented in the Chapter 5.

Nevertheless, regardless of the type of distance metric, a symmetric distance matrix in a square form is constantly obtained. This distance matrix is computed using each of the points from the segmented time series. In other words, given a set of segments $\mathbf{L}_i$ for *i=1, 2, 3…N*, and using an appropriate distance matrix, each of the segments would take form as distance matrices $\mathbf{D}_i$ for *i=1, 2, 3…N,* correspondingly after calculation. Once done, it is necessary to plot these distance matrices for visual inspection and texture analysis. Samples for the distance matrix plots are shown in **Figure 3-2**.

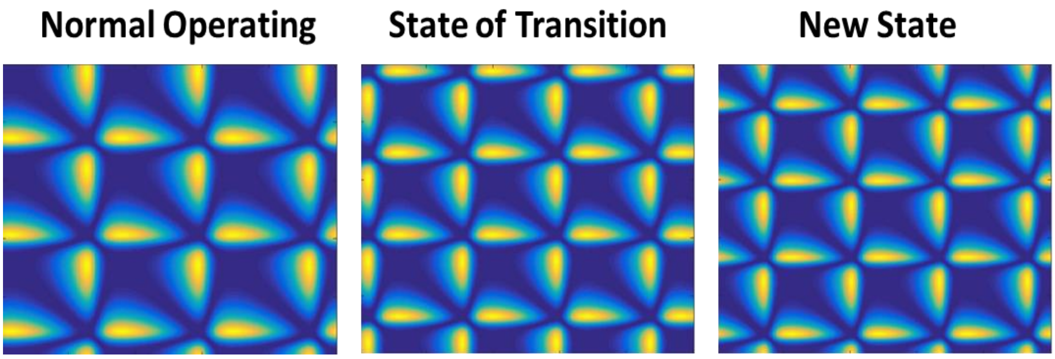


Figure 3-2. Samples of the distance matrices showing the different structures for different classes

3.3.3 Texture Feature Extraction

After plotting the distance matrices, the recurrence patterns are then analysed. In RTA, several texture extraction methods are employed to quantify the texture information present in the distance matrices. In principle, these textural features are the numerical representation of the recurrence exhibited in each discrete time series, and in the whole parent time series, in general. Furthermore, these textural features would serve as predictors in analysing the structures of time series, and to other applications, particularly in data mining tasks.

In this thesis, six textural feature algorithms, which are discussed in detail in the next sections, are considered here. These are:

- Grey-level co-occurrence matrix (GLCM)
- Wavelet transforms
- Local Binary Patterns
- Textons
- AlexNet
- VGG16

It should be noted that the full optimisation of the algorithms (i.e. hyperparameters optimisation) is beyond the scope of the study. Although, to some extent, this contributes to the overall performance and robustness of the proposed method, this is however, less prioritised. In the testing of the method, most of the time, the default hyperparameters for these algorithms are used. Furthermore, not all algorithms are considered in some studies. In one way or another, at least one algorithm is only used during the evaluation study of the method to several applications. In time series classification, however, these algorithms are all utilised, and their hyperparameters were optimised.

As summarised in **Table *3-1***, most of the features are extracted using a combination of these hyperparameters of the techniques using the default values.

Table 3-1. Hyperparameters of the texture extraction algorithms used in the study

Algorithm	Hyperparameter	Default values	No. of Features
GLCM	Number of grey levels	32	8
	Distance	1	
	Directions (degrees)	[0, 45, 90, 135]	

Wavelet	Type of wavelet	Daubechies (db4)	9
	Level of wavelet decomposition	3	
Local Binary Pattern	Radius of circular local pixel neighbourhood (R)	1	256
	No. of sampling points in the pixel neighbourhood	8	
	Map Type	No mapping	
Texton	Filter Name	Schmid	40
	Filter Size	49	
	No. of cluster centers	40	
AlexNet	No. of Convolutional Layer	5	4096
	No. of Pooling Layer	3	
	No. of Fully Connected Layer	3	
	Image Resolution (pixel)	[256 x 256]	
VGG16	No. of Convolutional Layer	13	4096
	No. of Pooling Layer	5	
	No. of Fully Connected Layer	3	
	Image Resolution (pixel)	[224 x 224]	

Once extracted, the features or textural descriptor variables extracted from the distance matrices of the time series are aggregated into a feature matrix $\mathbf{X} \in \mathbb{R}^{N \text{ x } F}$, where N is the number of time series segments, and F is the number of features depending on the type of algorithms used. In other words, the time series matrix $\mathbf{Y} \in \mathbb{R}^{n \text{ x } c}$ is converted to a data matrix with dimensions $N \ll n$ and $F \gg c$.

Some of the works have combined features from two algorithms so that more influential texture descriptors can be generated. For instance, in the process monitoring study, the GLCM features ($GLCM$) and wavelet features (W) are combined, thus $F = GLCM + W$.

3.3.4 Data Analysis

In the context of data analysis, the extracted RTA features are treated as predictors to several models, depending on the type of applications being considered (e.g, data visualisation, classification, fault detection). As such, this requires some of the techniques used in data analysis including dimensionality reduction and machine learning techniques. In this subsection, those methods that we used in this thesis are discussed in detail.

Principal Component Analysis

Principal component analysis (PCA) is one of the most common and powerful tools in dimensionality reduction and feature extraction. The common goal of this technique is to produce a set of uncorrelated variables or components which will represent the variance information of the original variables. This is done by projecting the data in a reduced hypersphere that is defined by orthogonal vectors called principal component (PC) (Ballabio, 2015; Jiang & Yan, 2014).

In general, PCA works by decomposing an autoscaled data matrix $\boldsymbol{X} \in \mathbb{R}^{N \times m}$ into a score matrix $\boldsymbol{T} \in \mathbb{R}^{N \times k}$, a loading matrix $\boldsymbol{P} \in \mathbb{R}^{m \times k}$ and a residual matrix $\mathbf{E}$, where N and m are the number of samples and number of variables in a certain data set, respectively, and k is the number of PCs retained. It can be written as:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{\boldsymbol{T}} + \boldsymbol{E} \qquad (3\text{-}6)$$

The k number of PCs retained can be determined through several methods. One of which is by cumulative percent variance (CPV) method where the first k PCs that represent the major variance information of the original data are selected (Jiang & Yan, 2014). It can be formulated as

$$CPV(\%) = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{m} \lambda_i} x100\% \tag{3-7}$$

where λ is called the eigenvalues, which is the variance of score matrix. However, there is no strict rule on how many percent variance is acceptable for the k PCs could be retained.

Linear Discriminant Analysis

The linear discriminant analysis (LDA) is one of the extensively studied and most popular methods used in classification problems and dimensionality reduction ((Safo & Ahn, 2016)). In this method, the continuous variables that could discriminate the classes are determined. It handles the case where the within-class frequencies are unequal. This also provides maximal class separability by maximizing the between-class over within-class variances in a particular set of data.

Formally, suppose there is a given data matrix $\mathbf{X} \in \mathbb{R}^{N \times m}$ of N samples in m-dimensional space, then LDA searches for maximum separability via looking through a transformation $\dot{\boldsymbol{w}}$ that maps the matrix $\mathbf{X}$ into a lower-dimensional latent variable f. That is:

$$f = \dot{w}^T \boldsymbol{X} + B \tag{3-8}$$

where B is referred to as a bias term.

In general, the l- dimensionality of f is satisfied using the eqn (3-9).

$$f_l = r - 1 \tag{3-9}$$

where r is the number of classes in the data set.

Gaussian Mixture Model

The Gaussian mixture model (GMM) is a probabilistic approach of assigning an observation to a certain cluster. Through this method, it is possible to fit a number of n-dimensional normal distributions to the data. By defining the probability of the observed data from each component distribution, clustering is then created.

Generally, the GMM having K Gaussians can be defined using the probability density function (Yu, 2012) which can be expressed as:

$$p(x|\Theta) = \sum_{i=1}^{K} \omega_i p(x|\theta_i) \tag{3-10}$$

where ω_i represents the prior probability of the i-th component, and $p(x|\theta_i)$ refers to the probability density functions of the ith component. This $p(x|\theta_i)$ can be further defined as:

$$p(x|\theta_i) = \frac{1}{2\pi^{D/2}|\sum_i|^{1/2}} exp\{-1/2(x - \mu_i)^T \textstyle\sum_i^{-1}(x - \mu_i)\} \tag{3-11}$$

where D represents the D-dimensional variable of x, θ_i ={μi, ∑i) is the probability distribution parameters of ith Gaussian component. It can also be noted that the ω_i, ∑i and μi are optimised using the expectation-minimization algorithm (Yu, 2012).

The Gaussian mixture models have to be fitted with care, to ensure that neither underfitting (too few Gaussians), nor overfitting (too many Gaussians) becomes a problem. The version of the Akaike information criterion (Akaike, 1974) that was used to optimize the value of K, is given by eqn (3-12) (Hurvich & Tsai, 1989) .In this equation, the criterion $AIC(\boldsymbol{\Theta})$ is minimized, given the Gaussian mixture model ($\boldsymbol{\Theta}$), the probability of the sample $\mathbf{x}$, given the model $p(\mathbf{x}|\boldsymbol{\Theta})$, the number of Gaussians (K), as well as the number of samples (K).

$$AIC(\boldsymbol{\Theta}) = -2\log[p(\mathbf{x}|\boldsymbol{\Theta})] + 2K\left(\frac{N}{N-K-1}\right) \tag{3-12}$$

Support Vector Machine

The support vector machine (SVM) is a supervised learning method where the maximal margin hyperplane is achieved, and the distance from the hyperplane to the nearest data points on each side is maximized (Kim, Choi, & Lee, 2015; Tax & Duin, 2004). More formally, in building the minimum hypersphere with center a, and radius R, an error function can be formulated as:

$$F(R,a) = R^2 + Cp\sum_i \xi i\,, i = 1, \dots, N \tag{3-13}$$

where ξi is the slack variable and Cp is the controlling parameter. However, to construe that almost all objects are within the hypersphere, the constraints should also be defined as referred to eqn (3-14).

$$||x_i - a||^2 \leq R^2 + \xi i, \qquad \xi i \geq 0, \qquad \forall i \tag{3-14}$$

It is noted that the controlling parameter Cp dictates the adjustment between the volume of the hypersphere and the errors. With the maximization of the corresponding primal Lagrangian, equation (3-15) could be transformed to function L as defined in eqn (3-15).

$$L = \sum_{i} \alpha_i (x_i \cdot x_i) - \sum_{i,j} \alpha_i \alpha_i \left(x_i \cdot x_j\right) \tag{3-15}$$

which has a constraint of $0 \leq \alpha_i \leq Cp$ and i=1,…, N. Thus, for a given test object z, the object is accepted if the calculated distance to the center of the sphere is smaller or equal than the radius R. This is better understand in eqn (3-16).

$$\begin{aligned} \|\boldsymbol{z} - a\|^2 = (\boldsymbol{z} \cdot \boldsymbol{z}) - 2\sum_{i} \alpha_i (\boldsymbol{z} \cdot x_i) + \sum_{i,j} \alpha_i \alpha_j \left(x_i \cdot x_j\right) \\ \leq R^2 \end{aligned} \tag{3-16}$$

where i, j =1,…, N .

The kernel function $K\ (x_i,\ x_j)$ can also be incorporated in SVM. It can simply be substituted into the inner products in eqn. (3-17) with a pre-defined function mapping $\psi\ (\ .\ )$. Having said this, it could be written as:

$$\begin{aligned} \|\varphi(\boldsymbol{z}) - a\|^2 = K(\boldsymbol{z} \cdot \boldsymbol{z}) - 2\sum_{i} \alpha_i K(\boldsymbol{z} \cdot x_i) \\ + \sum_{i,j} \alpha_i \alpha_j K\left(x_i \cdot x_j\right) \leq R^2 \end{aligned} \tag{3-17}$$

Note that the kernel function should first satisfy Mercer's theorem.

3.4 Grey Level Co-occurrence Matrix

The grey-level co-occurrence matrix (GLCM) is a well-established statistical-based method in examining the texture of an image. Proposed by Haralick in 1970s, GLCM is a measure of frequency of the combinations of pixel brightness values occuring in an image. It has been widely used in information theory, such as in image analysis, e.g. (Jemwa & Aldrich, 2012; Kistner, Jemwa, & Aldrich, 2013), in seismic signal interpretation (Eichkitz, Amtmann, & Schreilechner, 2013), and in medical sciences (e.g. (Kovalev, Kruggel, Gertz, & Cramon, 2001)), among others. In principle, GLCM provides features that are based on the spatial interrelationships of the images pixels at which the grey levels occur at a specified displacement (d) and angle (θ) from each other. Thus, the direction, adjacent interval and amplitude variation of an image (I) with grey levels i and j can then be determined as exemplified in eqn (3-18) (Ou, Pan, & Xiao, 2014).

$$P(i,j,d,\theta)=\sum_{x=0}^{n}\sum_{y=0}^{m}\{\begin{matrix}1, if\ I(x,y)=i\ and\ I(x+d\cos\theta,y+dsin\theta)=j\\ 0, otherwise\end{matrix} \qquad (3\text{-}18)$$

The GLCM could actually extract up to 14 GLCM features or the Haralick set, with energy (ENE), contrast (CON), correlation (COR), and homogeneity (HOM) being the most popular GLCM features. These features are usually computed over four different orientations, viz. 0°, 45°, 90° and 135°.

3.4.1 GLCM Calculation

To further understand how GLCM works, the methodology behind GLCM calculations are described in this subsection. Generally, prior extracting GLCM features, several calculations are performed, which could be divided into three: scaling, image transformation to GLCM image, and normalisation. Furthermore, since the GLCM of an image is a function of its orientation (i.e. 0^0, 45^0, 90^0, 135^0), it is indeed possible to calculate more than one GLCM for each image. In fact, Haralick (1979) and others used to calculate GLCM for four different displacements, viz., 0^0, 45^0, 90^0, 135^0 with a constant number of grey level *(G).*

In the scaling process, the image *(I)* is first converted to a greyscale image I_G which is commonly referred to as 8-bit images with ($2^8\ = 256$) pixel intensities or grey levels, g ($0 \leq g \leq 255, g \in \mathbb{R}$). The number of grey levels (G) is then selected to scale this greyscale image I_G and subsequently compute the GLCM. It should be noted that this parameter is an important factor which directs the computational time and the trade-off between the noise reduction and the loss of original image information (Clausi, 2002), thus this should be optimised. Usually, the G = 8 is used in most GLCM computation. Using the eqn (3-19), the scaled greyscale image I_{sc} having $n\ x\ m$ matrix is then computed.

$$I_{sc} = \frac{I_G - g_{min}}{g_{max} - g_{min} + 1} \qquad (3\text{-}19)$$

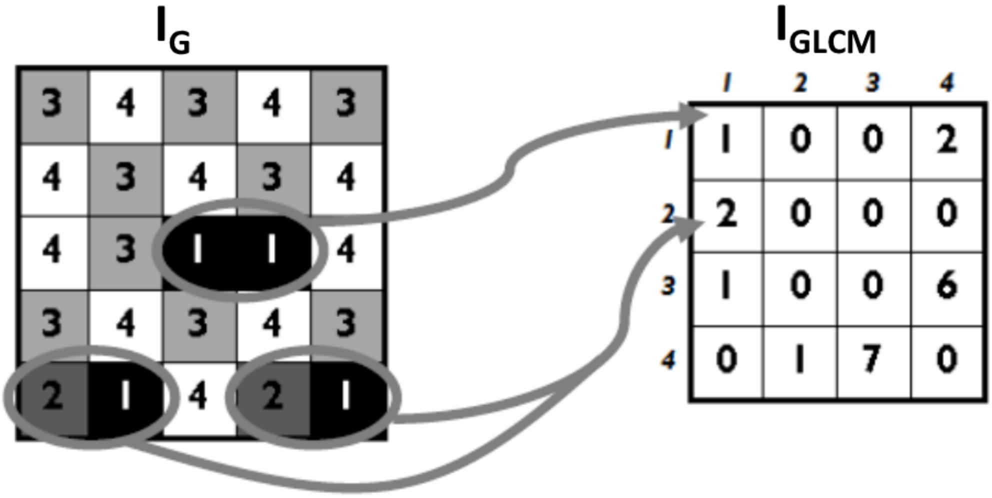
where g_{min} refers to the minimum grey level and g_{max} refers to the maximum grey level in the image I. To compute for the GLCM image (I_{GLCM}), a flooring operator $[.]$ is used as shown in the equation below.

$$I_{GLCM} = [I_{sc} x\ G] + J \qquad (3\text{-}20)$$

where J is a $n\ x\ m$ matrix containing all ones. In turn, I_{GLCM} is also a $n\ x\ m$ matrix containing grey levels $1 \leq g \leq G,\ g \in \mathbb{R}$. Once computed, the GLCM is then normalised so that the sum of its element should equal to 1 (Kistner, 2013).

In GLCM, the calculation uses pixel pairs as the pixel neighbourhood. In other words, each entry in the GLCM $(\widehat{p_{i,j}})$ corresponds to the number of occurrences where the grey level pair (g_i, g_j) is seen in the image for a particular displacement *(d).* For example as shown in **Figure** ***3-3*** considering a 5 x 5 image for a particular displacement d = (0,1) with G = 4, the 4 x 4 GLCM is then computed. The entries in the GLCM matrix are the number of times that a particular pixel pair has occurred in the image. For instance, the pixel pair (1,1) is only found once in the image, thus giving a value of 1 in the GLCM.

In contrast, the pixel pair (2,1) has occurred twice in the image, thus giving a value of 2 in the GLCM. Same goes for other entries. It is also noted that the ordering of grey levels (i.e. (2,1) ≠ (1,2)) in this example was considered, thus giving a non-linear symmetric matrix. However, it is also possible to disregard the ordering, which in turn yields a symmetric co-occurrence matrix.



3	4	3	4	3
4	3	4	3	4
4	3	1	1	4
3	4	3	4	3
2	1	4	2	1

	1	2	3	4
1	1	0	0	2
2	2	0	0	0
3	1	0	0	6
4	0	1	7	0

Figure 3-3. Illustration showing the GLCM calculation

As mentioned elsewhere, the normalisation of GLCM, given in eqn (3-21) is implemented after the calculation so that the GLCM is transformed into a probability matrix of joint pixel occurrences.

$$\hat{P}_i = \frac{P_i}{\sum_{i,j} p_{i,j}} \tag{3-21}$$

where $\hat{P}_i$ is the normalised form of P_i, and $p_{i,j}$ is the entry in (i,j).

3.4.2 GLCM Features

GLCM features are extracted using the normalised GLCM matrix. Most of the works that use GLCM use the Haralick features. The Haralick features, as popularised by Haralick (1979), are said to contain strong features that could characterise the texture of an image. It originally consisted of 14 features but was eventually reduced to 5, due to its high correlation to other features. These 5 features are energy, entropy, contrast, correlation and homogeneity. However, in the study of Clausi (2002), the energy and entropy were ascertained to be highly correlated as both of these features provide information on the uniformity of the GLCM. Accordingly, the entropy can then be removed from the feature set.

Energy is considered as an angular second moment. It measures the uniformity of the GLCM. On the other hand, contrast provides the inertia or the variance of the average grey level between pixel neighbours. Meanwhile, correlation is somewhat similar to contrast since this also measures variance. However, correlation describes the differences of a pixel from its neighbour with respect to the whole image. Moreover, it uses the mean and standard deviation of the GLCM in the computation. Lastly, the homogeneity compares the distribution of elements in the GLCM to that of the GLCM diagonal (Kistner, 2013). These features are usually computed over four different orientations 0°, 45°, 90° and 135°, and their equations are shown in eqns (3-22) - (3-25).

$$ENE = \sum_{i,j} {\hat{p}_{i,j}}^2 \tag{3-22}$$

$$CON = \sum_{i,j} |i-j|^2 \, \hat{p}_{i,j} \tag{3-23}$$

$$COR = \sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)\hat{p}_{i,j}}{\sigma_i \sigma_j} \tag{3-24}$$

$$HOM = \sum_{i,j} \frac{\hat{p}_{i,j}}{1+|i-j|}, \tag{3-25}$$

where $\hat{p}_{i,j}$ represents the entry in row i and column j of the normalised GLCM $\hat{P}_I$, while μ_i and σ_i are the respective mean and standard deviation of the i-th row of the GLCM, and μ_j and σ_j are the mean and standard deviation of the j-th column of the GLCM.

3.5 Wavelet Transforms

The wavelet transform is a powerful signal and image processing tool that is based on mathematical functions which uses a so-called wavelet. Properties for real-world signal analysis such as space-scale localisation, multiresolution, sparse representation and efficient computation are some of the attributes wavelet transforms possess, making this algorithm suitable for signal and image processing tasks. There are several recurrent families of wavelet transform such as Haar, Daubechies, and Symlet transforms, among others. However, several studies have shown that the Daubechies wavelets are best used when dealing with signal and image processing tasks (Ruiz de la Hermosa González-Carrato, García Márquez, Dimlaye, & Ruiz-Hernández, 2014).

The concept of wavelet transform is brought about by the quest for a solution on the futility of Fourier transform to deal with non-stationary signals. Most of the time, its form is regarded as either continuous or discrete wavelet transform.

3.5.1 Continuous wavelet transform

The continuous wavelet transform (CWT) analyses the signal using wavelets, ψ as analysing function. This wavelet is initially directed to transform through dilation or scaling, which involves shifting, compression or expansion of the function. Once already transformed, it is then compared to the signal and the similarity between the two signals are then measured. More formally, CWT can be written as:

$$W(s,\tau) = \int x(t)\psi_{s,\tau}^{*}(t)dt \tag{3-26}$$

where $W(s,\tau)$ is an integral of the transformed function of x(t), which is a signal in the time domain multiplied by a conjugate transposition of the wavelet $\psi_{s,\tau}$. The expression τ and s represent the translation and scales of the signal, respectively.

In multi-resolution analysis, the mother wavelet ψ is allowed to decompose into different wavelets using varying scales, s and translation, τ (Kistner, 2013). The decomposition is exemplified by eqn (3-27).

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right) \tag{3-27}$$

CWT can also be discretised by sampling the scale using a uniform sampling rate. In this sense, heavy computation could be addressed. Furthermore, when the sampling rate is non-uniform, the transform will then referred to as discrete wavelet transform.

3.5.2 Discrete wavelet transforms

Discrete wavelet transform (DWT) (Croisier, 1976) can be generated by sampling the translation-scale plane at non-uniform rate. More formally, DWT is:

$$\psi_{j.k}(t) = \frac{1}{\sqrt{s_0^j}}\psi\left(\frac{t-k\tau_0 s_0^j}{s_0^j}\right) \tag{3-28}$$

where s_0 and τ_0 are constant scaling and translation factors respectively. s_0^j and $k\tau_0$ are the new scale and translation factors of the discrete wavelet, while j =1, 2, 3, … J_{max}} indicate the levels of decomposition of the wavelets. J_{max} is the maximum level were the scale becomes too large to analyse. In this investigation, values of $s_0 = 2$ is commonly used, so that the scale is discretised on a logarithmic grid with base 2, and only the wavelets at scales {2, 4, 8, 16, …} are used for transformation. The translation factor was set at $\tau_0 = 1$.

3.5.3 Applying wavelet transforms to 2-D image

The two-dimensional wavelet transform is a straight-forward extension of the one-dimensional discrete wavelet. In this case, the vertical and horizontal spatial positions in the image correspond to time in a conventional signal, while the greyscale intensity corresponds to the amplitude of the signal. Moreover, the wavelet transform is performed by applying the technique to a matrix instead of a vector, i.e. the row or column vector of the pixel intensities of the images are considered the signal. It is performed by allowing the rows in the image to pass through a series of low pass and high pass filters. Calculation of wavelet coefficients are performed for every pixel in the image. Moreover, a columnwise decimation is applied on both matrices, such that a retention of one in two columns is ensured. Filtering of two columnwise decimated matrices are then executed but this time, in a row-wise fashion wherein one in two rows is retained (Aldrich, Marais, Shean, & Cilliers, 2010).

Figure ***3-4*** shows the j'th level two-dimensional discrete wavelet decomposition of an image. Initialisation of the algorithm takes place by setting $cA0$ as the original image. A low pass filter w_L and a high pass filter w_H is constructed from the scaling filter w. The image is *convolved* with these filters and down sampled. Convolution of an image I with any filter ω, yields a convolutional output c with the same dimensions as I. An element in the convolutional output, $\boldsymbol{c}(y_p, y_q)$, is defined as

$$\boldsymbol{c}(y_p, y_q) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(i,j)\omega(y_p - i, y_q - j) \qquad \text{(3-29)}$$

The computational procedure is illustrated in **Figure** *3-4*. At the j'th level of approximation, the horizontal wavelet coefficients $\boldsymbol{c}H_j$ are calculated as follows. First, the approximation coefficients $\boldsymbol{c}H_{j-1}$ are convolved with the low pass filter wL in a row-wise manner, after which the result is down sampled by keeping the evenly indexed columns. These columns are shown in **Figure** *3-4* by circles. The down sampled results are convolved column-wise with the high pass filter wH, after which the rows are down sampled by retaining the evenly indexed rows. This yields the horizontal approximation coefficients, $\boldsymbol{c}H_j$.

As indicated by the figure, the approximation image $\boldsymbol{c}A_{j-1}$ is decomposed into four sets of coefficients at each level *j*. That is, the approximation coefficients $\boldsymbol{c}A_j$ and the horizontal, vertical and diagonal detail coefficients ($\boldsymbol{c}H_j, cV_j$ and $\boldsymbol{c}D_j$). Decompositions at the next level *j*+1 is based on the same procedure applied to $\boldsymbol{c}A_j$. In wavelet packet analysis, decomposition of all four approximations and detail images are done, but this can rapidly lead to a very large number of features when *j* > 2.

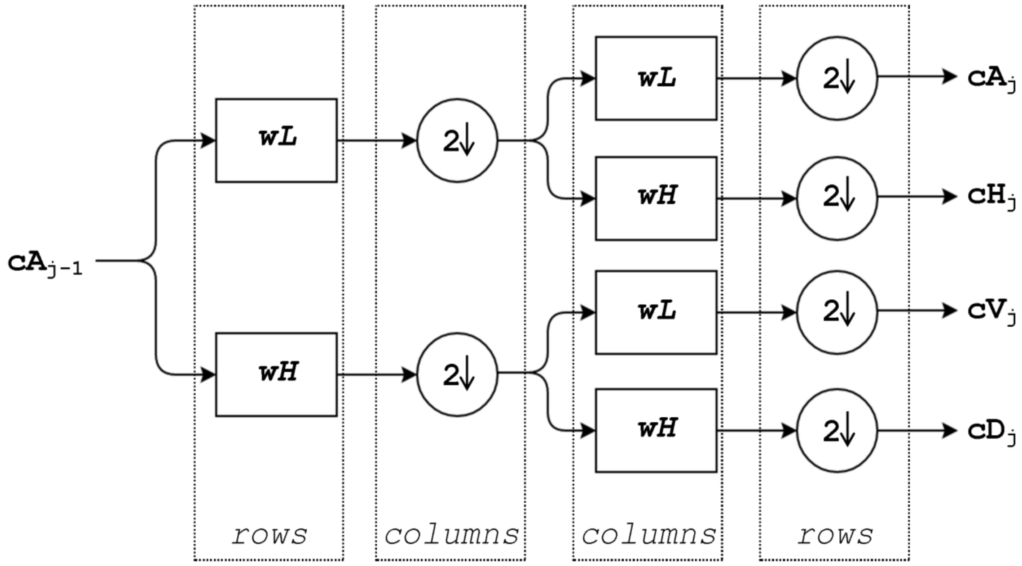


Figure 3-4. Discrete two-dimensional wavelet decomposition at level j. cA, cH, cV and cD refer to approximation and detail coefficients. wL and wH refer to the high pass and low pass filters, respectively. The circles containing "2" and a downward arrow indicate down sampling of the coefficients by retaining only every other row or column.

3.6 Local Binary Patterns

Initially proposed by Ojala, Pietikainen, and Harwood (1994), Local Binary Pattern (LBP) is a high-order statistics which captures both structural and statistical information of an image by comparing the intensities of its pixels to its pixel neigbourhood. It has already been successfully applied to several fields, including medical science ((Abbasi & Tajeripour, 2017; Rana et al., 2017), face recognition ((L. Liu, Fieguth, Zhao, Pietikäinen, & Hu, 2016), and remote sensing ((Vatsavai, Cheriyadat, & Gleason, 2010).

3.6.1 LBP Features

Generally, the LBP features are the frequency histogram of the LBP image after performing LBP calculation. The LBP operation consists of three major steps: pixel comparison to its pixel neighbourhood, transforming into thresholded values, and conversion of the neighbouring pixel values to powers of 2.

In the first step, each of the pixels of the greyscale images is compared to its local pixel neighbourhood using the LBP operators. Then, a binary thresholding function s is applied to the eight neighbouring pixels by comparing their intensities g_p $(p = 0,1, \dots, P - 1)$ to the intensity of the centre pixel (g_c):

$$\boldsymbol{s}(g_p - g_c) = \begin{cases} 1, & g_c \geq g_p \\ 0, & g_c < g_p \end{cases} \tag{3-30}$$

Once the thresholded values are calculated, the resultant LBP is then computed by converting the neighbouring pixel values to powers of 2 according to location and summed for the centre pixel, as shown in eqn *(3-31)*.

$$\mathrm{LBP} = \sum_{\mathrm{p}=0}^{\mathrm{P}-1} 2^{\mathrm{P}-1-\mathrm{p}} \boldsymbol{s}(\mathrm{g_p} - \mathrm{g_c}) \tag{3-31}$$

In essence, the “LBP image” is referred to the resultant LBPs ranging from 0 to 255. This is obtained after applying the LBP operator to each pixel in an image. Subsequently, the histogram of the LBP image is then calculated, which is considered as the LBP features of an image.

As an example, **Figure 3-5** shows the LBP operation of a single pixel of an image. The shaded part is the center pixel while the unshaded parts are its neighbouring pixels. In this example, the center pixel has an intensity value of 6. LBP operation is applied to get the thresholded values in Figure 3-5b. For the top-left corner, a thresholded value of 1 is obtained by first comparing 6 to

8 in Figure 3-5b. Since 8 is greater than 6, and by using the thresholded function, 1 is the result when the neigbouring intensity values are greater than that of the center pixel. Furthermore, the conversion weights are computed for the center pixel by summing up all its neighbouring pixel values that were converted to power of 2 (i.e. 186 = 128 + 32 + 16 + 8 + 2)

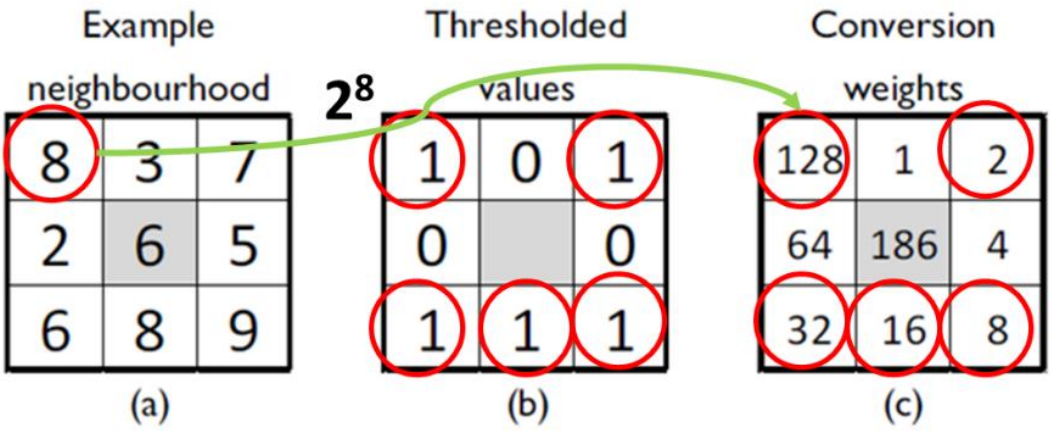


Figure 3-5. Local binary pattern operations, showing (a) the intensity values of the centre pixel (shaded) and its neighbours in the original image, (b) the corresponding thresholded values, and (c) the neighbouring pixel values converted to powers of 2 according to location and summed for the centre pixel.

3.6.2 Others LBP Operators

Extensions and some variations of LBP operators have been developed over the past few years. In the celebrated work of Ojala, Pietikainen, and Maenpaa (2002), they proposed several extensions of LBP. These are multi-scale representation, rotational invariance and proper representation of uniform patterns.

In multi-scale representation, a circular neighbourhood of radius R with equally spaced pixels P is used. Using the center pixel as the center of the circle, the comparison is facilitated by the circular neighbourhood that lie on its circumference. In this LBP variation, the LBP features are obtained by calculating the LBP image and subsequently concatenating the LBP histograms of all the LBP images into a single feature vector (Kistner, 2013).

In the rotational invariance, the transformation is performed by allowing each LBP to rotate from a reference position. This is done to ensure that all rotated versions of a binary number are the same (Kistner, 2013). In other words, the rotational invariance (LBP^{ri}) with respect to P and R can be defined as

$$LBP_{P,R}^{ri} = \min\{ROR(LBP_{P,R}, i) \,|\, i = 0, 1, \dots P-1\} \tag{3-32}$$

where *ROR(n,i)* takes the P-bit binary number *n* and rolls it *i* times to the right.

Lastly, as an extension of rotational invariance, the representation of uniform patterns is also developed (Kistner, 2013; Ojala et al., 2002). In this LBP variation, the uniformity measure U is computed using the eqn (3-33) as 0/1 and 1/0 transition in a binary pattern.

$$\begin{aligned} U(LBP_{P,R}) = {} & |s(g_c, g_{p-1}) - s(g_c, g_0)| \\ & + \sum_{p=1}^{P-1} |s(g_c, g_p) - s(g_c, g_{p-1})| \end{aligned} \tag{3-33}$$

Incorporating this uniformity measure U to the rotationally invariant texture invariant, as exemplified by eqn (3-34), the LBP image is then computed. Subsequently, the feature vector is obtained with P+2 dimensions, which serves as the LBP feature.

$$LBP_{P,R}^{ri} = \begin{cases} \dfrac{Z(LBP_{P,R})}{P+1} & \begin{matrix} if\ U(LBP_{P,R}) \leq 2 \\ otherwise \end{matrix} \end{cases} \tag{3-34}$$

where *Z* is a function that returns the number of zeros in a binary pattern.

3.7 Textons

The textons algorithm is one of the methods that is used to extract the features from image data using a filtering approach. The concept behind this algorithm is derived from alternative texture perception model, which believes that the texture discrimination is governed by a local process called textons instead of a global process that is governed by well-defined higher-order statistical properties, i.e. image pixel intensities (Julesz, 1984). Considered as textural 'primitives' including blobs, edges, line terminators and line crossings, textons only use its local changes of its density to discriminate between textures (Julesz, 1984; Kistner et al., 2013). This revolutionary concept carried the theory of texton into the attention of some researchers in the fields of machine vision.

The texton theory attained considerable achievement in binary images of synthetically generated textures (Bela, 1981), however, the theory was less preferred over other models such as filtering approach, during that time due to its deficiency in operational definition for greyscale images. Nevertheless, in the work of Leung and Malik (2001), the theory of texton has regained its popularity when they merged the concept with filtering approaches. This development gave birth to the texton algorithm. In this sense, the textons were redefined as cluster centers in a filter response space. Since the algorithm is a type of filtering approach, a number of researchers have attempted to optimise the algorithm by examining the use of different filter banks ((Leung & Malik, 2001; Cordelia Schmid, 2001; Varma & Zisserman, 2005). Moreover, researches on the application of this algorithm have grown considerably on different fields, including medical science (Fernández-Carrobles et al., 2015; Sudarshan et al., 2016), remote sensing (Zeki Yalniz & Aksoy, 2010), minerals processing (Kistner et al., 2013) and industrial inspection (Farinella, Allegra, Moltisanti, Stanco, & Battiato, 2016).

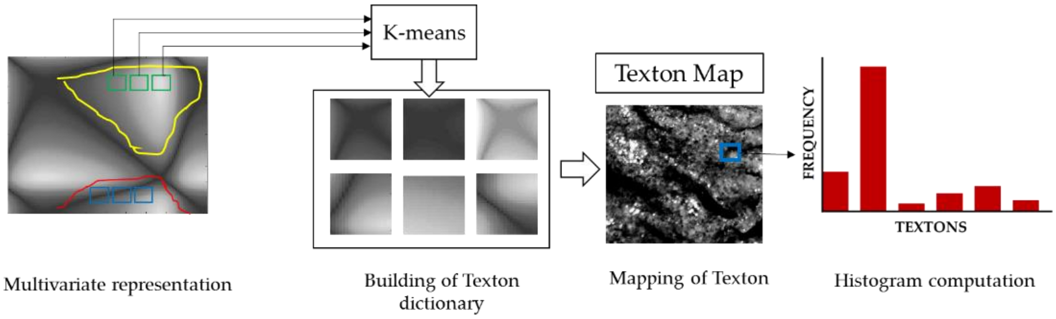


Figure 3-6. Learning steps involved in extracting texton features

In general, the textons are learned in the training stage by several steps including multivariate representation, building of texton dictionary, and histogram computation (Kistner, 2013; Rampun, Zheng, & Malcolm, 2015). The textons are defined by performing vector quantization e.g. k-means clustering (Hastie, 2009) on the achieved filtered responses of the images from the training set. These responses are obtained through convolution with Nf oriented linear spatial basis functions arranged in a filter bank wherein each pixel is mapped to an Nf-dimensional feature space. The resulting textons from each class N are then pooled together to constitute the universal textons or the texton dictionary. Filter responses for each image are subsequently mapped onto the texton dictionary resulting in a texton frequency representation or histogram, which eventually provides the learning model.

Essentially, for each training image *I*, the number of pixels in *q*-th texton channel would provide the *q*-th histogram bin. That is,

$$h_I(q) = \sum_{j \in I} I[T(\mathrm{j}) = q] \tag{3-35}$$

where *I* [.], and *T(j)* are the indicator function and the texton channel assigned to pixel *j*, respectively (Jemwa & Aldrich, 2012). This histogram of textons are then considered as the textural features extracted from the set of images (Kistner et al., 2013; Varma & Zisserman, 2005).

It should be noted that during the K-means clustering step, the texton algorithm usually incur iterative operations in computing distances between all pixels and their closest cluster centers, making it computationally extensive and thus, involve longer computer running times. Moreover, the overall performance of the algorithm is highly dependent on the design and choice of filter bank, which is why the selection of suitable filter banks should also be taken into account. In this work, three filter banks are considered namely, Schmid, Leung-Malik, and Maximum Response Filter Set, which are discussed in this subsection.

3.7.1 Leung-Malik Filter

The Leung-Malik (LM) filter bank is designed to have a good texture discrimination characteristic. It consists of 48 filters (36 oriented filters, and 8 center-surround Laplacian of Gausssian or LoG filters, and 4 low-pass Gaussian filters). With the notion that both rotational and spectral selectivity are covered in the transformation, this type of filter bank then becomes less powerful to reflect some variations in imaging conditions such as in-plane rotations and perspective changes (Jemwa & Aldrich, 2012; Leung & Malik, 2001).

3.7.2 Schmid Filter Bank

The Schmid filter set, as proposed by C. Schmid (2001), uses a two-layer representation to capture 'texture-like' visual structures efficiently. The layers consist of descriptors and the joint probability on the frequencies of these over neighbourhoods. In essence, these descriptors are the sets of similar rotational invariant feature vectors. In other words, this filter set worked similar to Gabor filter which is given by the equation below on the transformation of each pixel location. This filter has complete rotational symmetry but lack sensitivity to anisotropic features. (Jemwa & Aldrich, 2012).

$$F(x, y, \dot{t}, \sigma) = F_0(\dot{t}, \sigma) + \cos(\frac{\pi\dot{t}\sqrt{x^2 + y^2}}{\sigma}) \exp(-\frac{x^2 + y^2}{2\sigma^2}) \quad (3\text{-}36)$$

where $\dot{t}$ is the cycle count in the harmonic function and $F_0(\dot{t}, \sigma)$ is the zero DC component guarantor. Note that to avoid high frequency responses, smaller $\dot{t}$ is used at small scales.

3.7.3 Maximum Response or Root Filter Set

To address the lack of robustness in variations in Leung-Malik filter bank, the maximum response filter set is proposed (Varma & Zisserman, 2005). In this type of filter bank, the discriminative power-invariance trade-off is now incorporated. This is performed by retaining the maximum filter responses in all orientations and scales with the help of a base filter set, a filter set with 3 subfilters. Unlike Leung-Malik filter bank, this type of filter set only has 38 filters because only the maximum responses for all subfilters are retained and considered.

This type of filter set demonstrates few advantages compared to Leung-Malik and Schmid. The advantages include incorporation of both isotropic and anisotropic features, its low dimensionality characteristics, and its capability to include the maximum response angle, which is valuable in discriminating between very similar textures (Jemwa & Aldrich, 2012).

3.8 Convolutional Neural Networks

A convolutional neural network is a deep learning algorithm that is inspired from the natural visual perception mechanism of animal cortex. It belongs to a class of deep, feed-forward artificial neural networks which simulate the activity in layers of neurons in the neocortex, which is responsible for detecting lights in the receptive fields (Ferreira & Giraldi, 2017; Gu et al., 2017). In computer vision, CNN is said to have a power to capture complex features from image data. Its potential is underscored by excellent results in image classification which can outperform human accuracy in some cases, and the reason why it is receiving growing interest in the research community.

As an example, CNNs has reached a significant milestone in the ImageNet Challenge (Russakovsky et al., 2015) where it achieved comparable error rates to those of a well-trained human in classifying 100,000 images that were only seen by the model for the first time. The ImageNet Challenge is an annual image classification competition wherein some 1.2 million images are available for training the models to recognise any of a 1000 everyday objects in diverse settings and 100,000 images for testing the models. As seen in **Figure 3-7**, the classification error rates drastically decreased through the years and even surpassed the trained human with error rate of 5.1% beginning 2015.

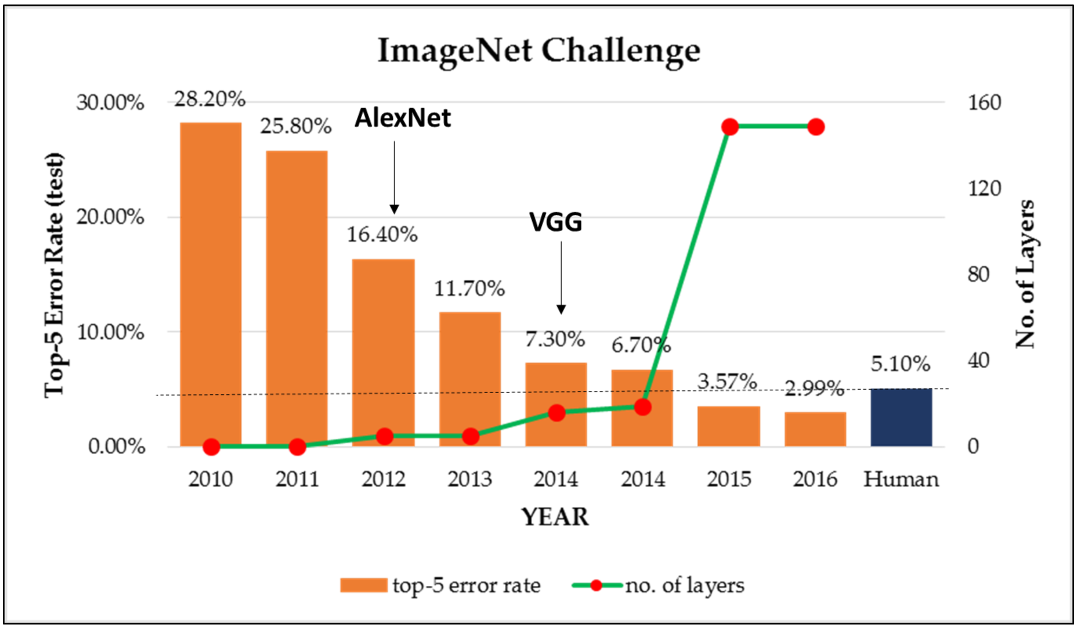


Figure 3-7. Historical performance of ImageNet Challenge from 2010 to 2016 showing both the error rate and the number of layers of the networks.

3.8.1 Basic Components of CNN

As presented in **Figure** 3-8, the architecture of CNN is normally made up of five groups of layers, viz. an input layer, convolutional layers, pooling layers, fully-connected layers, and an output layer.

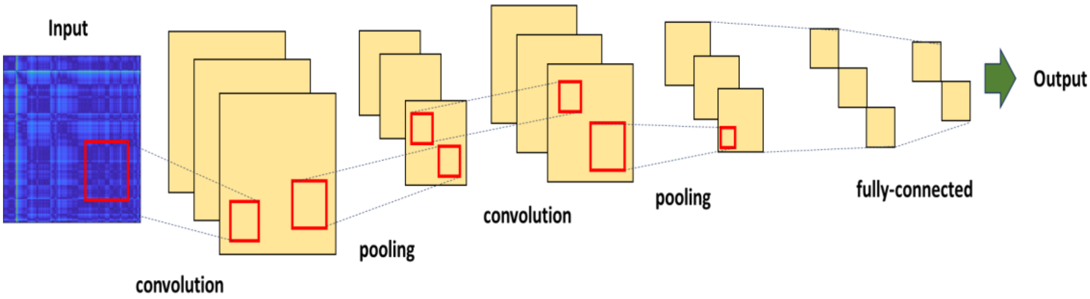


Figure 3-8. Typical CNN architecture

Input layer

The input layer consists of the image data that need to be fed to the network. The data can be either raw image pixels or some image transformations through resizing of the image to a specific aspect required in the training of the network.

Convolutional layer

The convolutional layer is made up of a number of convolution kernels which is responsible for the computation of a so-called feature maps. The feature maps are generated two steps: first convolving the input with the learned kernel which is shared by all spatial locations of the input, and then applying an element-wise nonlinear activation function on the convolved results. It is worth to highlight that the neuron of a feature map must be connected to a region of neighbouring neurons in the previous layer. In general, the feature value, $z_{i,j,k}^{l}$, at location *(i,j)* in the *k*th feature map can be calculated by:

$$z_{i,j,k}^{l} = {\boldsymbol{w}_k^l}^T \boldsymbol{x}_{i,j}^{l} + b_k^l \tag{3-37}$$

where $\boldsymbol{w}_k^l$ and b_k^l are the weight vector and bias term of the *k*th filter in the *l*th layer, respectively and $\boldsymbol{x}_{i,j}^{l}$ is the input at location *(i,j)* of the *l*th layer. The feature map also uses activation functions, which is required for multi-layer networks to detect nonlinear features.

Pooling layer

Pooling is an important concept of CNN which is responsible for its computational efficiency through reduction of the number of connections between convolutional layers. The pooling layer ensures that the shift-invariance of the feature maps are achieved. This is normally done by reducing the resolution of the feature maps, that is, through summarisation of the data by sliding window across the feature maps and application of some linear or nonlinear operations, e.g. local mean or max, and by reduction of the its dimensionality. Thus, for each feature map $\boldsymbol{z}^l_{m,n,k}$, the feature value $y^l_{i,j,k}$ can be obtained from eqn (3-38).

$$y^l_{i,j,k} = pool(z^l_{m,n,k}), \forall(m,n) \in \mathcal{R}_{i,j} \tag{3-38}$$

where $\mathcal{R}_{i,j}$ is a local neighbourhood around location *(i,j)*. Normally, the pooling layer is situated between the two convolutional layers.

Fully-connected layer

In the fully-connected layer, the neurons in this layer are in full connections to all activations in the previous layer. This is purposely performed to extract high-level reasoning of the patterns generated by the previous layers. More often, the activations uses rectified linear unit (ReLU) to facilitate training of CNNs. This is done by applying non-linear functions to the outputs for faster convergence. Moreover, transfer learning method can also be used in this layer to extract features which can be then be used for classification.

Output layer

The output layer specify how the network errors are penalised during training of the network. Depending on the tasks, this layer uses several loss functions such as softmax, Euclidean loss, and Sigmoid cross-entropy.

3.8.2 Pretrained CNN with transfer learning

CNNs are preferred over traditional fully connected neural networks owing to the convenience in training fewer parameters by use of backpropagation and stochastic gradient descent (Rumelhart, Hinton, & Williams, 1986). Extraction of high-level and complex features can be effectively carried out using convolutional layers with a small kernels which are then fed to fully connected layers. Although considered a powerful state-of-the-art algorithm in extracting patterns in image data, the training of CNNs usually requires thousands or even millions of labelled data in many domains. This is a massive challenge and labelling those data is considered costly (Gopalakrishnan, Khaitan, Choudhary, & Agrawal, 2017), on top of the actual training of these data which is computationally expensive.

One of the solutions to this drawback is the use of transfer learning. Transfer learning is machine learning which uses additional source of information from a pretrained network, apart from the standard training of the actual data. It requires the transfer of knowledge from a source task to a target task. In a sense, this drives the CNNs to transplant the learned feature layers from one CNN to initialise another which does not necessarily need the source data; only the source concept is required (Afridi, Ross, & Shapiro, 2018). With this concept, it allows other researchers to share and reuse the previously learned CNN models freely, and as a result, considerable cost of developing large CNNs can be avoided (Y. Fu & Aldrich, 2018).

Given these advantages, the CNN with use of transfer learning is then explored in this thesis. More particularly, the use of 'off-the-shelf' CNNs namely AlexNet and VGGNet, which are both pretrained on large-scale ImageNet datasets are considered. These algorithms are used in extracting features from distance matrix plots.

3.8.3 AlexNet

Developed by Krizhevsky, Sutskever, and Hinton (2012), AlexNet is a convolutional neural network which won the ImageNet Large-Scale Visual Recognition Challenge (ILSCRC) in 2012 with an achievement of 16.4% top-5 test error rate. This model is trained using this ImageNet dataset which consist of millions of variable-resolution images. It uses non-saturating neurons and efficient GPU implementation of the convolution operation for efficient training of the network, while a drop out method is implemented in the first 2 fully connected layers to minimise overfitting (Krizhevsky et al., 2012). Dropout method is done by setting the output of each hidden layer with probability of 0.5 to zero. As a result, drop out neurons do not participate in back propagation and do not contribute to the forward pass, which ensued different architecture of neural network for every input while all the architectures share weights.

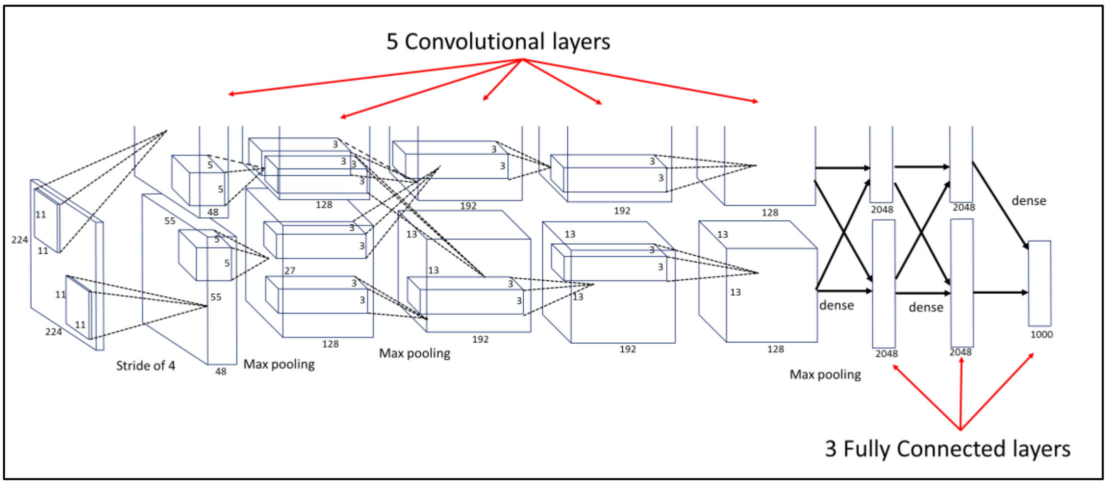


Figure 3-9. The architecture of AlexNet

AlexNet is consist of 5 convolutional layers, 3 max pooling layers and 3 fully connected layers. The architecture of this CNN along with the sizes of each layer is shown in **Figure 3-9**. The input images entail a fixed resolution of 256 x 256 pixels. The first convolutional layer contains 96 filters of size 11 x 11. Its stride is 4 pixels and its padding of 2 pixels while the remaining four convolutional layers have both stride and padding equal to 1 pixel. The size of the second convolutional layer is size 5 x 5 while the remaining three layers all have 3x3 sizes. The second and fifth convolutional layers both have 256 filters while the third and the fourth convolutional layers have 384 filters. Moreover, the last fully connected layer connects to a 1000 classes rest of the network can be considered as a feature extractor. AlexNet can then produce 4096 feature set for each image, which contains the activations of the hidden layer immediately preceding the output layer.

3.8.4 VGG16

VGGNet is an ILSCRC 2014 entry which won second place with 7.3% top-5 error rate. The configuration of the network is the same with AlexNet with much deeper architecture. VGG16 consists of 13 convolutional layers, 5 pooling layers and 3 fully connected layers. Its architecture is simplified in **Figure *3-10***.

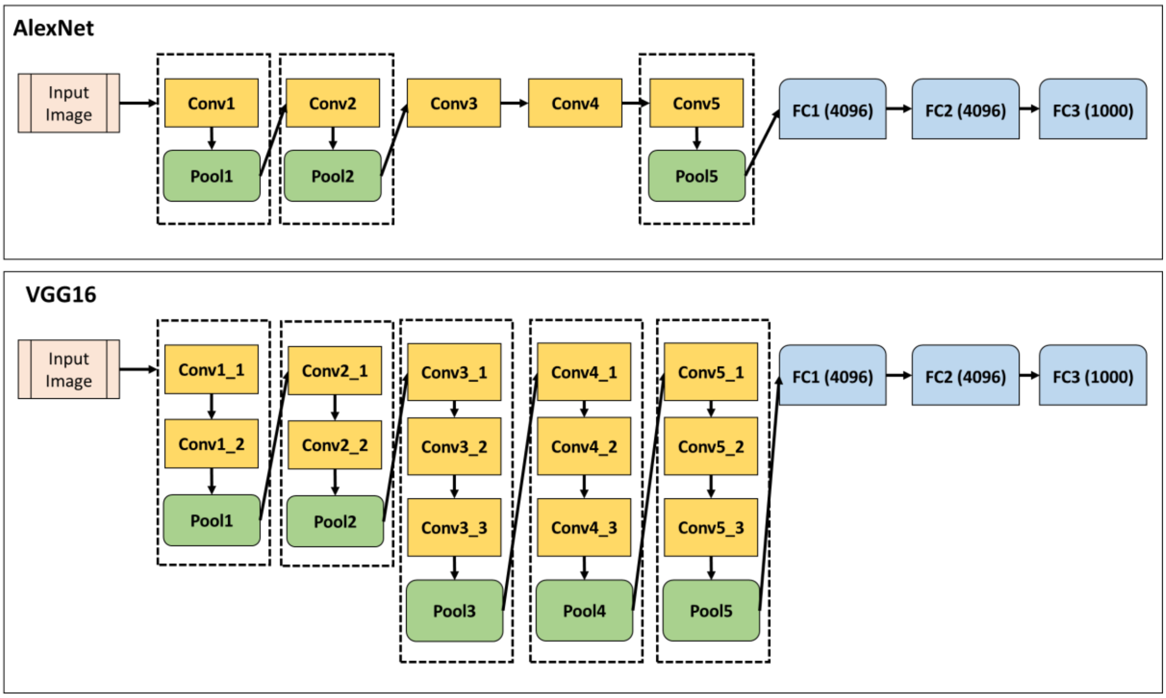


Figure 3-10. The simplified diagrams showing the architectures of AlexNet (top) and VGG16 (bottom). It highlights the similarity and differences in terms of the structures of convolutional (Conv), pooling (Pool), and fully-connected (FC) layers.

The input data of this network requires a fixed-size of 224 x 224 RGB image. All convolutional layers in VGG16 has stride and padding set to 1 pixel. Moreover, each group of convolutional layer has pooling layer which is carried out over a 2 x 2 window with stride 2. The number of filters of convolutional layer group starts from 64 in the first group and then increases by a factor of 2 after each pooling layer, until it reaches 512 (Simonyan & Zisserman, 2014). The fully connected layers are the same with AlexNet, thus, it also produce 4096 features for each image.

In this study, both the pretrained AlexNet and VGG16 network are used for feature extraction of the distance matrices. All the implementation of CNNs are carried out in MATLAB® R2017b. Prior to feature extraction, all images are preprocessed by automatically resizing to [227 227] and [224 224] for AlexNet and VGG16, respectively.

3.9 Final Remarks

In this chapter, the detailed description of the recurrence texture analysis, with emphasis on the discussion of the involved steps and on the textural feature extraction algorithms, were presented. In order to validate the applicability of RTA, the evaluation on the method including the impact of the parameters involved in the process and the application of the method to dynamic process system datasets should be carried out thoroughly. In other words, the next step is to evaluate its applicability to deal with problems involving dynamic process systems, which include capturing the dynamics of the time series data. In the following 4 chapters, RTA is applied to several systems requiring the characterisation of its dynamic behaviour. As an alternative to RQA, the RTA features could then be used to explain several occurrences of the dynamic process systems wherein the structure of its time series could be analysed with the method, and can be extended for developing prediction models. However, prior to actual application, it is first necessary to evaluate the influence of the parameters involved in RTA as part of the optimisation of the method. As seen in this chapter, RTA requires calculation of distance matrix which uses a particular distance metric. Moreover, the segmentation of the time series entails windowing parameters which may affect how the textural features. These queries need to be addressed prior to actual application. In the next chapter, the influence of these RTA parameters (e.g. distance metrics, windowing parameters) are investigated. Additionally, the results are also compared to RQA.

4. EVALUATION OF RECURRENCE TEXTURE ANALYSIS

4.1 Introduction

The recurrence texture analysis is comprised of steps that require input of parameters, i.e. distance metrics, window width *(b)*, sliding step *(m)*, hyperparameters of textural algorithms. Most of the time, these parameters are usually defined by the user, although there are expressions that could serve as guides in choosing values for these parameters.

The effect of segmentation parameters (window width and sliding step) and the distance metrics to the overall performance of the method in capturing the dynamic behaviour of the system are explored in this chapter. Several sets of parameter values are considered in the models representing different scenarios, e.g. high, normal, and low. Two different distance metrics are considered, namely the Euclidean distance and the Chebychev distance, since these are the most commonly used when quantifying the RPs using RQA. Moreover, RQA is also implemented and compared with the results to RTA. Several threshold values are taken into account in this matter. In this undertaking, the time series from the Lotka – Volterra predator-prey system is employed as benchmark test dataset in this analysis.

Both the proposed method and RQA are evaluated from their power to cluster and classify data. Qualitative investigation via visualisation of data is carried out to assess the quality of clusters, and to explain its reliability to cluster features of the same class. Data visualisation is performed by projecting the features into 3-D subspace using their first three principal component scores. Moreover, classification is also implemented to quantify clusters and evaluate the classification performance of the method. The SVM is used as classifier to train and classify the data, and the classification accuracy is calculated for each scenarios.

4.2 Lotka – Volterra predator – prey system

The Lotka-Volterra model, as proposed by Lindfield (2012) describes the interaction between predator and prey species, which is based on the following differential equations:

$$\frac{d\dot{x}}{dt} = k_1\dot{x} - \hat{C}\dot{x}\dot{y} \tag{4-1}$$

$$\frac{d\dot{y}}{dt} = -k_2\dot{y} + \widehat{D}\dot{x}\dot{y} \tag{4-2}$$

where $\dot{x}$ and $\dot{y}$ are the number of prey and the predator, respectively, and the k$_1$ and k$_2$ are the prey population growth rate and predator mortality rate, respectively. Lastly, $\hat{C}$ corresponds to the reaction rate or the frequency of the contact of the prey and predator, and $\widehat{D}$ is the conversion efficiency or the efficiency of predators in converting food into offspring. In this study, however, only a univariate time series was considered. It was simulated with the ODE45 subroutine in MATLAB R2015a by considering the change in z_1 with time.

Throughout the simulation, the number of prey ($\dot{x}$) and predator ($\dot{y}$) are held constant. Only the values of k_1, $k_{2,}$ $\widehat{D}$ and $\hat{C}$ are varied to obtain two change conditions. The parameters used in the simulation are summarized in Table 4-1.

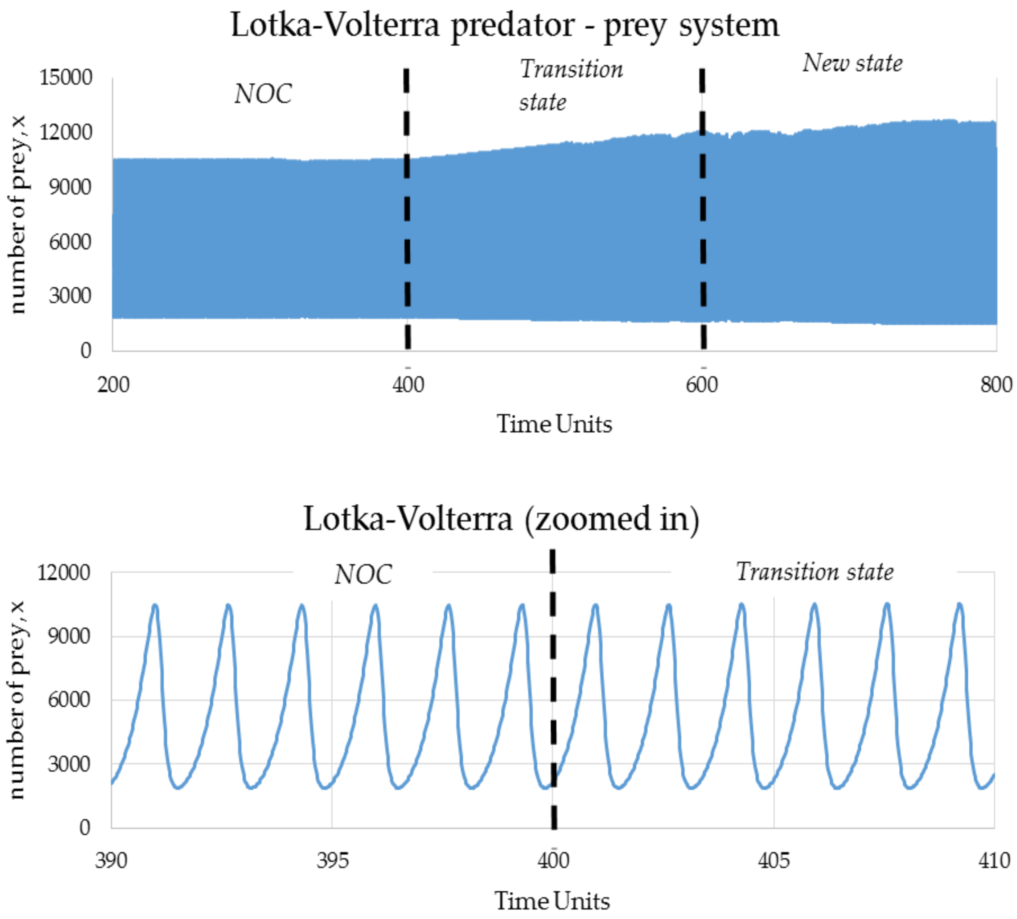


Figure 4-1. Simulated observations of the Lotka-Volterra predator-prey model.

The simulation as shown in Figure 4-1 was carried out with equal time steps of 0.02 time units. 20 000 samples were generated with the parameters shown in the first column of Table 4-1, another 10 000 samples were generated with the parameters shown in the second column of Table 4-1. These parameters changed linearly over the time interval to yield data corresponding to a state of transition. Finally, another 10,000 samples were generated with the parameters shown in the last column in Table 4-1. This was considered to be a new state of the system.

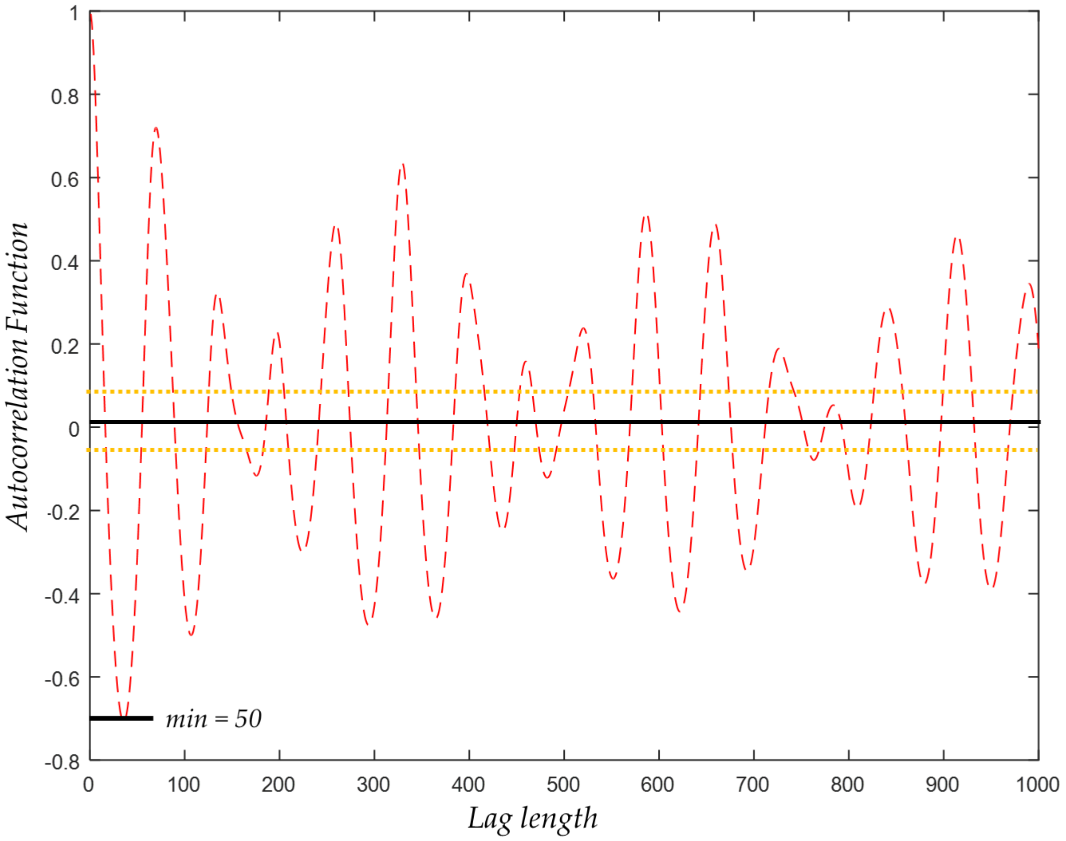


Figure 4-2. Autocorrelation function of Lotka-Volterra Predator-prey system

In the evaluation, it should be noted that the first 10,000 samples in the NOC was eliminated to ensure that the NOC system being considered is in steady-state. Thus, each state contains 10,000 samples for analysis.

Table 4-1. Summary of the parameters used in three condition states (NOC, state of transition and new state)

Parameters	**NOC**	**Transition**	**New State**
$\dot{x}$	5,000	5,000	5,000
$\dot{y}$	100	100	100
k_1	2	2 to 3	3
k_2	10	10 to 11	11
$\hat{C}$	0.0010	0.0010 to 0.0011	0.0011
$\widehat{D}$	0.0020	0.0020 to 0.0021	0.0021
Time units	0 to 400	400 to 600	600 to 800

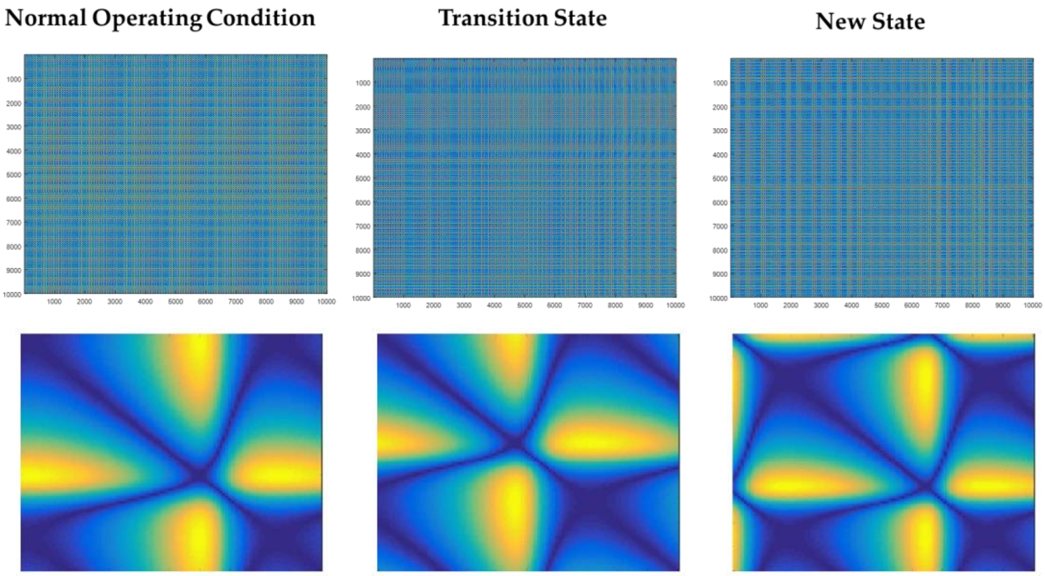


Figure 4-3. Distance matrix plots of three states of the system using Euclidean distance. Top and bottom plots show the *10,000-by-10,000* and *100-by-100* matrix plots of the time series, respectively.

4.3 Effect of Window Width and Step Size

Three different values of window width *b* (50, 100, 250) are considered in this case study using Euclidean distance and a fixed value of sliding size *m* of 50. It should be noted that the window width *b*=50 are segmented in a fixed window fashion. To evaluate further, the influence of window width with the same kind of windowing approach was also facilitated. In other words, the other window width values were also segmented in a fixed windowing fashion. Thus, five different window width – sliding step combination were considered in this study.

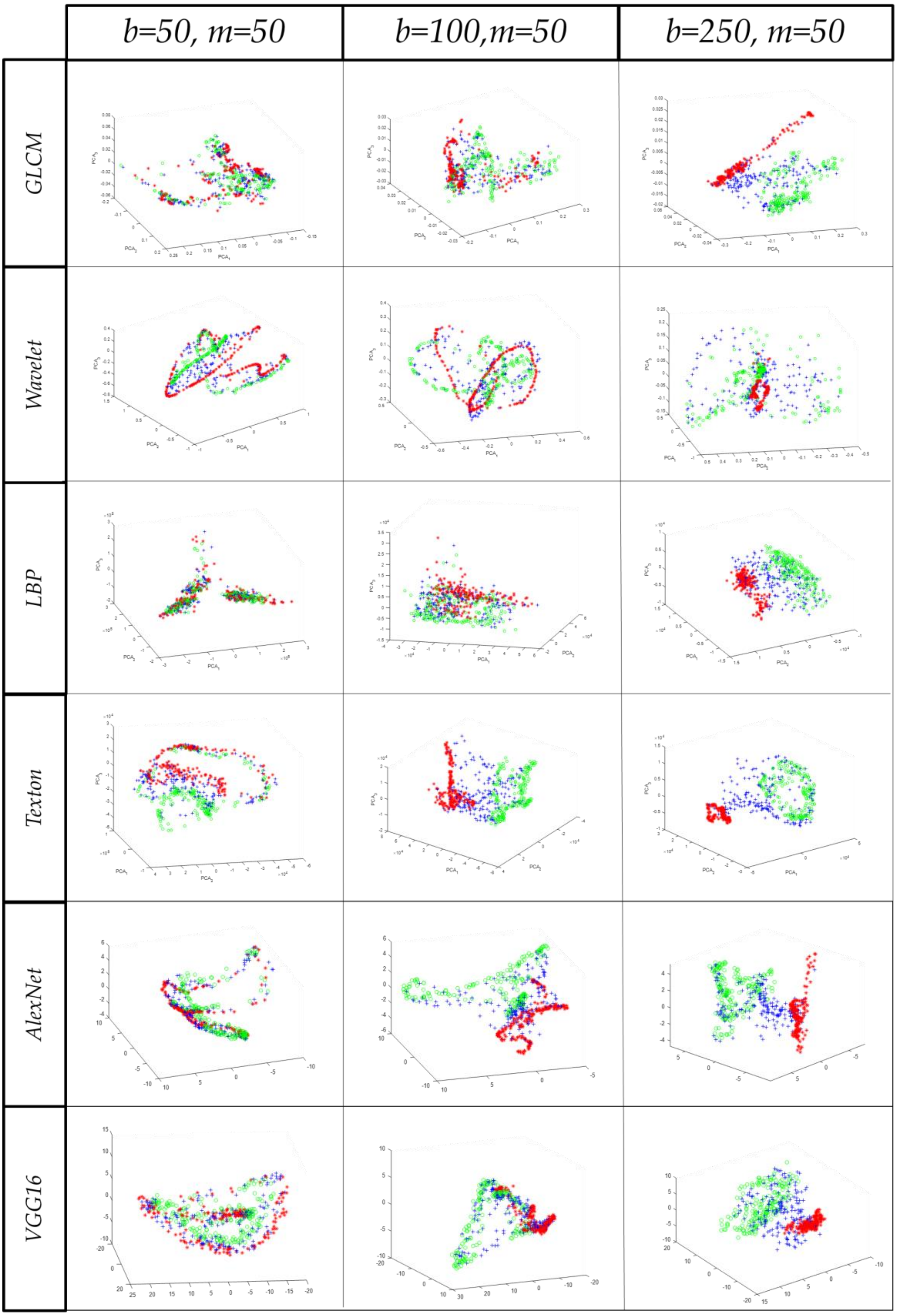


Figure 4-4. RTA feature sets projected in 3-D principal component subspace, using different values of window width *b*: *b*=50 (left), *b*=100 (middle), and *b*=250 (right). The distance matrix is constructed using Euclidean distance with fixed sliding step

m=50. The legends red (*), blue (+) and green (o) correspond to NOC, state of transition, and new state, respectively.

Based from the plots in Figure 4-4, it seemed that the distinction of clusters and its separation to other clusters become more visible as the window width increases, which could also mean that the number of observations or distance matrix plots are fewer. For example, a huge difference in clusters can be discerned from the plots of the features from the segmented time series using *b*=50 and *b*=100. It is seen that the former could not give reasonable clusters of the states whereas the plot using the latter provided the better separation. In the case of *b*=100 and *b*=250, although a reasonable result is obtained in *b*=100, better clustering and separation of clusters were noticeably seen in *b*=250. In this case study, it can further inferred that for a fixed sliding step, there is a linear correlation between the clustering power of the features and window width. Thus, better clustering could be achieved using a longer window width. This observation is clearly seen in the classification results as summarised in Table 4-2. The four RTA features have greatly improved the classification performance from *b=50* to *b*=100 and it continued to improve in *b*=250.

Another way to look at the effect of window width is to segment the time series with same value with the sliding step. In other words, the windowing method is still fixed windowing approach but the number of observations in the segments varied. Once again, the data are visualised and are presented in Figure 4-5. It is obvious that the size of window width has a strong influence in the clustering of the features. The features are fairly clustered in *b*=100 while distinct clusters are observed in *b*=250. Moreover, better separation of clusters were also seen as window width becomes longer. These results were clearly showed in their classification accuracy as presented in Table 4-2. Thus, it is generally inferred that window width is an important parameter in RTA that needs to be chosen by the users appropriately in segmenting the time series. In other words, the choice of the value of this parameter should be taken earnestly since the features are dependent on how the parent time series data are represented into sets of discrete time series data.

As mentioned in the previous chapter, the autocorrelation function could be used as a guide in estimating this parameter. The minimum value of lagged length *b* where ACF is at the minimum could be the used as the minimum value of the window width. It should be noted that this is only a guide and there is no such strict rule in determining the value of this parameter. Moreover, the choice of *b* would greatly depends on the type of RTA applications and on the kind of structural changes the researchers are being investigated. If window width *b* is quite large, the probability of missing the changes of the time series is high, which could give false information. On the other hand, if *b* is too small, the outlier could indicate as a structural change thus also giving a false interpretation (Auret & Aldrich, 2010). Thus, the choice of *b* is also a trade-off between the information being examined and the number of features being considered. In the case of this study, although the minimum *b* to where ACF is at the first minimum is seen to be *b*=50, as shown in Figure 4-2, the results somehow gave poor clustering and classification.

However, the researchers should consider the window width *b* to be used is somewhat close to this value. Thus, it is reasonable to choose *b*=100 to be the window width in this point as this value gave fair clustering and improved classification performance as compared to *b*=50. One could argue that the choice of *b*=250 should be chosen as this window width provided the best results among them. This might be true, however, as mentioned elsewhere, the choice of *b* is dependent on the problem being addressed. This value could be used if the application is a purely classification problem. If the problem entails identifying the structural changes of the time series such as change-point detection problem, the shorter *b* should be selected. And since this study entails evaluation of separation of clusters between states as a means to detect the changes of the system, shorter *b* is an appropriate choice.

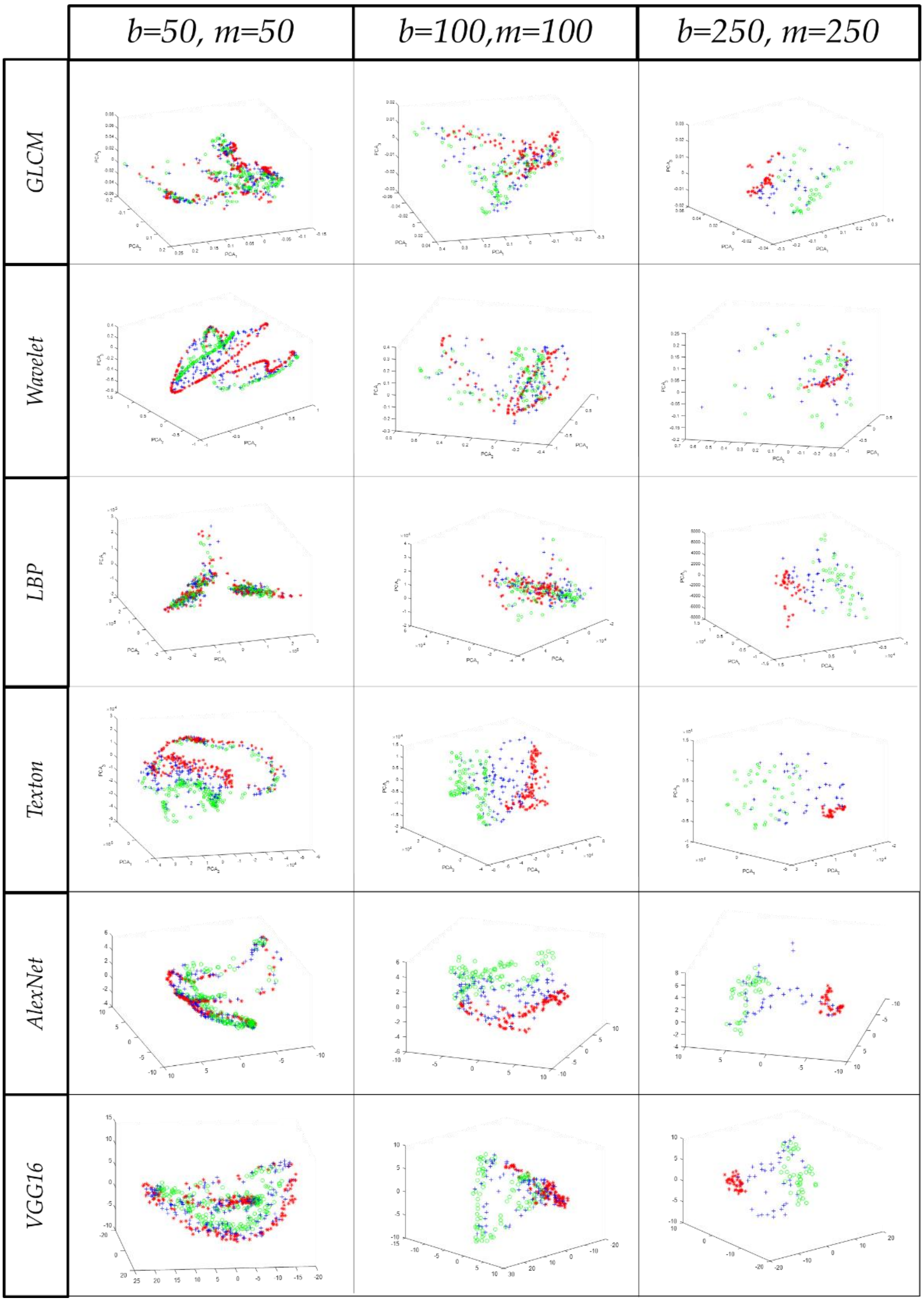


Figure 4-5. RTA feature sets projected in 3-D principal component subspace, using fixed windowing. The distance matrix is constructed using Euclidean distance with

three different window widths, *b*=50, 100 and 250. Legends are : red (*), blue (+) and green (o) correspond to NOC, state of transition, and new state, respectively.

The effect of sliding step is also investigated in this case study. As shown in Figure 4-4 and Figure 4-5, it is noticeable that data are not significantly affected by the sliding step.

For *b=100,* the clustering and the separation of the clusters of *m*=100 have seen slight to no change with *m=100*. This observation also held true to *b=250.* Aside from the fact that the number of observations increases when the sliding step *m* decreases, when it comes to clustering capability of the features, this parameter is considered not critical in RTA. This inference can also be seen in the classification accuracy as summarised in Table 4-2 wherein no general trend is observed in terms of their accuracy percentages. For example, a slight increase of accuracy percentages is observed in GLCM and LBP features when the sliding step is increased. This observation is inversely seen in wavelet and textons features. More importantly, the change is marginal that one could say that this is due to the change in the number of observations. With all these thoughts, it could then be understood that the sliding step is not a critical parameter in utilising RTA, other than providing different number of segments. In other words, the researchers could use either fixed or moving windowing approach which depends on the nature of time series being considered and on their computational capability. Shorter parent time series should entail moving window while longer time series data could be processed using fixed windowing.

Additionally, the selection of *m* is a trade-off between computation efficiency and the classification performance of the proposed method. Normally, *m* is estimated as half of the window width (*m=b/2)* to give a reliable capture on the information of the time series (Moskvina & Zhigljavsky, 2003).

Table 4-2. Influence of windowing parameters to the overall classification performance of RTA features (classification accuracy on the test data)

Window Width, b	$b = 50$	$b = 100$		$b = 250$	
Sliding step, m	$m = 50$ *(fixed)*	$m = 50$ *(moving)*	$m = 100$ *(fixed)*	$m = 50$ *(moving)*	$m = 250$ *(fixed)*
No. of segments, N	600	597	300	588	120
FEATURE SETS	**Classification Accuracy (%)**				
GLCM	37.8%	68.5%	68.9%	85.2%	88.9%
Wavelet	50.6%	78.8%	61.6%	89.8%	72.2%
LBP	61.6%	82.1%	79.0%	85.8%	86.1%
Texton	75.6%	93.9%	92.2%	97.2%	95.4%
CNN-AlexNet	73.1%	93.3%	93.1%	95.1%	97.1%
CNN-VGG16	73.9%	93.2%	93.0%	97.1%	97.3%

4.4 Effect of Distance Metric

The Euclidean distance and the Chebychev distance are used to study the effect of the distance metric to the performance of the proposed method. These distance metrics are both derived from Minkowski distance, as exemplified in eqn (4-3) using p=2 and p=∞ for Euclidean distance and Cheychev distance, respectively.

$$d_{ij} = \sqrt[p]{\sum_{i,j=1}^{n} \left\| x_i - x_j \right\|^p} \tag{4-3}$$

The time series of Lotka-Volterra predator prey system of three different system states, i.e. NOC, transition and new state, are segmented according to the parameters in Table 4-3. As a result, a total of 597 segments are obtained from this time series with 30,000 observations (10,000 for each state). These segments were processed into distance matrices using the said distance metrics. Further, texture features of these distance matrices were then extracted to form feature matrices, which were visualised in 3-D principal component subspace. These were also used as predictors in the classification analysis.

Table 4-3. Parameters used in the study on the influence of distance metric to the overall performance of RTA

Parameters	Values
Distance metric	Euclidean, Chebychev
Window width, *b*	100
Sliding step, *m*	50

Figure 4-3 and Figure 4-6 show distance matrices of the time series for each state calculated using Euclidean distance (Euclidean norm) and Chebychev distance (Maximum Norm), respectively. In general, it is difficult to pinpoint the differences of the plots calculated using these distance metrics. The textures of the plots for each state are somewhat indistinguishable for both macroscopic and microscopic viewpoints. For the texture comparison of the plots for states, it is apparent that the each state has its distinct recurrence textures. The distance matrix texture of the new state is mostly coarse while the new state appears to have finer textures.

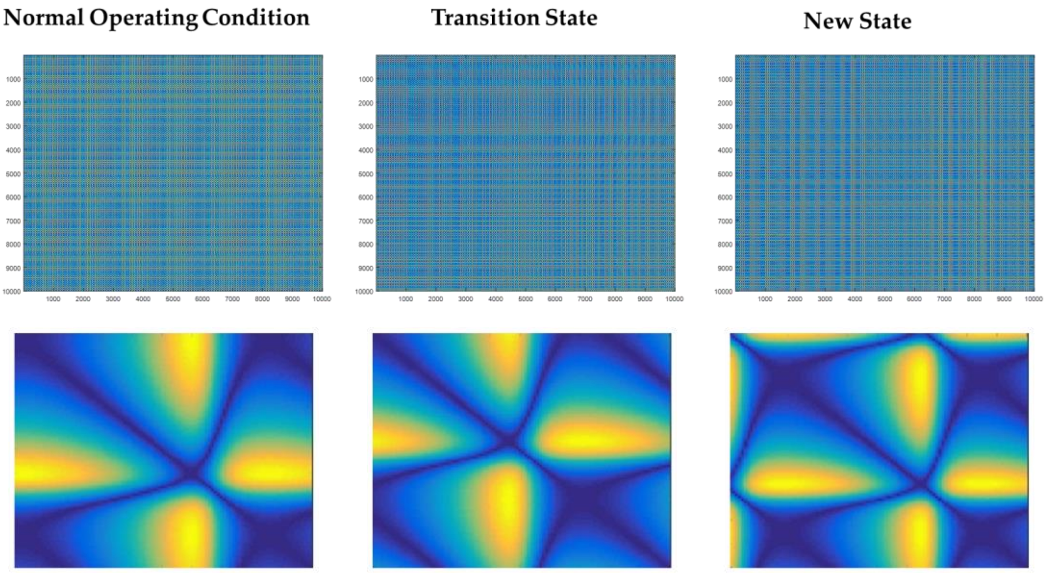


Figure 4-6. Distance matrix plots of three states of the system using Chebychev distance or Maximum Norm. Top and bottom plots show the *10,000-by-10,000* and *100-by-100* matrix plots of the time series, respectively.

Table 4-4. Comparison of the classification performance of the RTA and RQA feature sets using Euclidean and Chebychev distance metrics. The reported classification accuracy is based on the test datasets.

Feature Sets		Classification Accuracy	
		Euclidean Distance	**Chebychev Distance**
RTA	**GLCM**	68.9%	70.9%
	Wavelet	79.3%	73.2%
	LBP	83.2%	80.5%
	Texton	92.2%	90.5%
	AlexNet	93.3%	92.4%
	VGG16	93.2%	92.7%

RQA	**Using ε = 0.1**	83.2%	83.8%
	Using ε = 0.2	84.9%	84.4%
	Using ε = 0.5	77.7%	81.6%

As shown in Figure 4-7, the CNN-AlexNet, CNN-VG16, texton and LBP features showed distinct clusters representing the three states of the system. In LBP and VGG16, the separation is somehow less visible than the other feature sets since most of the transition state data are somewhat located in the NOC cluster. For the texton and AlexNet, on the other hand, the distinction of clusters and also its separation to other clusters are apparent but not totally separable.

With respect to the effect of distance metrics, it is observed that the data behaved similarly regardless of the kind of distance metrics. In fact, it is apparent that the clusters and its separation for all feature sets are quite alike.

It is interesting to note the superficial overlapping of the cluster of transition state data to both new and NOC clusters. Conversely, the NOC and the new state clusters, on the other hand, are entirely separable with each other. It could be said that this overlapping is due to the non-steady state behaviour of the transition state since the data is still in the stage of shifting from NOC to another state. This also happened to transition state and new state cluster overlapping, i.e. evolution of state behaviour from transition to new state. It could then be regarded that the transition state is a pseudo-steady state of the system, while the NOC and new state are already in the steady state.

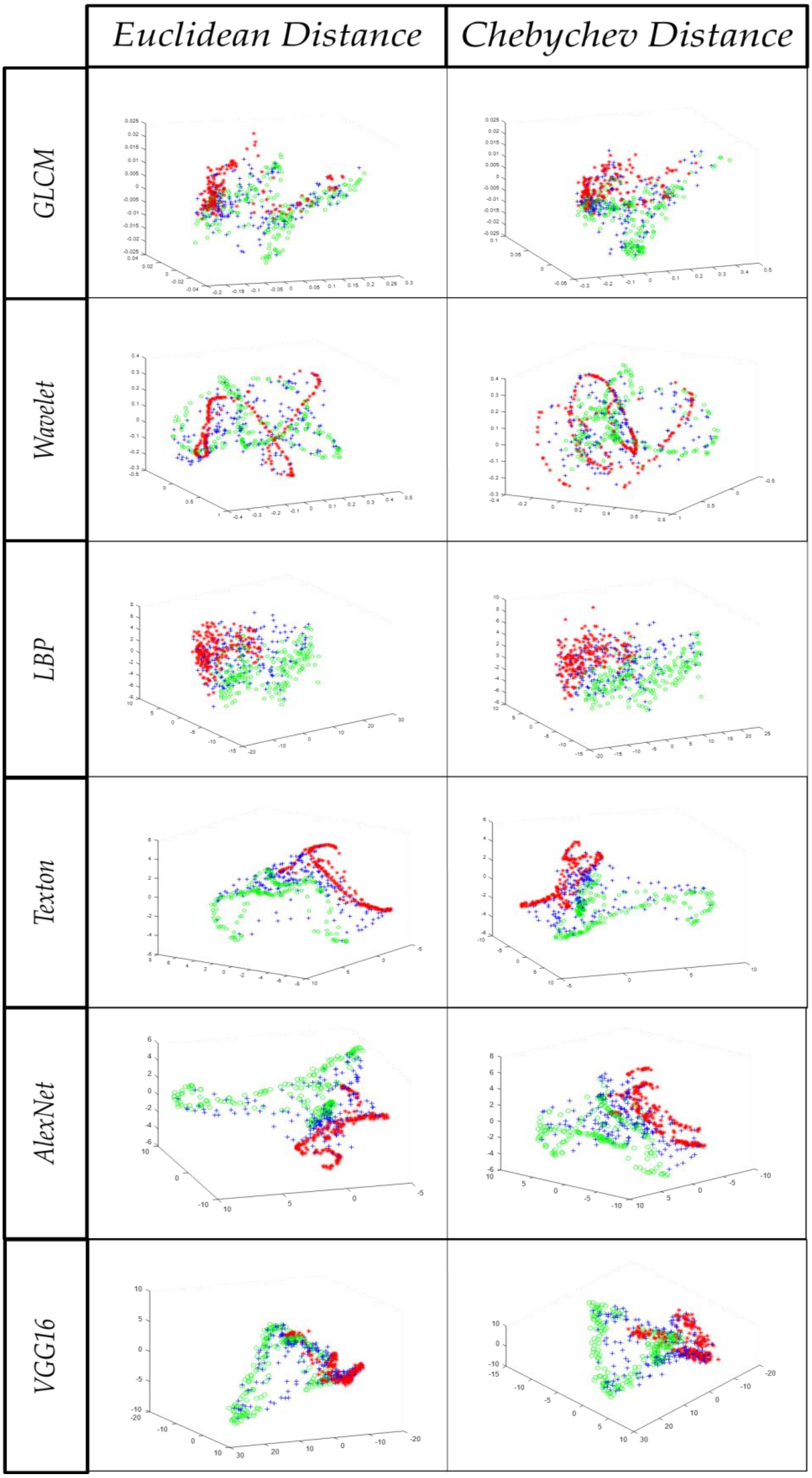


Figure 4-7. Visualisation of the RTA feature sets using the Euclidean distance (left) and the Chebychev distance (right) as projected into 3-D subspace using the first 3

principal component scores. Legends are : red (*), blue (+) and green (o) correspond to NOC, state of transition, and new state, respectively.

While the AlexNet, VGG16, texton and LBP features achieved reliable results in clustering the states, the wavelet feature set provided average result while the GLCM features gave poor clustering performance. In other words, no clusters and distinct separation of clusters are perceived using GLCM feature sets. These are also seen in their classification performance in terms of accuracy percentages. Regardless of the distance metric used, the power of the feature sets to classify the data into respective classes is quite noticeable. GLCM seemed to give consistent limited classification. The RTA-wavelet features, on the other hand, could provide fair classification. Note that the RTA-LBP could also deliver better classification performance. In contrast, it is consistent that the RTA-texton, AlexNet and VGG16 features outperformed for all scenarios. Both clustering and classification of the data were outstanding using these three feature sets.

In general, both Euclidean and Chebychev distance metrics provided good classification performance. Although, the Euclidean-based RTA generally obtained higher slightly classification accuracy than Chebychev-based RTA, the variance, however, is statistically marginal and the results of the latter can also be regarded as equivalent to the former.

Comparison with RQA

The RQA features are also extracted using the RPs constructed from distance matrices. Three threshold values are considered here: 0.1, 0.2, and 0.5. Figure 4-8 presents the RPs having these values. It is apparent that as the threshold value increases, the coarse texture of RPs also increases.

The plots of RQA features in principal component subspace is presented in Figure 4-9. Overall, the clusters of the state systems are obvious in both RQA using ε of 0.1 and 0.2. On the contrary, the ε=0.5 gave poor clusters and no distinct separation are observed in the plots. Furthermore, regardless of ε values, it is also observed that the clusters are equivalent for RQA features that used Euclidean and Chebychev distances. This observation has seen clearly in the classification of the states with only marginal differences are noted in the classification accuracy for all ε values. It is also interesting to point out that the ε=0.5 obtained the poorest classification performance among the three considered ε. This inference could give an idea that the higher values of ε would give noise to the RPs, thus providing poor RQA features and may cause misinterpretation on the dynamic behaviour of the system.

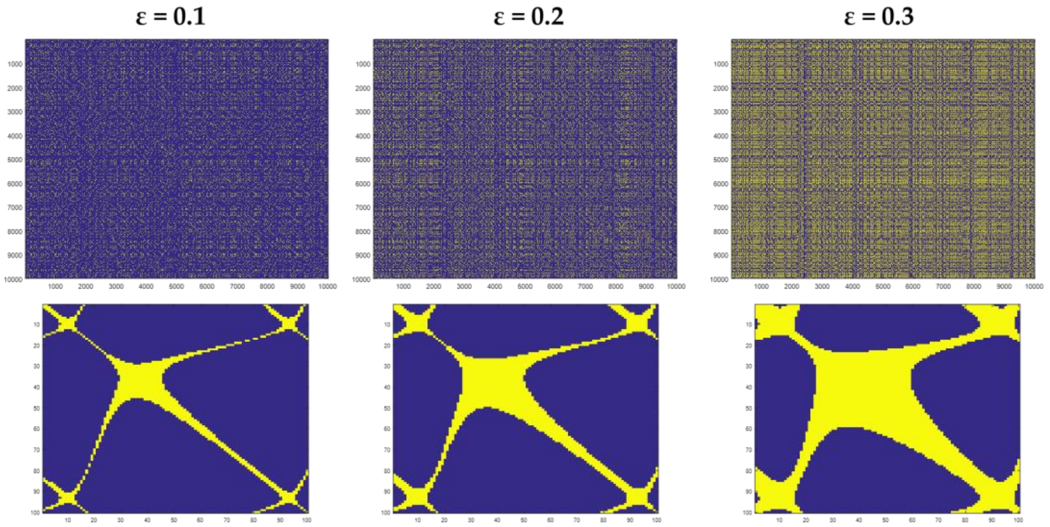


Figure 4-8. Recurrence plots having different ε: ε=0.1 (left), ε=0.2 (middle), ε=0.5 (right), using Euclidean norm. Top and bottom plots correspond to *10,000-by-10,000* and *100-by-100* matrix recurrence plots.

Comparing the results of RQA to the proposed method, it can be inferred that regardless of the distance metric, the results of RTA features gave at par classification performance for wavelet and LBP features, and better performance for texton, AlexNet and VGG16.

To supplement the analysis, the texton are mapped into nonlinear PCA to visualise how the features behaved in a nonlinear principal component analysis (NLPCA) subspace. Using the circular inverse of NLPCA based on autoassociative neural network (AANN), the separation of the clusters using texton features projected into NLPCA subspace is very discernible, as presented in Figure 4-10. Although some of the transition state data points are still seen in the NOC clusters, the isolation between the NOC and new state clusters are clearly seen in the plot.

Overall, some observations can be drawn from this study including outperformance of RTA, particularly RTA-texton, AlexNet and VGG16 over RQA in giving clusters and classification of the state systems, and the poor performance of RQA when higher values of ε (e.g. ε=0.5) are used in constructing RPs. Most importantly, it can also be inferred that the type of distance metric in calculating the distance matrix is insignificant in the general performance of the method and RQA.

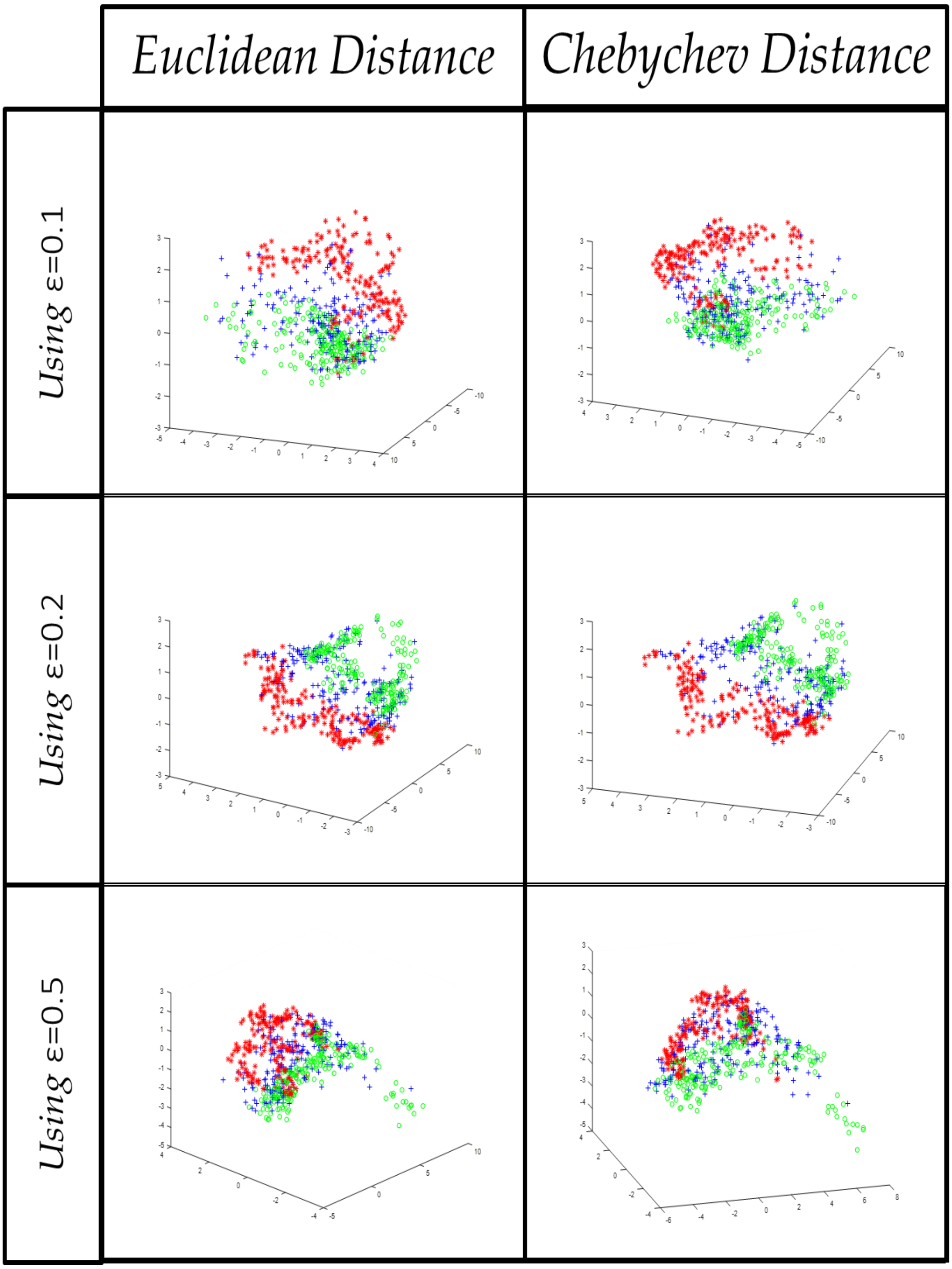


Figure 4-9. Visualisation of the RQA feature sets with different ε values using the Euclidean distance (left) and the Chebychev distance (right) as projected into 3-D subspace using the first 3 principal component scores. The time series is segmented using *b*=100 and *m*=50. Legends are: red (*), blue (+) and green (o) correspond to NOC, state of transition, and new state, respectively.

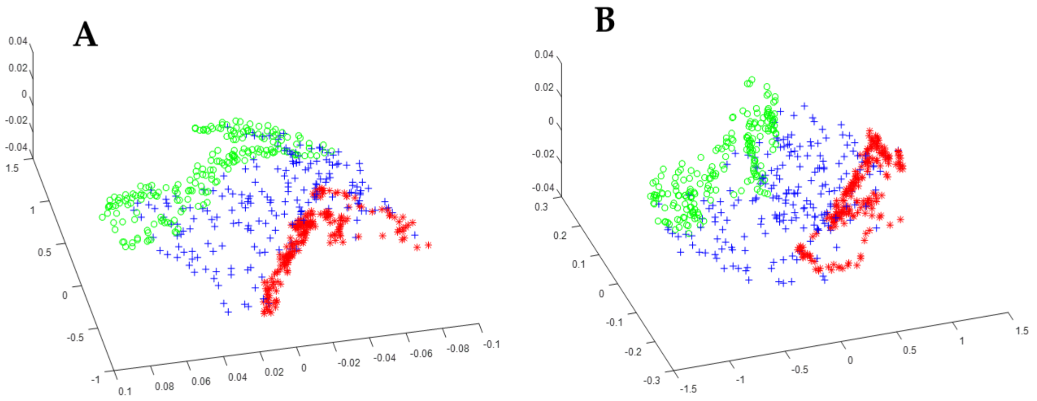


Figure 4-10. Texton features as projected into 3-D nonlinear principal component subspace using (A) Euclidean distance, and (B) Chebychev distance. Legends are : red (*), blue (+) and green (o) correspond to NOC, state of transition, and new state, respectively.

4.6 Application of the method to recurrence plots

For further evaluation, the proposed method is also applied to the recurrence plots and the results are compared to RQA features. This is facilitated by constructing a number of recurrence plots with ε = 0.2. RTA and RQA features are then derived using these plots.

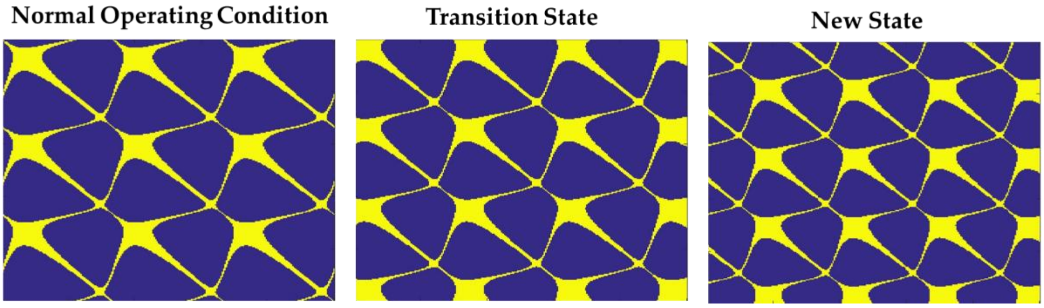


Figure 4-11. RP plots of three state systems of Lotka-Volterra predator prey system. Constructed using Euclidean norm, ε = 0.2.

Clearly, the RPs of the three state systems exhibit periodic typology and the textures of the plots are different from one another. The RPs of NOC tend to have smoother texture whereas the RPs of new state have coarser texture appearance. The recurrence are quite prevalent in the new state. The texture of the RPs of the transition state, on the other hand, is seen intermediary.

As presented in Figure 4-12, distinct clusters can be drawn from all feature sets with clearer agglomeration of data points seen in texton, VGG16 and RQA features. The separation between clusters were also well-defined to these features. The LBP and Alexnet features also provided reliable clusters of the state systems, although some of the features of the transition states were quite spread out across the other clusters.

Table 4-5. Classification performance of RTA features applied to recurrence plots

Feature Sets (using ε = 0.2)		Classification Accuracy (%)
		Euclidean Distance
RTA	GLCM	72.6%
	Wavelet	78.2%
	LBP	73.7%
	Texton	82.6%
	AlexNet	80.1%
	VGG16	87.1%
RQA		82.7%

Evidently, these observations were also seen in the classification of the clusters. Emphasis could be drawn to VGG16 that achieved outstanding accuracy of 87% compared to RQA and other feature sets of only 82%. Furthermore, it can also be said that the RTA-texton is at par to RQA.

In general, it is demonstrated that most of the RTA features are at par to the RQA in quantifying the RPs. Highlight could be given to VGG16, AlexNet and texton which are consistent in providing reliable results in terms of classification and clustering. In other words, this study clearly showed that RTA is a versatile method that could quantify the recurrence in the system regardless the nature of recurrence plots, e.g. thresholded or unthresholded.

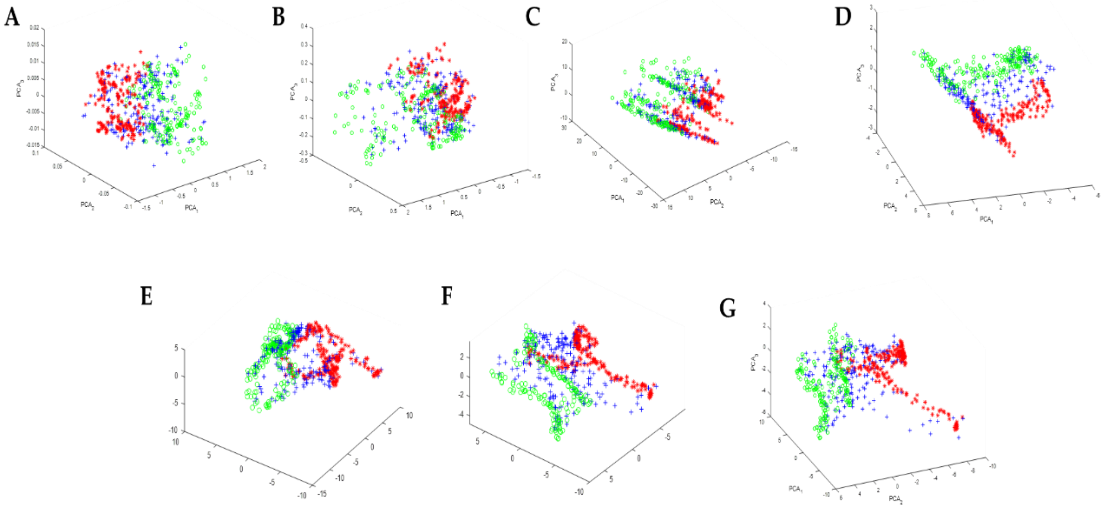


Figure 4-12. 3-D plots of different feature sets obtained from recurrence plots: GLCM (A), wavelet (B), LBP (C), textons (D), Alexnet (E), VGG16 (F) and RQA (G). The plots are constructed in 3-D using the first three principal component scores. Legends are : red (*), blue (+) and green (o) correspond to NOC, state of transition, and new state, respectively.

4.7 Conclusions

In this chapter, the influence of the windowing parameters and the distance metrics in the performance of the RTA in terms of data cluster analysis and classification are studied in detail. The method is also applied to the thresholded recurrence plots to evaluate its versatility in quantifying the recurrence contained in the plots. With this preliminary study, the following observations were discerned:

1. The window width b is a critical parameter that needs to be selected sensibly. Longer choice of window width b could possibly overlook the structural changes of time series, while shorter b could generate false indication of change due to outlier. From this case study, it was observed that longer window width resulted to better clustering and classification of the data. Thus, it can be said that the choice of b is application dependent. Longer window width could be selected when

investigating time series analysis such as classification. On the other hand, shorter window width is favoured in change detection problems.. Moreover, the autocorrelation function could be used as a guide in estimating this parameter. The acceptable rule is that the minimum value of lagged length where ACF is at the minimum should be used as the minimum value of *b* in segmentation of the time series.

2. The sliding step *m* or the windowing method does not significantly influence the performance of RTA, other than providing greater number of segments when smaller *m* is used. The windowing method does not considerably contribute to the overall performance of RTA. The use of either fixed or moving windowing approach is on users' discretion which highly depends on the nature of time series being studied and on the computational capability. Relatively shorter time series may entail moving window to generate more observations for analysis. On the other hand, relatively longer time series data could be processed using fixed windowing as long as the number of segments are enough for subsequent analysis.
3. The type of distance metrics does not significantly affect the performance of the method. In other words, no strong correlation was found between the capability of RTA to cluster and classify the data and the type of distance metrics used in constructing the distance matrix. This observation is also seen when RQA features were considered.
4. The application of RTA in quantifying the texture of unthresholded recurrence plots is found solidly effective. In the clustering of data and eventual classification, it was inferred that RTA features, particularly the texton features, achieved comparative performance to RQA features.
5. In the application of RTA to recurrence plots, it was found out that VGG16 features achieved superior performance against RQA and other

RTA features. Moreover, both the classification results using texton and AlexNet features are comparable to RQA features.

6. Overall, VGG16, AlexNet and texton provided remarkable performance in this case study, owing to its high classification accuracy and good clusters of the data. It outperformed other RTA feature sets and even with RQA features. Moreover, the performance using the GLCM, wavelet and LBP features were seen at par to the RQA features.

This preliminary study provided an insight on how the method is performed in terms of capturing the structure of the dynamics of data through forming the clusters of similar structures.

5. APPLICATION: TIME SERIES CLASSIFICATION

5.1 Introduction

Time series data exist in many scientific fields, including medicine, biology, economy and signal processing, which makes this a good theme of study to most data mining-related research. In fact, in the data mining stance, several tasks have already been explored using the time series that have provided relevant explanation on how certain processes behave and change through time (Esling & Agon, 2012; T.-c. Fu, 2011). These tasks are clustering, classification, segmentation, prediction and fault or anomaly detection, among others.

Time series clustering is a process of finding clusters or natural groups in a data set. It is looking for the groups with maximum and minimum inter-cluster and intra-cluster variances, respectively using a suitable clustering algorithm. Unlike the conventional clustering for static data, time series clustering requires an additional step on data pre-processing via conversion of time series data into some variations of static data. This task does not necessarily require advanced knowledge of classes in the dataset. In contrast, time series classification entails prior information of classes in the training stage. In time series classification, a class label is assigned to an unknown time series. At first, the data with a known label or class are trained using a classifier. Once taught, this trained classifier is then used to identify the class of the new data. The performance of the classifier is typically measured by its classification accuracy.

Segmentation or summarisation of time series is also a common data mining task which involves generating of approximation of time series. The approximation is possible by reducing its dimensionality without losing its important features. This task is usually performed in smoothing the "noisy" time series. The prediction task is regarded as one of the most extensively studied and applied data mining tasks. Basically, this kind of task uses a model in forecasting new values based on the previously observed time series. Lastly, anomaly or fault detection is a task where the main goal is to seek abnormal or fault subsequence in a time series. In essence, this task could give information and pinpoint the irregularities of the series, which qualifies to be applied in process monitoring, fault detection and alarm system.

In this chapter, the proposed method is explored as a means to perform one of the time series data mining task which is time series classification. In this task, both the UCR Time Series Dataset Archive and simulated autocatalytic reaction system are used as time series datasets. Moreover, the optimisation of the four RTA texture algorithms is also carried out in this task.

5.2 Data Description

5.2.1 UCR Benchmark Datasets

The UCR Time Series Archive is a database of time series that are used in many time series analysis studies, particularly in time series classification. In other words, this is an archive of several time series (both simulated and real) data sets from different data creators, which were collated through the effort of Y. Chen et al. (2015). The time series data sets come in different lengths and number of classes and are grouped into training and testing sets.

In this study, only the time series that represent the dynamic process system are considered. These are ECG200, ECG5000 and Chlorine concentration datasets. In addition, the yoga datasets is also included to evaluate the RTA to non-process time series. More information of the datasets are presented in Table 5-1.

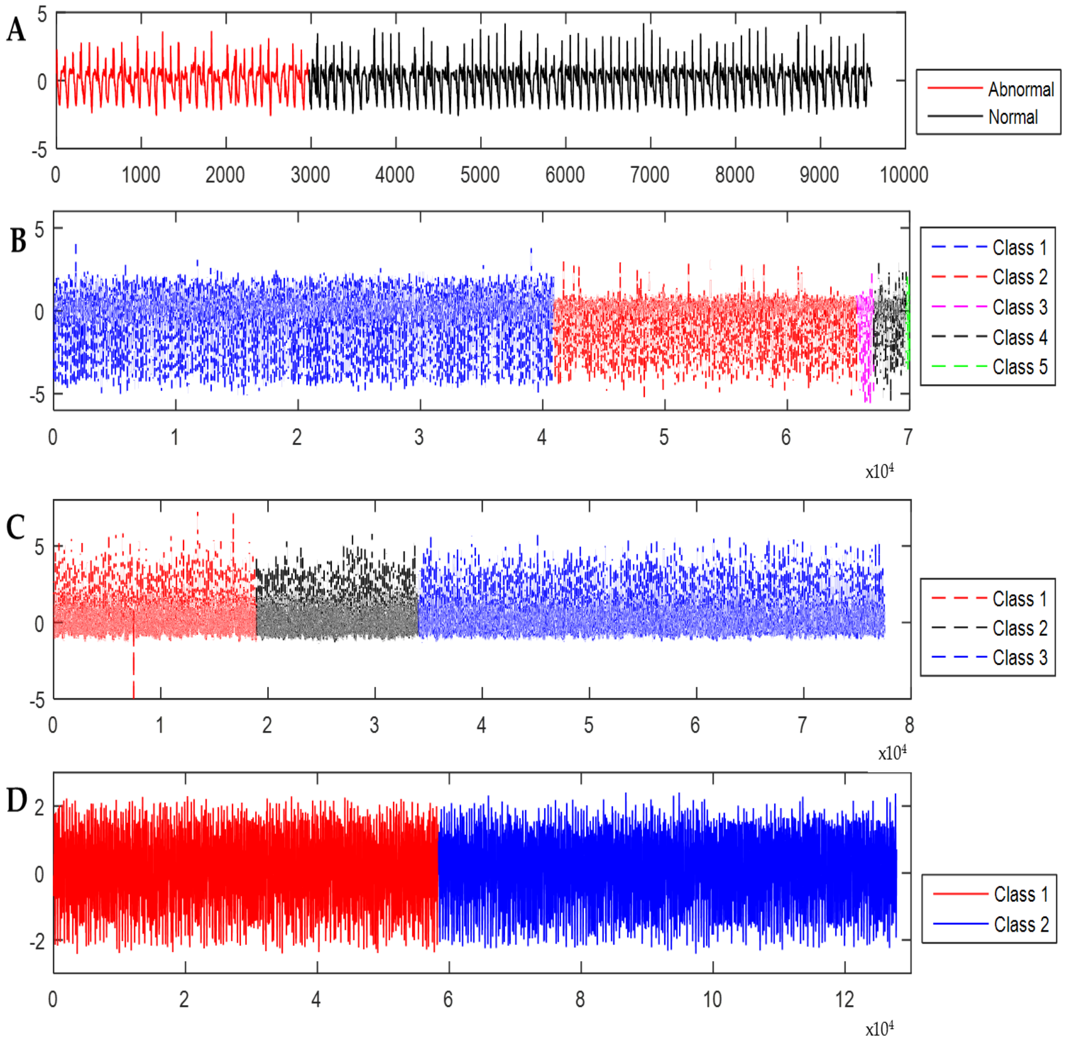


Figure 5-1. Plots of the training datasets of (A) ECG200, (B) ECG5000, (C) Chlorine Concentration, and (D) Yoga time series. The plots are also colour-coded that correspond to the classes.

Table 5-1. Summary of the four UCR time series datasets used in this case study

Datasets	**No. of Classes**	**window width, *b***	**Size of Datasets (No. of segments, N)**	
			Training	**Testing**
ECG200	2	96	100	100
ECG5000	5	140	500	4500
Chlorine Concentration	3	166	467	3840
Yoga	2	426	300	3000

5.2.2 Autocatalytic Reaction System

The simulated time series data of an autocatalytic reaction system are based from an autocatalytic process of two parallel, isothermal autocatalytic reactions in a continuous stirred tank reactor allowing three reactions to occur at different rates (Lee & Chang, 1996). The autocatalytic process is represented by three differential equations as stated below:

$$\frac{dx}{dt} = 1 - x - axz^2 \tag{5-1}$$

$$\frac{dy}{dt} = 1 - y - byz^2 \tag{5-2}$$

$$\frac{dz}{dt} = 1 - (1 + c)z + \gamma_1 axz^2 - \gamma_2 byz^2 \tag{5-3}$$

where x, y, and z are feed concentration variables of the three involved chemical species, and a, b, and c are the scaled time ratios of the reactions. Moreover, the γ_1 and γ_2 represent the scaled feed concentration ratios of the involved chemical species.

In this process, the values of a, b, and c were kept constant throughout the reaction. Only the values of γ_1 and γ_2 were varied. Moreover, only the value of x per time was determined to obtain a univariate time-series which facilitates simple analysis. The values of a, b, and c were fixed to 18,000, 400, and 80, respectively. By modifying only the values of the feed ratio parameters, three states of the system were created. At first, the feed ratio parameters γ_1 and γ_2 were first set to 1.50 and 4.20, respectively, for this first 100 time units. This system is labelled as the normal operating condition (NOC). From there, a linear change on the values of the feed ratios was introduced until 150 time units. The γ_1 and γ_2 were linearly increased to 1.55 and 4.25, respectively. This is the first change of the condition in the process.

After this, another change was introduced to the system by keeping the said values of the feed ratios constant until 200 time units (2nd change).

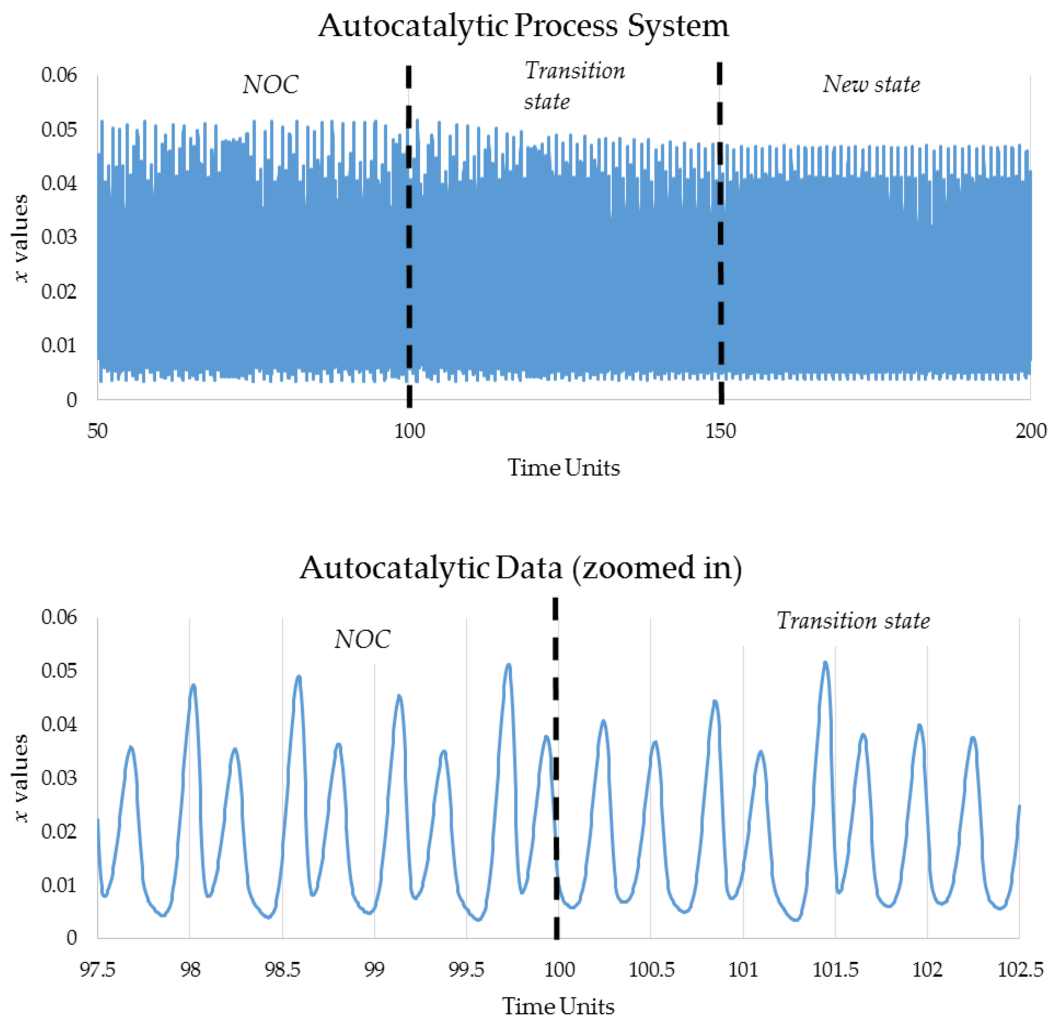


Figure 5-2. The Autocatalytic process time-series data

The simulated time-series data as shown in Figure 5-2 were modelled with equal time step of 0.005 time units, thus, creating a total of 40,000 equidistant values of x. However, the first 10,000 samples (correspond to the first 50 time units) were disregarded from the analysis to ensure that the NOC samples to be used in the analysis are in steady-state condition. Thus, a total of 30,000 samples were used in the analysis with 10,000 samples each for the 3 states.

5.3 Time series Classification

The classification of time series is regarded as the most common data mining task (Ge & Ge, 2016; Silva et al., 2013). In fact, this task has achieved considerable success to several fields, including finance (T.-c. Fu, Law, Chan, Chung, & Ng, 2006), medical science (Chaovalitwongse, Pottenger, Wang, Fan, & Iasemidis, 2011; Jovic & Jovic, 2017), and agriculture (Tatsumi, Yamashiki, Canales Torres, & Taipe, 2015), among others. In time series classification, a class label is assigned to an unknown time series. At first, the data with a known label or class are first trained using a classifier. Once taught, this trained classifier is then used to identify the class of the new data. The performance of the classifier is typically measured by its classification accuracy. However, unlike the traditional classification problems, time series classification requires some representation of data. It is in this regard that the recurrence texture analysis is considered.

5.3.1 Related Works

Based on literature, the use of textural features of distance matrices as predictors in time series classification is a novel theme in data mining world. Limited studies have used texture analysis alone in dealing with the problem. Moreover, these studies only addressed specific problems. In the work of Yanhua et al. (2006), the co-occurrence based temporal textures are extracted from time series flourescence microscope images and used as predictors in the classification of subcellular location patterns. They regarded the co-occurrence based temporal textures as robust features as these give both temporal and spatial information which become the basis of attaining high classification accuracy. Similar work was done by Singha et al. (2017) wherein they used a combination of temporal features extracted from coarse resolution time series data and spectral features of fine resolution data for object-based paddy rice mapping application. The temporal features are extracted on the Moderate resolution imaging spectroradiometer (MODIS) of the remote sensing of paddy rice.

The closest work to the method is method proposed by Souza et al. (2014) wherein they used textural algorithms to extract features in the recurrence plots. In this work, they used the combination of all extracted features from Local Binary Pattern (LBP), Grey-Level Co-occurrence Matrix (GLCM), Gabor filters, and Segmentation-based Fractal Texture Analysis (SFTA) as predictors in classification of the UCR Time Series Archive. They referred the method as " Texture Features from Recurrence Patterns (TFRP)". Moreover, the SVM was used a classifier in their study and compared to other methods that also employ recurrence plots in dealing the classification problem. One of the which is the work presented by Silva et al. (2013) which they named as "Recurrence Patterns Compression Distance (RPCD)". RPCD uses recurrence plots to represent the time series data, and 1-NN algorithm to estimate the similarity of the two recurrence plots via employing a video compression based distance measure (CK-1). Moreover, RPCD compares the texture similarity between two images using Kolmogorov complexity.

5.3.2 Methodology

Textural Extraction Algorithms

In this task, the Recurrence Texture Analysis was used as a representation of time series to obtain predictors for classification. In essence, the time series are represented by a set of distance matrices and its textural features were consequently extracted. The texture features are the treated as predictors for classification.

The four texture extraction algorithms, namely GLCM, wavelet, local binary pattern and texton were employed in this study. The features are extracted using a combination of hyperparameters of the techniques with varying level of values, as tabulated in Table 5-2. It was emphasised that some of its parameters were optimised in this study.

Table 5-2. Hyperparameters of the texture extraction algorithms used in the study

Algorithm	**Hyperparameter**	**Values used**	**Remarks**
GLCM	Number of grey levels	8, 32, 64	For optimisation
	Distance	1	
	Directions (degrees)	[0, 45, 90, 135]	
Wavelet	Type of wavelet	Daubechies (db4)	
	Level of wavelet decomposition	2, 3, 4	For optimisation
Local Binary Pattern	Radius of circular local pixel neighbourhood (R)	0.5, 1, 2	For optimisation
	No. of sampling points in the pixel neighbourhood	8	
	Map Type	No mapping	
Texton	Filter Name	Schmid, Leung-Malik, Root Filter Set	For optimisation
	Filter Size	49	
	No. of cluster centers	40	
AlexNet	No. of Convolutional Layer	5	
	No. of Pooling Layer	3	
	No. of Fully Connected Layer	3	
	Image Resolution (pixel)	[256 x 256]	

VGG16	No. of Convolutional Layer	13	
	No. of Pooling Layer	5	
	No. of Fully Connected Layer	3	
	Image Resolution (pixel)	[224 x 224]	

Building of Classification Model

For the classification part, building of the classification model was done using Support Vector Machine which was previously discussed in Chapter 4. This classifier was trained using the training data set with known class label. Once trained, this was then used to classify another set of data (test data set) and the classification accuracy was computed on this matter.

It should be noted that during the training of the models, the parameters of the classifiers should be optimised so that misclassification errors and penalty terms are minimised. Moreover, validation technique should also be carried out to validate the result and to minimize, if not eliminate, the overfitting of data. In this regard, cross-validation is normally used. It works by selecting the number of cross-validation folds. The model for each fold are trained using all the data outside the fold. Furthermore, the model is tested using the data inside the fold and subsequently calculated the average test error over all folds, and a good estimate of the classification accuracy is achieved. In this case, the study employed five-fold cross validation. The trained models are then used to classify the test datasets. The classes were predicted and the classification errors were obtained.

5.4 Results and Discussion

5.4.1 Time series Analysis on UCR Time Series Data

Four datasets were used to exhibit the robustness of RTA to deal with time series analysis. RTA features were extracted from their distance matrices that were subsequently used as predictors for SVM classification. Representative samples of distance matrices for each classes are shown in Figure 5-3 . Prior to the classification, the RTA features were optimised by optimising the hyperparameters of the four textural extraction algorithms.

The optimisation of the hyperparameters of these algorithms, as shown in Table 5-2, were initially carried out using the training datasets. Moreover, the SVM classifier was also optimised by adjusting the box constraint level so that the optimised classification accuracy is attained.

Once the parameters were identified, the classification of test data was conducted. The accuracy rates were determined for each texture algorithm and were used to compare their performances. Table 5-3 shows the summary of the classification accuracy percentages in both training and testing stages.

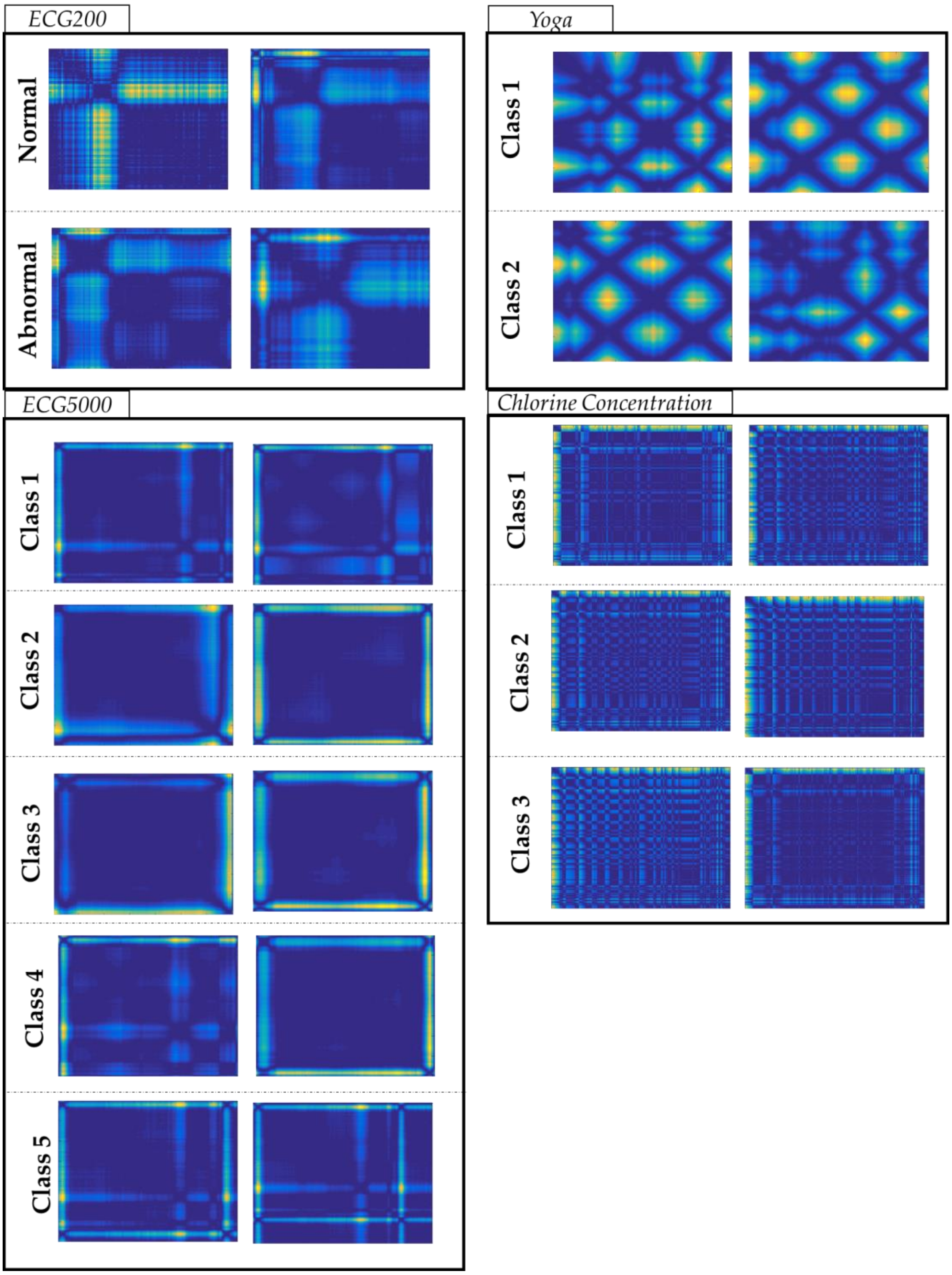


Figure 5-3. Representative samples of distance matrix plots of the training datasets of four UCR Time Series Archive used in the study

Table 5-3. Results of the classification performance of RTA during the training and testing stages

		ECG200	*ECG5000*	*Chlorine Conc*	*Yoga*
Training	GLCM	78.0%	90.4%	58.5%	64.3%
	Wavelet	79.0%	91.2%	58.0%	66.0%
	LBP	79.0%	89.6%	59.3%	63.0%
	Texton	85.0%	94.0%	58.2%	67.7%
	CNN-AlexNet	86.0%	94.6%	65.7%	79.0%
	CNN-VGG16	**87.3%**	**95.1%**	**66.2%**	**83.1%**
Testing	GLCM	77.0%	90.0%	56.3%	62.9%
	Wavelet	74.0%	91.7%	55.1%	65.6%
	LBP	74.0%	91.0%	56.1%	68.2%
	Texton	84.0%	94.1%	55.7%	60.9%
	CNN-AlexNet	90.0%	94.2%	71.1%	86.9%
	CNN-VGG16	**93.2%**	**95.3%**	**75.2%**	**89.5%**

The accuracy rates obtained during the testing stage was used to evaluate the performances of each texture extraction algorithm. Based on the table above, in general, the VGG16 algorithm provided an outstanding result in the classification. It obtained the highest classification accuracy for all the considered datasets. Moreover, the AlexNet and texton features also provided average results.

To present the reliability of the approach further, the results of the RTA were also compared to the other published methods which also utilized the same sets of time series data. One of which was the method proposed by Souza et al. (2014), referred as TFRP, which employed the texture features of recurrence plots. In their paper, they have considered two sets of features, i.e. the combination of all texture features and the reduced features amounting to 20%. Furthermore, the work presented by Silva et al. (2013), referred as RPCD, was also considered in the comparative evaluation. Even though the method did not involve extracting texture feature, the method is still valid for comparison since they used similar time series data sets in their classification.

Table 5-4. Comparison of the error rates of the proposed method (RTA) to other approaches (TFRP and RPCD)

Data Sets	**RTA**						**TFRP**		**RPCD**
	GLCM	**Wavelet**	**LBP**	**Texton**	**AlexNet**	**VGG16**	**All**	**20%**	
ECG 200	0.23	0.26	0.26	0.16	0.1	**0.07**	0.17	0.17	0.14
ECG 5000	0.10	0.08	0.09	0.06	0.06	**0.05**	No Data	No Data	No Data
Cl_2 Conc.	0.44	0.45	0.44	0.44	0.29	**0.25**	0.30	0.29	0.49
Yoga	0.37	0.34	0.32	0.39	0.13	**0.11**	0.14	0.13	0.13

Based on the table, TFRP presented a considerable degree of success in the classification of the considered UCR datasets as shown by its low error rates for most of the datasets. Meanwhile, it was seen that the RTA-texton is at par to this method. More importantly, the better performance of both CNN features (AlexNet and VGG16) are clearly demonstrated in this undertaking. It can be argued that these state of the art methods reliably classify these datasets which are found superior even to the published methods. It can also be observed here that the VGG16 was found significantly better than AlexNet.

This is an interesting inference which is consistent with the premise that better classification could be obtained when neural networks are becoming deeper (Mehdipour Ghazi, Yanikoglu, & Aptoula, 2017; Simonyan & Zisserman, 2014).

In sum, RTA is a reliable method to deal with time series analysis. In fact, it was demonstrated that the proposed method signifantly discriminated the classes of the UCR time series, and even outperformed some its peer published methods.

5.4.2 Time series analysis on autocatalytic reaction system

In the pre-processing of the data, a window width of *b*=25, is used to partition the time series with step size of 10, which resulted to 1191 segments. Each segment is processed through calculating the distance matrix and subsequently extracting the RTA features. Samples of distance matrix plots are shown in Figure 5-4. By looking the forms of the distance matrices, it can be said that it is difficult to distinguish the differences between classes. The prominent cyclic features and textures of the plots are very much alike to one another.

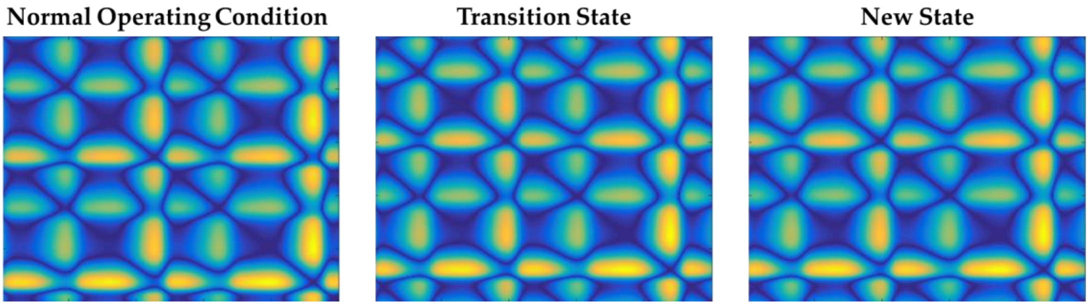


Figure 5-4. Exemplary distance matrix plots of the autocatalytic process system showing NOC (left), transition (middle) and new state system (right)

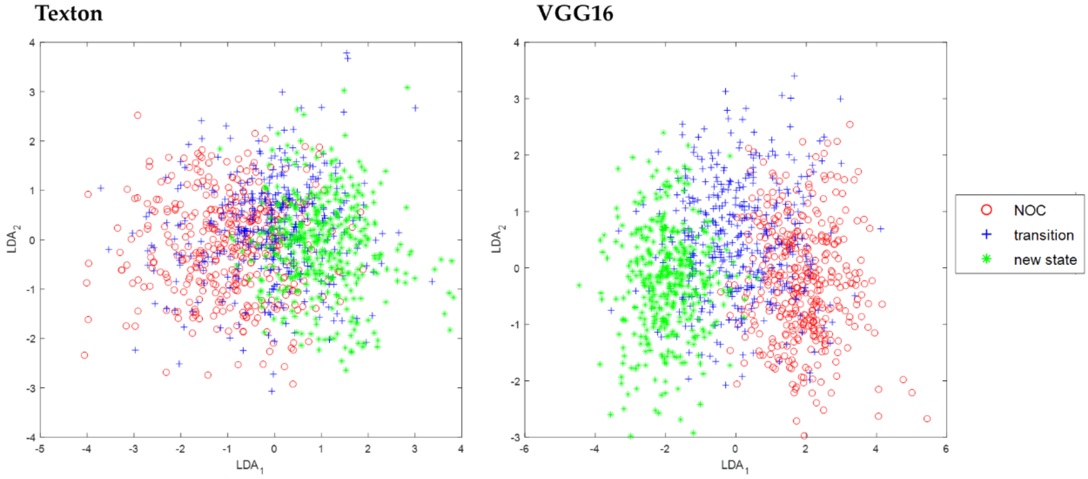
Optimisation of the hyperparamaters of textural extraction algorithms was first performed to obtain augmented RTA features to be used in classification using SVM classifier. During the classification, the model are trained using the training datasets which composed of 70% of the total datasets and was subsequently validated using five-fold cross validation method. The remaining 30% of the datasets was then used to test the model and the classification accuracy for each RTA algorithms were reported. The results were in summarised in Table 6-5. Based from the table, most of the RTA features got nearly poor classification on the states of the system, which concurred to the difficulty in visual detection in the differences of distance matrices of each state. The poor classification may also indicate that a single variable of system would not be sufficient for classification problem. In other words, more variables (i.e., y and z) could be incorporated to build a more reliable classification model using RTA features.

Table 5-5. Results of the classification accuracy on the test dataset of autocatalytic process

Feature Set	**No. of dimensions**	**Classification Accuracy (%)**
RTA – GLCM	8	42.8 %
RTA – Wavelet	9	39.8%
RTA – LBP	256	54.5%
RTA – Texton	38	67.7%
All RTA features	311	55.6%
CNN – AlexNet	4096	61.7%
CNN – VGG16	4096	**72.3%**
RQA (ε = 0.1)	12	50.1%

Among the RTA features, the VGG16 considerably discerned the classes of the dataset, followed by texton with 72% and 68% accuracy percentages, respectively. Moreover, their classification results, along with AlexNet, are even higher than the combined GLCM, wavelet, LBP and texton features with only 56% accuracy, and to the RQA features which only correctly classified 50% of the datasets.

Figure 5-5 presents the plots of the first 2 linear discriminant scores of the VGG16 and texton features displaying the maximum separation between classes. It is apparent that most of the features of transition state (blue '+') are scattered in NOC and new state. This inference is consistent to the concept of this system since the transition is a pseudo-state where the shifting process from one stable state (NOC) to another one (new) is taking place. It is also worth to mention that the NOC and new states are completely separable to each other in VGG16, which could give an insight that VGG16 could essentially differentiate the change of the process and hence, could be used in predicting the new state from NOC.


Texton

VGG16

Figure 5-5. RTA-Texton (left) and CNN-VGG16 (right) features showing maximum separability between states in linear discriminant subspace

The RQA was also employed in the time series and the results were compared to the RTA feature sets to facilitate comparative evaluation of the two similar approaches. Using the same windowing parameters, 12 RQA features were extracted from recurrence plots of the time series using different threshold value ε, ranging from 0.01 to 0.5, which was optimised prior classification. The value of ε = 0.1 was then selected in this regard. Using these

RQA features as predictors for SVM classifier, the classification of the states of the system resulted to only 50% accuracy, which is only at par to the rest of the RTA features. More interestingly, the RQA features did not fairly classify the states as compared to two CNN features and to RTA-texton. In this case, it can be then said that the RTA is a more robust feature set that can be used in the change-point detection and in classification task even when employed in a more challenging process system such as in autocatalytic process. In other note, the fact that the GLCM, wavelet and LBP also provided at par and acceptable results to RQA, those feature sets could also be regarded as adequate which could be employed to other time series problems.

In this case study, it was showed that the RTA is a good alternative method that could be used in time series analysis, particularly in time series classification. The RTA feature sets are reliable predictors to SVM classification models to various time series datasets. In fact, RTA performed well compared to other similar approaches.

5.5 Conclusion and Recommendation

In this chapter, the RTA is used in time series classification. In this regard, RTA is used as a pre-processing tool to extract meaningful information of the time series and the features were used for classification. In particular, the method is applied to classify some of the public datasets which are often used in studying time series analysis. Additionally, the simulated autocatalytic process time series is also included in this undertaking to classify the states of the system. More importantly, this chapter included the optimisation of the hyperparameters of the four textural extraction algorithms (GLCM, wavelet, LBP, textons). The classification performance was assessed using SVM and the results were compared to other results of other similar approaches.

Using the public time series dataset, the evaluation of the technique showed that all the four RTA features, AlexNet, and VGG16 generated a satisfactory level of success in terms of classifying the classes of time series considered in this study. Interestingly, the texton, AlexNet and VGG16 delivered outstanding classification performance compared to the other three algorithms. In the comparative evaluation, the RTA could be regarded as a reliable tool in time series classification since the results showed that it was at par to the other similar methods, with VGG16 showed a very competent method in dealing the classification task, even beating other published methods. Moreover, it was also confirmed here that better classification could be obtained from deeper neural network (VGG16). However, while these observations held true to the four considered, further evaluation of other publicly available data sets needs to be pursued to strongly validate this claim. Moreover, these RTA features could also be combined which could be served as combined predictors, with option to reduce the dimension using some appropriate techniques such as PCA.

In the simulated autocatalytic process time series, it was showed that RTA is a reliable method to classify the states of the system, both VGG16 and texton displayed the best classification performance among the other feature sets. In further evaluation, the method was compared to RQA in terms of the power to classify the states. It was showed that VGG16, AlexNet and texton provided a better classification accuracy against RQA features. The GLCM, wavelet, and LBP, on the other hand, found to be at par to RQA feature sets. More importantly, the visualisation of the RTA features into 2-D LDA subspace offered a good insight on how the data is clustered which can be used as an extension study like process monitoring, change-point detection and fault detection. RTA could be used as a base method in constructing monitoring tool for dynamic process system.

Overall, it was showed in this chapter that RTA features, particularly the CNN and texton, is a good alternative method for time series analysis, i.e. time series classification. The RTA feature sets are reliable predictors to SVM classification models to various time series datasets. Lastly, the method performed well in the classification task compared to other similar approaches (i.e. RQA, TFRP, RPCD).

6. APPLICATION: CAPTURING THE DYNAMICS OF SOLIDS PROCESSING DATA

6.1 Overview

Most natural and chemical processes are considered dynamic systems owing to their complex and dynamic characteristics. In particular, processes related to solid processing, i.e. milling and powder flow, usually behave in complex and nonlinear manner, due to several factors including the interaction of the solid particles during the process and the influence of the external elements (e.g. temperature, pressure). The dynamic behaviour of the processes is actually a major challenge in its precise modelling and optimisation of the process.

In this chapter, the recurrence texture analysis is applied to real data sets on solids processing. Three case studies are considered here. The first one is the characterisation of the dynamic behaviour of an autogenous mill using only one mill parameter. The mill load is used in identifying the mill controller states. In the second case study, two mill parameters are then considered: the power draw and outlet temperature of the IsaMill. Initially, this study employed RTA to characterise the behaviour of the IsaMill using these parameters and understand how it interacts with each other. Then, the study is extended in predicting the particle sizes of the feed using the labelled P80 values. In this occasion, the predictive modelling is employed by treating the RTA features as predictors in the model. The RTA-texton is used in both case studies due to its reliability as shown in the preliminary study. In the last study, the powder flow behaviour is characterised using the time series of its masses. In this study, all RTA features are considered in the classification model.

6.2 Identification of autogenous mill controller states using mill load data

The use of autogenous mills has become widespread, owing to their capacity and favourable capital and operating costs. Optimal operation of these systems is critical, given their high energy consumption and affect on downstream processing. A considerable effort has been spent over the years in the development of advanced control systems to achieve a better performance in this regard. However, despite significant advances, the problem still remains challenging, owing to the nonlinear behaviour of the system with respect to the mill variables, and, the unmeasured ore variability come from the mine.

The mill load, in particular, can be seen as a state variable of the mill, which could be important for; modelling and control, and, important insights into the mill behaviour. In this investigation, time series cluster analysis was performed using online mill load measurements in order to identify the different control states of the mill. More specifically, the RTA-texton is applied to the mill load time series data. These texton features are subsequently be used to promptly identify the normal operating conditions from other mill states, viz. feed disturbances and feed limited in the mill.

6.2.1 The Mill Load data

The load data is generated from a fully autogenous closed-loop mill with a recycling load that overflowed from a screen unit. The basic diagram of the mill is presented in Figure 6-1. The time series data are used to distinguish the type of state conditions of the mill. The mill load data contain a total of 12,600 observations from three operational states, namely normal operating condition (NOC), feed disturbance (FD) and feed limited (FL), as defined by an expert mill controller. The NOC state is considered the default state of the mill or when none of the other conditions were flagged by the controller. On the other hand, FD is labelled when the combined ore feed rate is in excess of the average feed rate by a set threshold. However, when the mill load has the load rate of change less than zero while the combined ore feed rate is still in excess to the average rate, this is then labelled as FL.

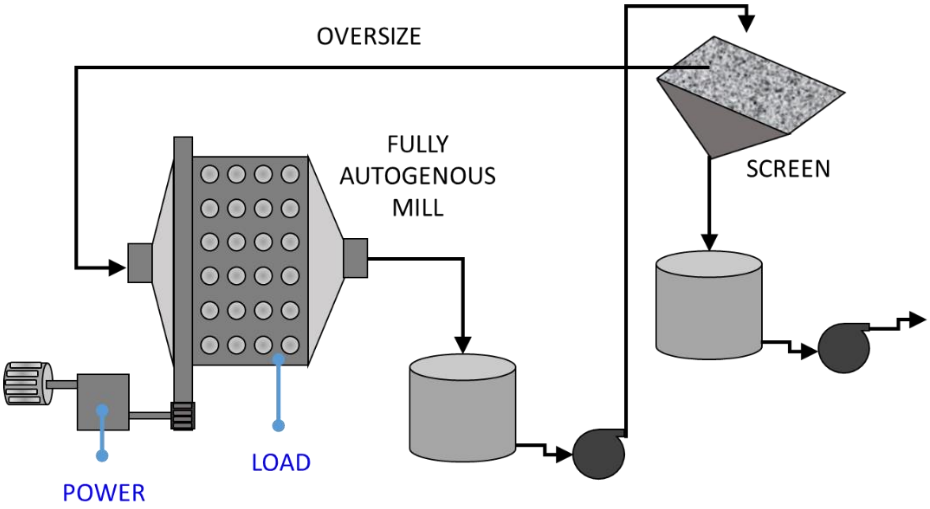


Figure 6-1. Basic diagram of a fully-autogenous mill

The distribution of the number of observations per states as well as the normalised time-series load data were given in Table 6-1 and Figure 6-2, respectively.

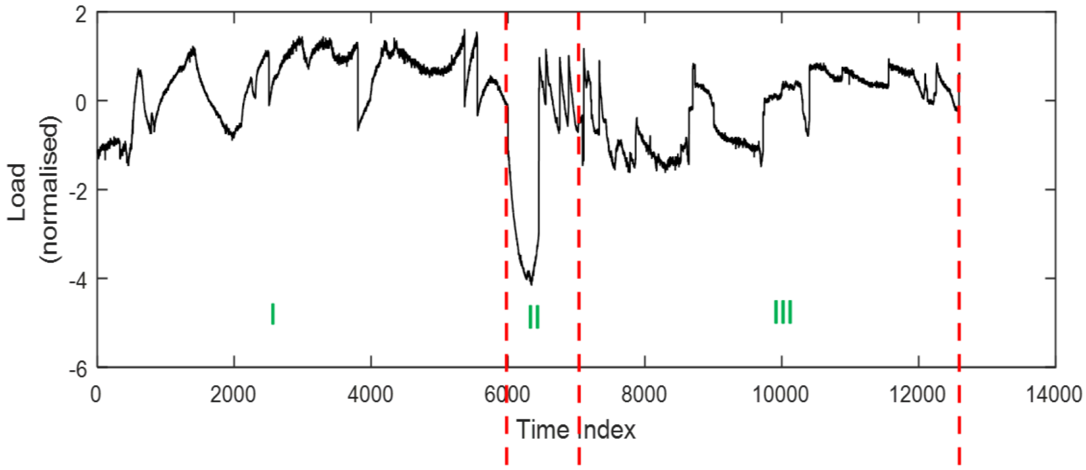


Figure 6-2. The normalised time-series mill load with labelled states: (I) NOC, (II) Feed disturbance, (III) Feed limited

This time series was segmented into *N* segments using fixed windowing, thus each segment contains *b* observations that was estimated using the ACF as shown in Figure 6-3. A total of 50 segments, each has a segment length of 250, are used to calculate the Euclidean distance matrices.

Table 6-1. Mill load time-series data

Mill **states**	**No. of Observations** ***(n)***	**No. of Segments (N)** ***(for b=250)***
Normal Operating Condition	6,000	24
Feed Disturbance	1,500	6
Feed Limited	5,100	20
TOTAL	**12,600**	**50**

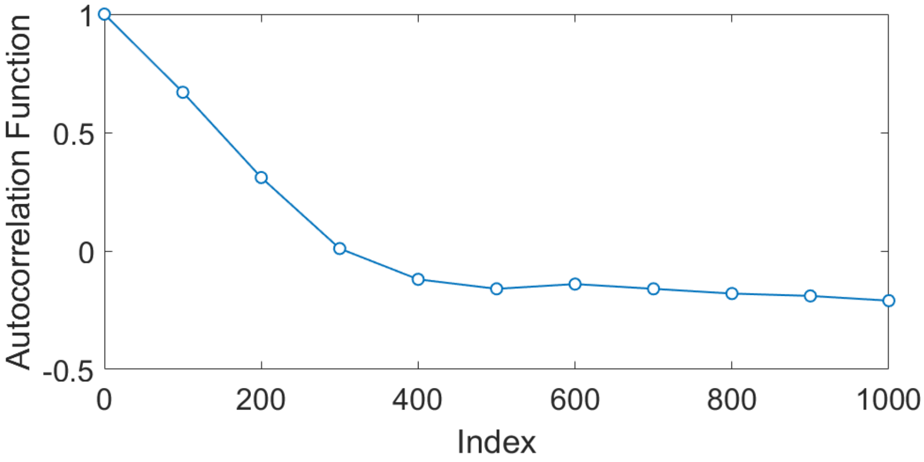


Figure 6-3. The Autocorrelation Function (ACF) of the mill load time series showing b=250.

6.2.2 Results and Discussion

The time-series load data was used to classify the three mill operational states, i.e. normal operating condition, feed disturbance, and feed limited. Euclidean distance was used to calculate the distance matrix of the segmented time series. The representative plots of the distance matrices are displayed in Figure 6-4. It is interesting to note that the visual textural appearance between the distance matrices of NOC and FL are apparently comparable. Moreover, the distance matrix of FD look finer than the other two.

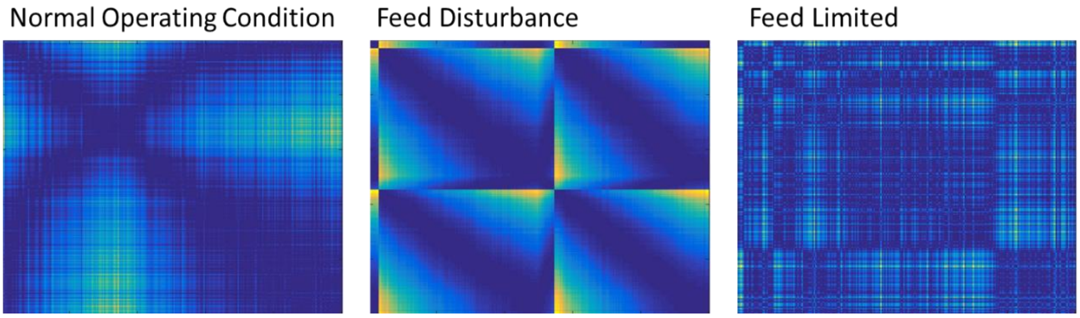


Figure 6-4. Representative plot of the Euclidean distance matrices of three operational states of the autogenous mill

The texton features were extracted from these Euclidean distance matrices, and subsequently visualised by projecting the linear discriminant scores onto 2D space. As presented in Figure 6-5, the NOC state formed a good cluster group and acquired total separability to the FD datapoints. However, there is an observed overlap with the clusters of FL. Conversely, the FL has also generated a quite defined cluster group.

With this promising result, it could then be discerned that the proposed method can potentially characterise the mill states using the mill load parameter only. Although this could only be validated if more data points are used and calibration of the model is carried out. The sliding window can also be considered for in-depth analysis of the data. The fault detection scheme can also be derived using this approach via training the NOC data and treating the other mill states as faults or abnormal datasets. Control limits can be computed in the NOC data and decision boundary can be derived during the training. The use of one-class data description (e.g. SVDD, GMM) is a good consideration for this approach.

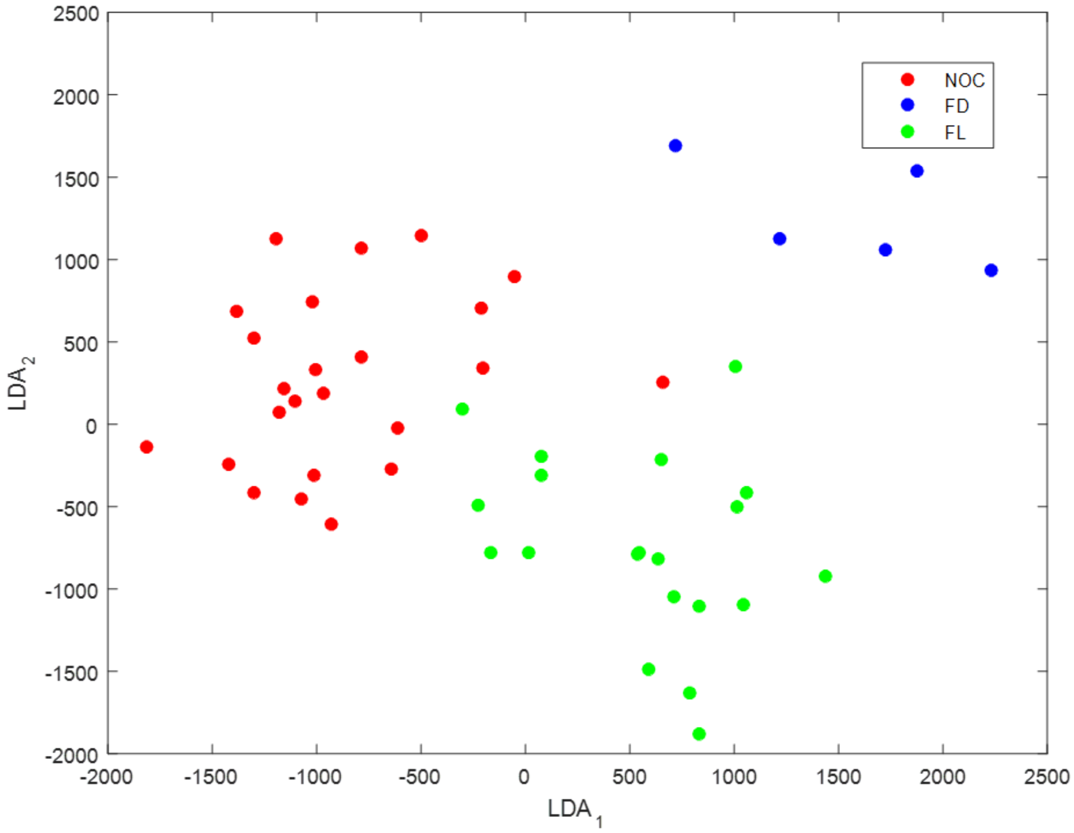


Figure 6-5. Visualisation of the texton-schmid feature sets of mill load data as projected onto 2D space using the linear discriminant scores.

One significant interpretation that could be drawn in this case study is the use of only one operational variable in evaluating the performance of the autogenous mill. The load could alone be used by the controller to monitor the operating condition of the mill. Moreover, the models considered here could be in principle be implemented online once appropriately calibrated. Subsequently, this could form a foundation for more advanced process control models for autogenous mill.

6.3 Estimation of feed particle sizes in a horizontal stirred mill (IsaMill) using power consumption and temperature time series data

Energy consumption in comminution circuits is often the single largest component of the energy expenditure in concentrator circuits, owing to the inefficiency of the processes used to reduce the size of ore particles (Fuerstenau & Abouzeid, 2002), among others. Considerable effort has been expended to increase the efficiency of grinding circuits by improvements in the design of comminution equipment, as well as smarter operation or better control of operations (Aguila-Camacho, Le Roux, Duarte-Mermoud, & Orchard, 2017). The limited availability of reliable models is a significant and ongoing challenge in these efforts, as comminution circuits can exhibit complex nonlinear behavior that can be difficult to capture in first-principles models (Aguila-Camacho et al., 2017).

As a result, the use of data-based process models is increasingly attracting attention, as data are becoming abundant and the means to process these data are likewise becoming increasingly sophisticated and cheap. Over the last decade, a number of studies have appeared in this regard, e.g., in the analysis of vibration and acoustic signals from comminution equipment (Tang, Chai, Yu, & Zhao, 2012; Zeng & Forssberg, 1994; Zeng & Forssberg, 1996), and soft sensor approaches to the monitoring of mill operations (Shao, 2005).

In many of these studies, nonlinear time series analysis, in one form or another, plays a critical role; in this investigation, the recurrence texture analysis is applied to explore the behaviour of grinding equipment. In particular, this section highlights the estimation of the particle sizes of the feed in the stirred horizontal mill (or also known as IsaMill) of an Australian base metal plant using the time series data of both power consumption and outlet temperature. Feature extraction algorithms such as texton, CNN-AlexNet, and CNN-VGG16 are used in this undertaking, and the results are compared to the traditional nonlinear time series method which is the delay vector or lagged trajectory coordinates.

6.3.1 Power draw and outlet temperature of IsaMill

Two time series both having a length of 74,312 observations from power draw (measured in kW) and outlet temperature (in ^{0}C) along with the measured P80 values (in μm) ranging from 13 to 21 μm, as indicated in Figure 6-7, were obtained from a horizontal stirred mill (IsaMill) operating on a base metal plant in Australia. The simplified diagram of this IsaMill is shown in Figure 6-6.

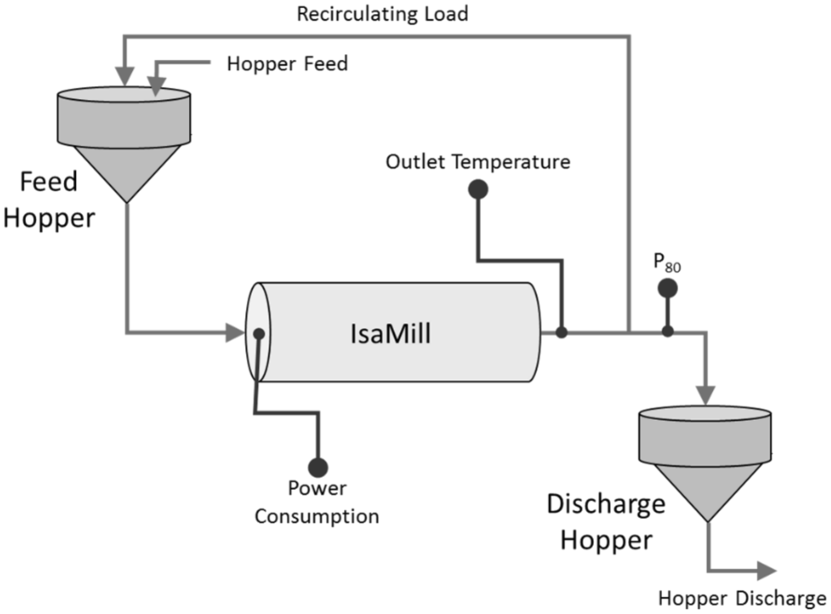


Figure 6-6. Flow diagram of the IsaMill grinding circuit, showing the measuring points of the power, temperature, and P80 particle sizes

The data are collected over a period of approximately two months wherein the product size (P80 values) of the mill was measured off-line once every 24 h and had to be fused with the temperature and power data that were collected every minute. This was done based on a 'hold until updated' approach on the same timeline as the power and temperature samples.

Prior the application of RTA, the raw data are first plotted into 2-D subspace to understand how the two mill parameters are correlated to each other. As shown in Figure 6-8, the data are overlapping to each other and is difficult to discern the pattern.

The P80 values were initially divided into three groups that would represent the classes or the texture of the feed. The division of the P80 values was carried out using a hierarchical clustering approach with ward linkage specifying the wanted number of class. As a result, the time series were partitioned into 3 groups (fine, intermediate, coarse). The distribution of particle sizes are summarized in Table 6-2 and Figure 6-9.

The analysis of temperature and power draw time series were treated separately. As shown in Figure 6-10, both time series has a minimum autocorrelation of 720, thus the window width is pegged on that value. For the sliding step, the author selected a value equal to 100. As a result, these values generated a total of 723 discrete time series (or segments) spread across three classes using a moving window segmentation approach. These segments are used to calculate Euclidean distance matrices. In addition, the cross-distance matrix between these segments are also calculated. This is made to evaluate the effect of these two parameters in the capacity to classify the particle size.

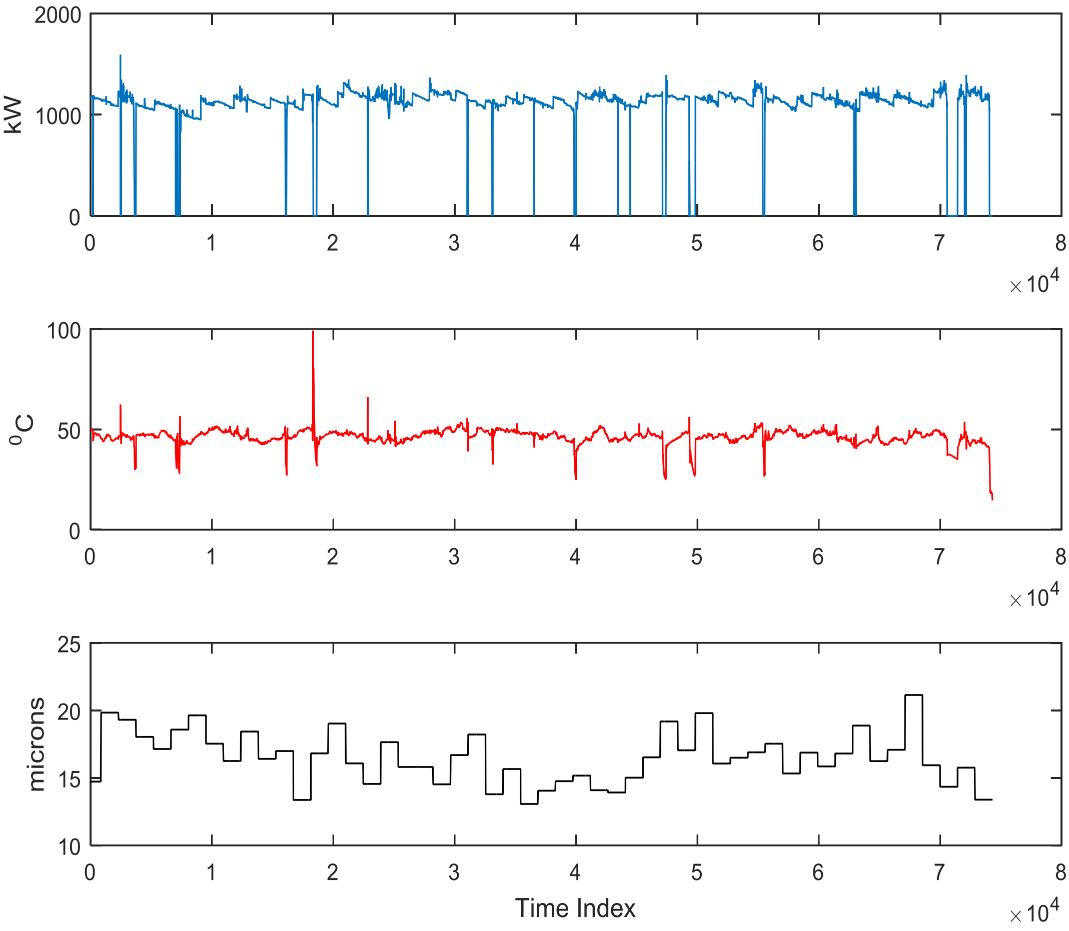


Figure 6-7. The raw time-series of power draw (top) and temperature (middle) of the autogenous mill, and the corresponding particle size measurements (bottom)

In this study, the particle sizes are estimated using kernel Support Vector Machine with the unreduced features as inputs (predictors) and the classes of the mean particle sizes as outputs (response). In the classification, four sets of each of the texton, AlexNet and VGG16 features are considered, i.e. features extracted from power, temperature, and cross distance matrices and the combined features of power and temperature.

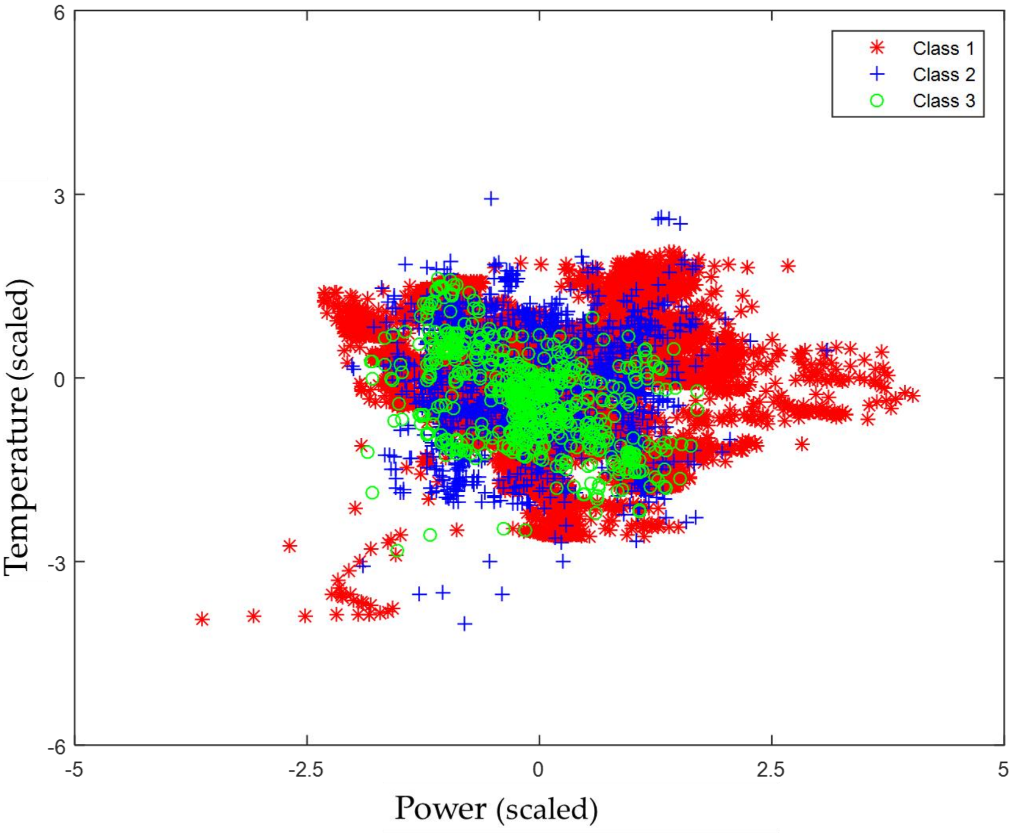


Figure 6-8. The outlet temperature versus power draw plot with corresponding colour legends on the particle size classes.

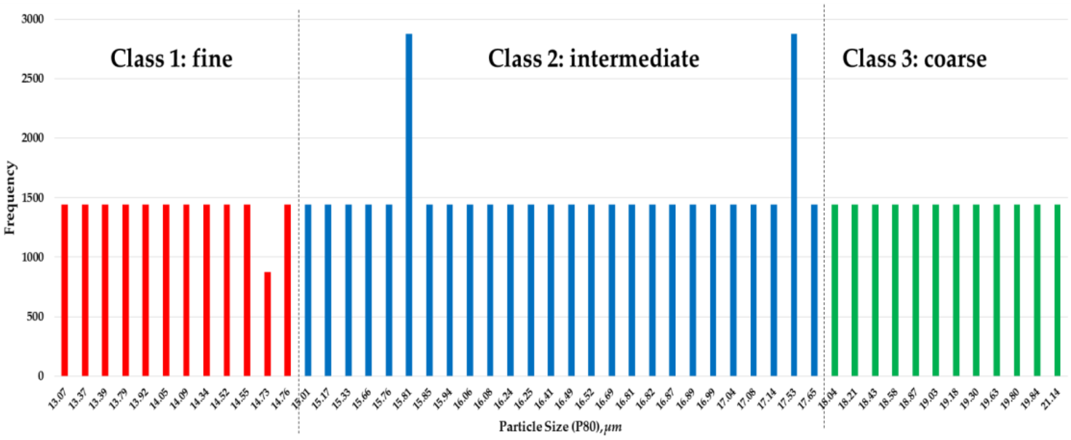


Figure 6-9. The distribution of particle sizes showing the partitions of 3 classes (fine, intermediate, and coarse)

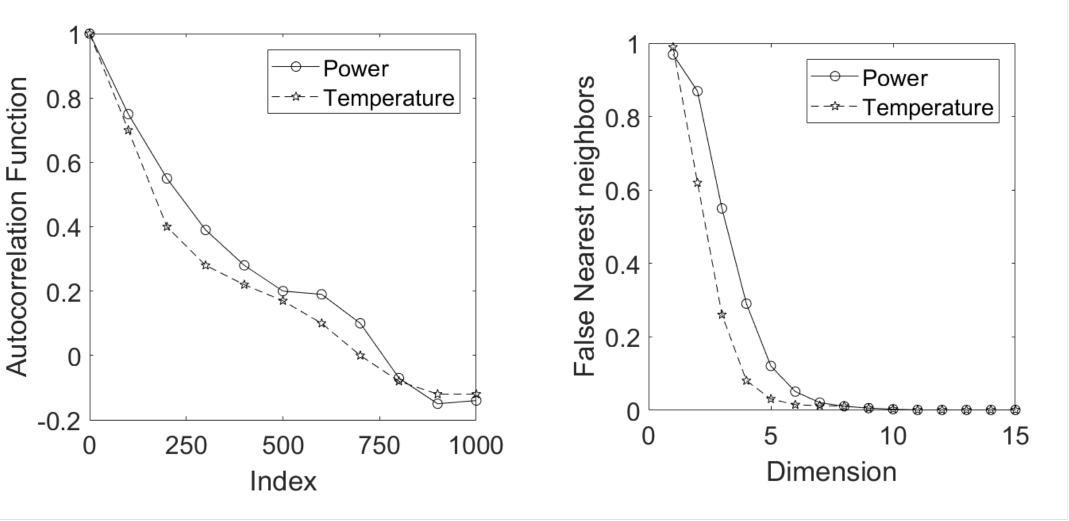


Figure 6-10. The autocorrelation function (left) and false nearest neighbour plots (right) of power draw and outlet temperature time series

Table 6-2. Power Draw and Outlet Temperature data

Class		**Particle size (P80),** ***µm***	**Class mean Particle Size,** ***µm***	**No. of observations**	**No. of segments (using** ***b*****=720,** ***m*****=100)**
1	Fine	13.068 – 14.757	13.91	16,712	160
2	Intermediate	15.014 – 17.647	16.33	40,320	397
3	Coarse	18.037 – 21.139	19.59	17,280	166
	TOTALS			**74,312**	**723**

Three classes are made to correspond the texture of particle sizes (fine, intermediate, coarse). A total of 50 different P80 values are obtained in this undertaking. These P80 values were divided into three that would represent the classes or the texture of the feed. The evaluation of the model is facilitated by first randomly splitting the P80 values into 80% of training data and 20% of testing data. In other words, the training and testing data contain 40 and 10 different P80 values, respectively. Using the training dataset, the models are trained and optimised by controlling the box constraint level of SVM. Further, the cross-validation technique is implemented on the training data sets to validate the results and to avoid the overfitting of the data. Cross validation works by selecting the number cross validation folds. In this case, five folds of cross validation was selected to partition the data where each fold is held out for testing. The model for each fold are trained using all the data outside the fold. Furthermore, the model is tested using the data inside the fold and subsequently calculated the average test error over all folds. In this case, overfitting is minimized, thus, provide good estimate of the feed particle size.

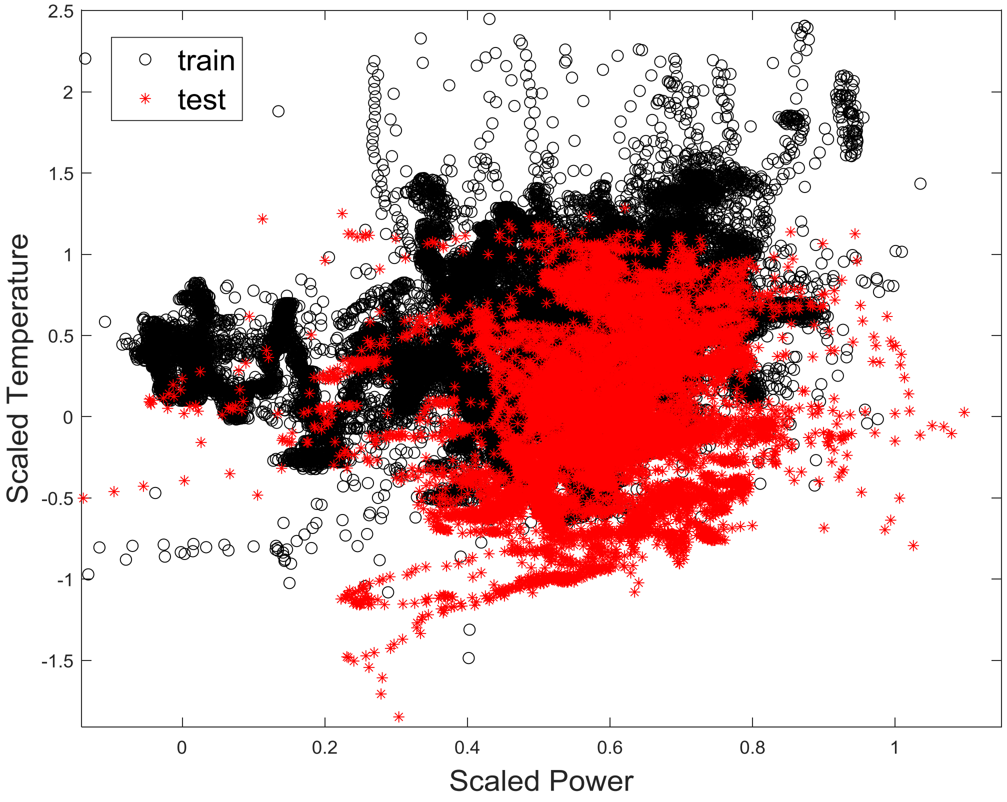


Figure 6-11. Distribution of training (black ‘o’) and test (red ‘*’) data sets.

In the testing stage, the prediction is assisted using these built models and using the test datasets. Classification accuracy were computed by comparing the predicted classes to the true classes of these datasets.

6.3.2 Delay Vector or Lagged Trajectory Coordinates

For comparative purposes, the time series was also analysed by means of phase space reconstruction. This entailed the construction of a set of predictor variables based on lagged copies of the original predictor variables themselves, to enable classification of the time series by models of the form

$$\mathrm{C}(n) \ = f(y(n), y(n-l_e), \dots, y\big(n - l_e(d_e - 1)\big). \tag{6-1}$$

In equation (6-1), $\mathrm{C}(n)$ denotes the class to which the time series observation $y(n)$ belongs at time n. The parameters l_e and d_e are referred to as the embedding lag and embedding dimension of the time series, respectively. The embedding lag parameters were determined via analysis of the autocorrelation function of the time series data, while the embedding dimension was determined with the method of false nearest neighbours, as described by Barnard, Aldrich, and Gerber (2001).

The observations of the variables and their lagged copies yield a so-called lagged trajectory matrix (LTM) that represents a trajectory of the evolution of the dynamical system represented by the observed time series (Aldrich & Auret, 2013).

In this study, these variables $y(n - l_e), \dots, y(n - l_e(d_e - 1)$ were used as predictors to classify the groups of the feed particle sizes.

6.3.2 Results and Discussion

As shown in Figure 6-12, the small distances are represented by dark blue hues, while large distances are indicated by light green and bright yellow. The bright yellow horizontal and vertical lines in the top and middle panels are indicative of possible outliers in the data, but these were not removed. Moreover, it is also interesting to note that the "drift" typology in these plots are apparent, which may mean that the system is non-stationary and may contain a trend or a drift. Verification of this visual interpretation is facilitated by using RTA, particularly texton, CNN-AlexNet, and CNN-VGG16.

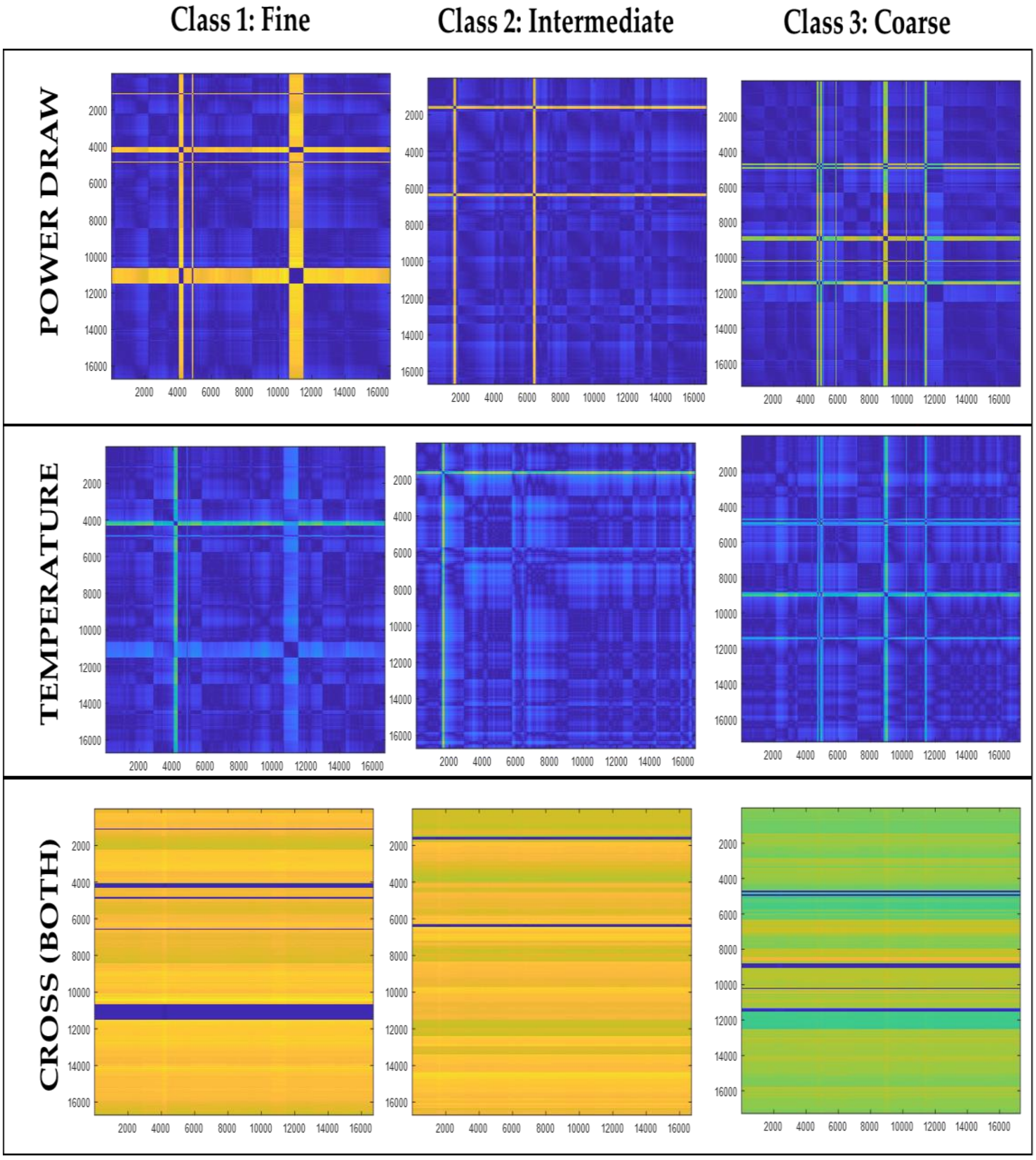


Figure 6-12. Distance matrix plots of 3 classes: fine (left), intermediate (middle) and coarse (right) using time series of power draw (top), outlet temperature (middle) and cross distance of these variables (bottom)

The bottom panels in the figure show the Euclidean distance matrices derived from both variables simultaneously, or the cross distances between observations. That is, the distances were determined from $d_{ij}^{k} = \left\|x_i - y_j\right\|_k^2$, for $i, j = 1, 2, \ldots, b$, where x = power and y = temperature.

A total of 11 feature sets were extracted from the distance matrices of the power variable ($\mathbf{F}_{power}^{tex} \in \mathbb{R}^{723\times37}, \mathbf{F}_{power}^{Alex} \in \mathbb{R}^{723\times4096}, \mathbf{F}_{power}^{VGG16} \in \mathbb{R}^{723\times4096}$), the temperature variable ($\mathbf{F}_{temp}^{tex} \in \mathbb{R}^{723\times37}, \mathbf{F}_{temp}^{Alex} \in \mathbb{R}^{723\times4096}, \mathbf{F}_{temp}^{VGG16} \in \mathbb{R}^{723\times4096}$), cross distance matrices of power and temperature variables ($\mathbf{F}_{cross}^{tex} \in \mathbb{R}^{723\times37}, \mathbf{F}_{cross}^{Alex} \in \mathbb{R}^{723\times4096}, \mathbf{F}_{temp}^{VGG16} \in \mathbb{R}^{723\times4096}$), and feature sets derived from the lagged trajectory (LTM) embeddings of the power and temperature variables$\left(\mathbf{F}_{power}^{LTM} \in \mathbb{R}^{723\times10}, \mathbf{F}_{temp}^{LTM} \in \mathbb{R}^{723\times10}\right)$. The embedding parameters $l_e = 720$ and $l_e = 10$ were determined from the autocorrelation functions and by use of the false nearest neighbour algorithm as shown in Figure 6-10. The above feature matrices were used as predictors in building the classification models.

Since the AlexNet and VGG16 feature sets were comparatively large (4096 features each), the authors also performed dimensionality reduction of these features using principal component analysis, but using the first 20 principal component scores of these large feature sets did not result in better SVM models than when all the features were used.

Five-fold cross-validation was used to partition the data. For each fold, a model was trained using all the data outside the fold, followed by testing using the data inside the fold. The results of using the above features sets to classify the P80 values are shown in Table 6-3 to

Table 6-6.

Table 6-3. Results of classification (% correct) with predictor sets derived from the mill power data and use of a cubic kernel support vector machine.

RUN	POWER

	TEXTONS		ALEXNET		VGG16		LTM	
	Train	Test	Train	Test	Train	Test	Train	Test
1	83.3	60.0	71.3	62.7	76.6	63.4	59.5	60.9
2	86.5	41.9	76.7	47.3	77.9	46.5	65.5	37.1
3	85.0	44.7	74.2	57.5	78.3	48.1	62.7	53.6
4	83.7	64.9	71.6	65.0	73.6	66.4	65.4	55.3
5	84.3	54.2	72.0	58.8	78.1	62.6	62.9	54.6
AVG	**84.6**	**53.1**	**73.2**	**56.3**	**76.9**	**57.4**	**63.2**	**52.3**
STD	**1.26**	**9.80**	**2.28**	**8.39**	**1.96**	**9.35**	**2.46**	**8.96**

Table 6-4. Results of classification (% correct) with predictor sets derived from the mill temperature data and use of a cubic kernel support vector machine.

RUN	**TEMPERATURE**							
	TEXTONS		ALEXNET		VGG16		LTM	
	Train	Test	Train	Test	Train	Test	Train	Test
1	79.1	57.0	72.1	59.2	75.6	59.9	66.3	68.9
2	84.6	36.6	75.5	46.5	77.9	47.3	69.6	55.4
3	79.5	48.8	73.9	47.3	76.3	49.6	70.8	58.3
4	80.3	58.0	78.6	58.0	75.9	62.6	71.3	61.1
5	79.1	63.4	74.8	60.3	76.0	64.1	68.5	63.5
AVG	**80.5**	**52.8**	**75.0**	**54.3**	**76.3**	**56.7**	**69.3**	**61.4**
STD	**2.33**	**10.43**	**2.39**	**6.77**	**0.91**	**7.72**	**2.00**	**5.16**

Table 6-5. Results of classification (% correct) with predictor sets derived from the cross recurrence combination of mill temperature and power data and use of a cubic kernel support vector machine.

RUN	**CROSS**							
	TEXTONS		ALEXNET		VGG16		LTM	
	Train	Test	Train	Test	Train	Test	Train	Test
1	69.9	59.2	59.0	57.8	69.2	63.1	NA	NA
2	76.4	50.3	65.2	45.0	74.2	46.5	NA	NA
3	74.6	46.5	67.1	46.5	74.6	48.0	NA	NA
4	72.8	64.1	59.8	67.9	73.1	66.4	NA	NA
5	72.4	58.8	63.0	61.8	72.2	60.3	NA	NA

AVG	**73.2**	55.8	**62.8**	55.8	**72.7**	**56.9**	NA	NA
STD	**2.44**	**7.18**	**3.45**	**9.87**	**2.15**	**9.05**	NA	NA

Table 6-6. Results of classification (% correct) with predictor sets derived from combination of the mill temperature and power data and use of a cubic kernel support vector machine.

RUN	**COMBINED POWER AND TEMPERATURE**							
	TEXTONS		ALEXNET		VGG16		LTM	
	Train	Test	Train	Test	Train	Test	Train	Test
1	86.1	78.9	80.9	77.5	85.2	78.2	60.5	61.3
2	89.7	45.0	80.6	47.3	86.3	47.3	77.4	41.9
3	91.1	43.1	80.4	51.9	85.7	52.6	75.3	55.9
4	89.0	65.0	80.7	59.8	83.4	66.4	72.6	61.2
5	88.8	74.1	78.9	69.2	86.2	71.8	73.4	51.2
AVG	**88.9**	**61.2**	**80.3**	**61.1**	**85.4**	63.3	**71.8**	**54.3**
STD	**1.83**	**16.5**	**0.80**	**12.4**	**1.18**	**13.0**	**6.60**	**8.10**

As indicated in these tables, it is not possible to distinguish statistically between the different algorithms based on the average values (% overall reliability of the classifiers), owing to the large standard deviations associated with these values. However, since the folds were not selected randomly from the data, and each fold represents a unique portion of the data set, the classifiers could also be scored on their performance based on the folds, with the winner allotted a score of 1 and all the others zero. On this basis, the algorithms could be ranked as follows (with scores in parentheses): VGG16 (7.5), LTM (5), ALEX (1.5), and TEX (1).

It is interesting that the VGG16 performed significantly better than AlexNet over all feature sets, which is in keeping with other comparative studies between these two algorithms. Also of interest is the fact that the VGG16 algorithm scored higher than the LTM approach often used in the development of dynamic process models. This observation is consistent with the premise that deeper neural networks can lead to better classification models in complex decision spaces (Mehdipour Ghazi et al., 2017; Simonyan & Zisserman, 2014)

The reconstructed attractors for both variables can be visualized by plotting the first three dimensions of the trajectory matrix in each case, as shown in Figure 18. These plots are not necessarily able to capture all the variance in the data, but provide some indication of the segregation of the data labels in these phase spaces.

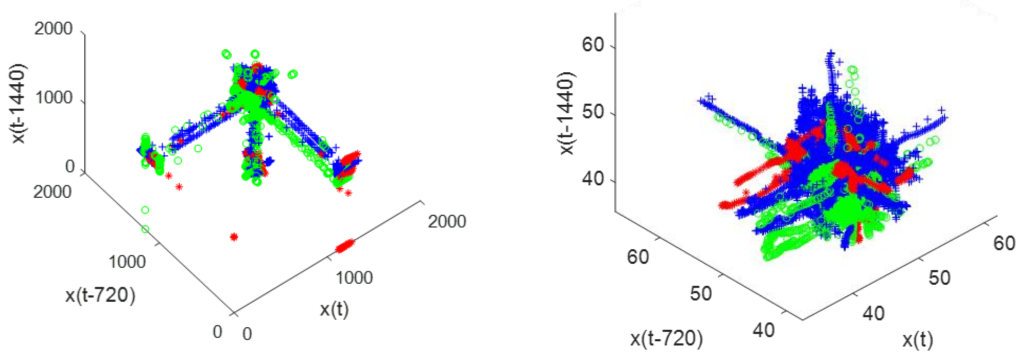


Figure 6-13. The reconstructed attractors of power draw (left) and outlet temperature (right) time series data with legend: fine (red '∗'), intermediate (blue '+'), and coarse (green 'o').

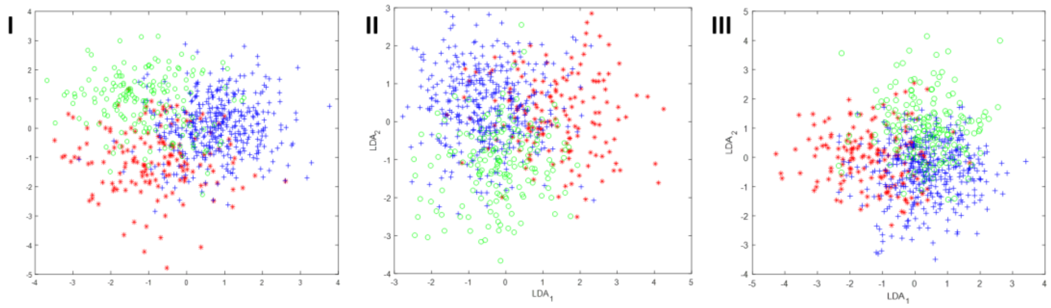


Figure 6-14. Linear projection of combined power draw and temperature texton (I), AlexNet (II) and VGG16 (III) features showing maximum separability between classes: fine (red '*'), intermediate (blue '+') and coarse (green 'o')

6.4 Characterisation of powder flow behaviour

The understanding of flow of a powder is very important in the powder-bed-based additive manufacturing particularly in the development of the process (Spierings, Voegtlin, Bauer, & Wegener, 2016) and to the powder industries, in general, in handling, storage, and processing of powders. Solid flow behaviour is complex in nature owing to its dynamic flow characteristics and to its non-uniformity with respect to its physical and chemical features. The effect of factors on flow behaviour such as shear properties, particle size, and moisture have been identified and are well-understood, but still no single property or index value can reliably describe and quantify the flowability of powders (Ambrose, Jan, & Siliveru, 2016). One way or another, the relative flowability indices (e.g. Hausner's ratio, Carr index, angle of repose, Jenike flow factor) require some measurements of other powder flow properties to characterise the flow of the powders.

The flowability is an integral material property that depends on the dynamic behaviour of the flow of the powder. Most of the time, this also depends on its bulk properties (e.g. moisture content, shape, particle size distribution) which could abruptly change depending on external factors (e.g. temperature, air relative humidity) (Ambrose et al., 2016). Given the importance of powder flow, it is then relevant to examine and characterise it non-linear dynamic behaviour without considering other powder properties and measurements.

A case study is presented here that evaluate the applicability of RTA in capturing the dynamics of the powder flow from flow rate measurements. Both the unreduced and PCA-reduced RTA features are considered in the analysis for comparison purposes. Cluster analysis and classification are also carried out to further assess the applicability of the method.

6.4.1 Powder Flow Data

Five commercially available powders of different particle shapes and sizes and known flowability were considered as test powders in this study. These are Portland cement, cake flour, maize flour, quartz, and table salt. These powders were utilized as subjects to examine the applicability of the proposed approach to characterise the powder flow behaviour.

As shown in the experimental setup in Figure 6-15, the time series were obtained through real-time measurement of the mass of the powders in the digital balance. In particular, the time series data were gathered by allowing the powders to flow through an orifice onto a base plate, from which it overflowed onto a digital balance linked to a computer that logged the mass in grams instantaneously

A sum of 33,914 observations that spread across these powders were recorded. These were subsequently analysed using the recurrence texture analysis. For the segmentation of the time series, a window width of b = 1000 was estimated using the autocorrelation function where the sliding step was chosen to be m =50. The distribution of segments using the said parameters are tabulated in Table 6-7. A total of 580 segments are considered in this study using the estimated windowing dimensions. Each of these segments are transformed into distance matrices using Euclidean distances, from where the textural features are extracted.

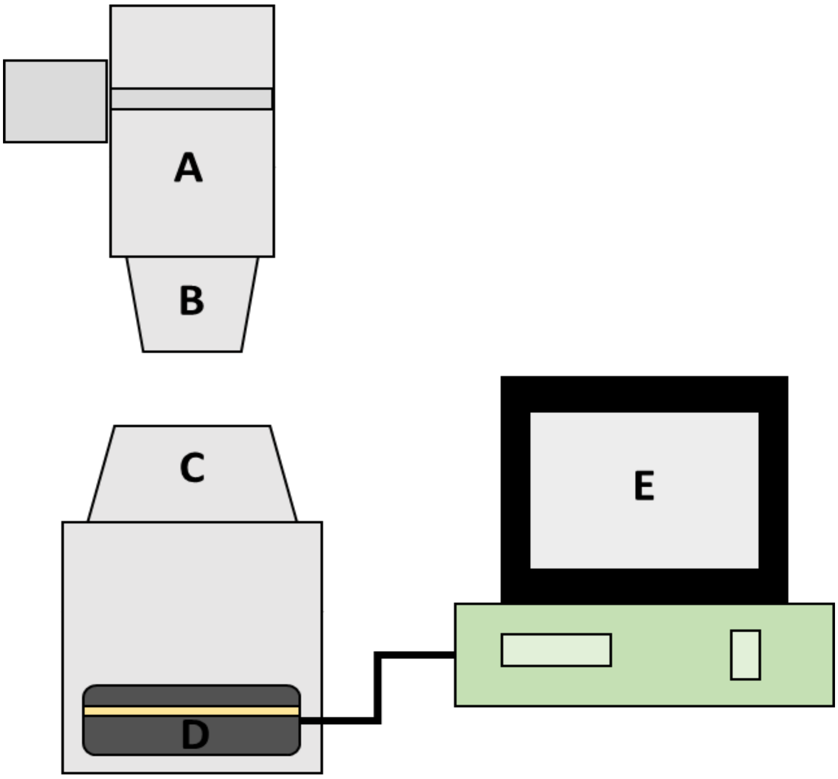


Figure 6-15. Schematic diagram of the experimental setup showing the a vessel (A) powders flow through an orifice (B) onto a base plate (C) and overflow is measured in digital balance (D) connected to computer (E)

Table 6-7. Mass data of the powders

Type of Powder (Acronym)	**No. of observations (*n*)**	**No. of Segments *(N)* (using *b=1000, m=50*)**
Portland Cement (PC)	7,126	123
Cake Flour (CF)	6,733	115
Maize Flour (MF)	10,922	199
Quartz (Q)	6,144	103
Table Salt (TS)	2,989	40
TOTAL	**33,914**	**580**

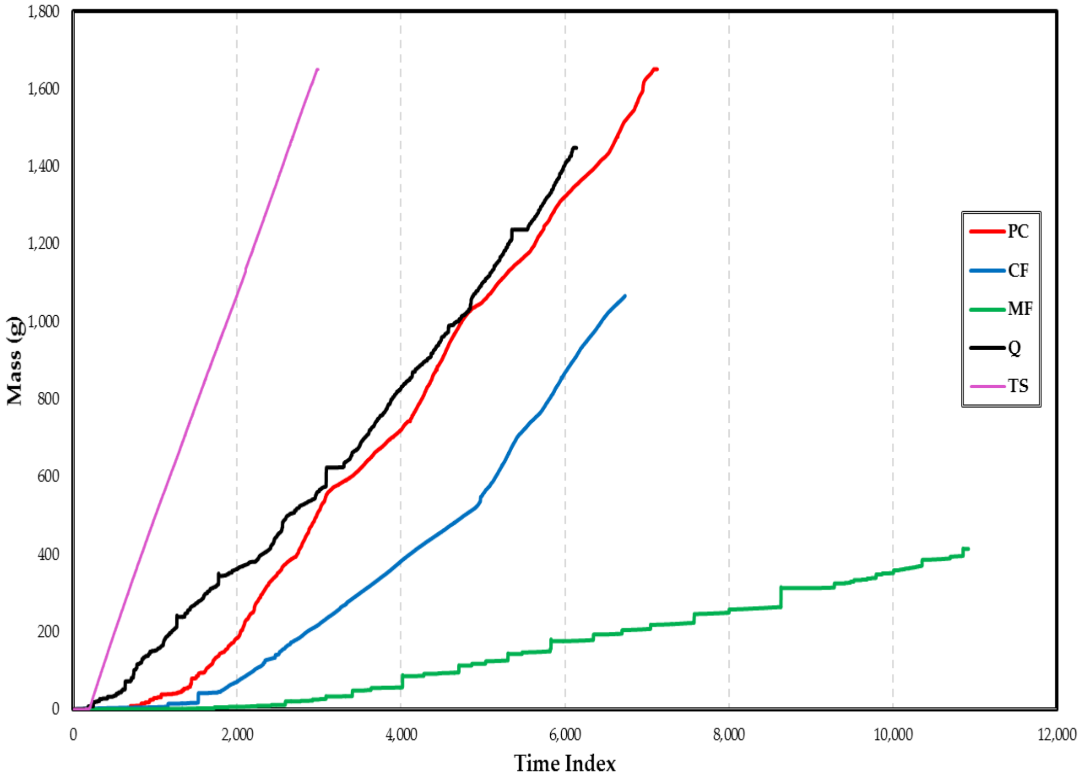


Figure 6-16. The raw time-series mass data of the five powders

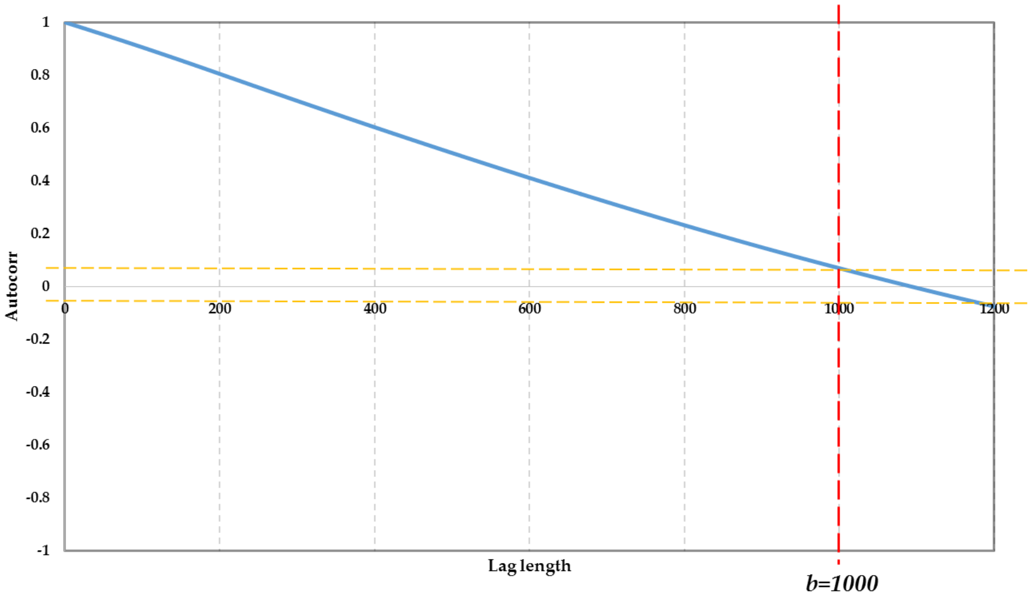


Figure 6-17. The autocorrelation function of the powder time series data, showing b=1000.

6.4.2 Results and Discussion

The distance matrices of each powder is presented in Figure 6-18. Visually, in general, it seemed that the distance matrices of PC, CF and TS are quite identical with regards to its features. The distance matrices of Q and MF, on the other hand, appear unique with the latter has the coarsest distance matrix among all of them.

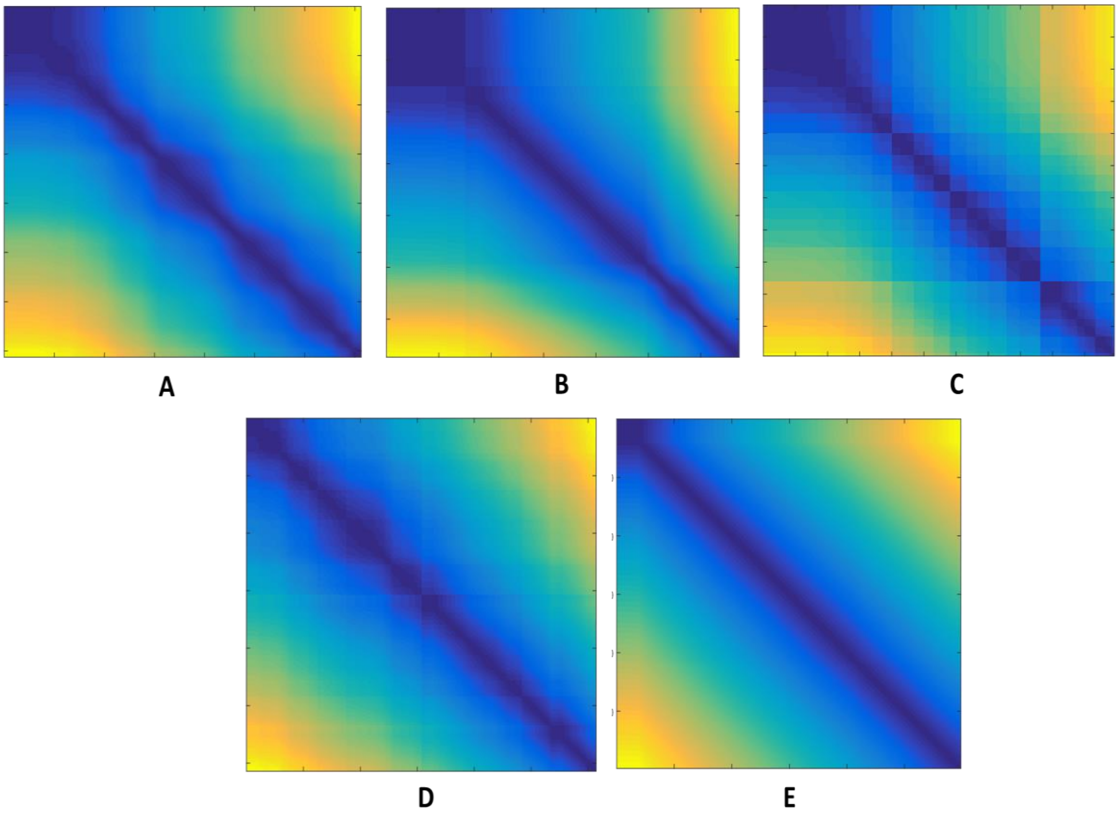


Figure 6-18. Euclidean distance matrices (with colour maps) of the Portland cement (A), Cake flour (B), Maize flour (C), Quartz (D), and Table salt (E)

It is apparent that the distance matrix features of PC and CF are quite similar, with some resemblance to that of the TS. The textures are relatively finer. On the contrary, the textures of MF distance matrix are coarse, while the distance matrix of Q is a bit finer than MF but looks coarser than the other three. Coarse textures are related to a tendency to avalanche.

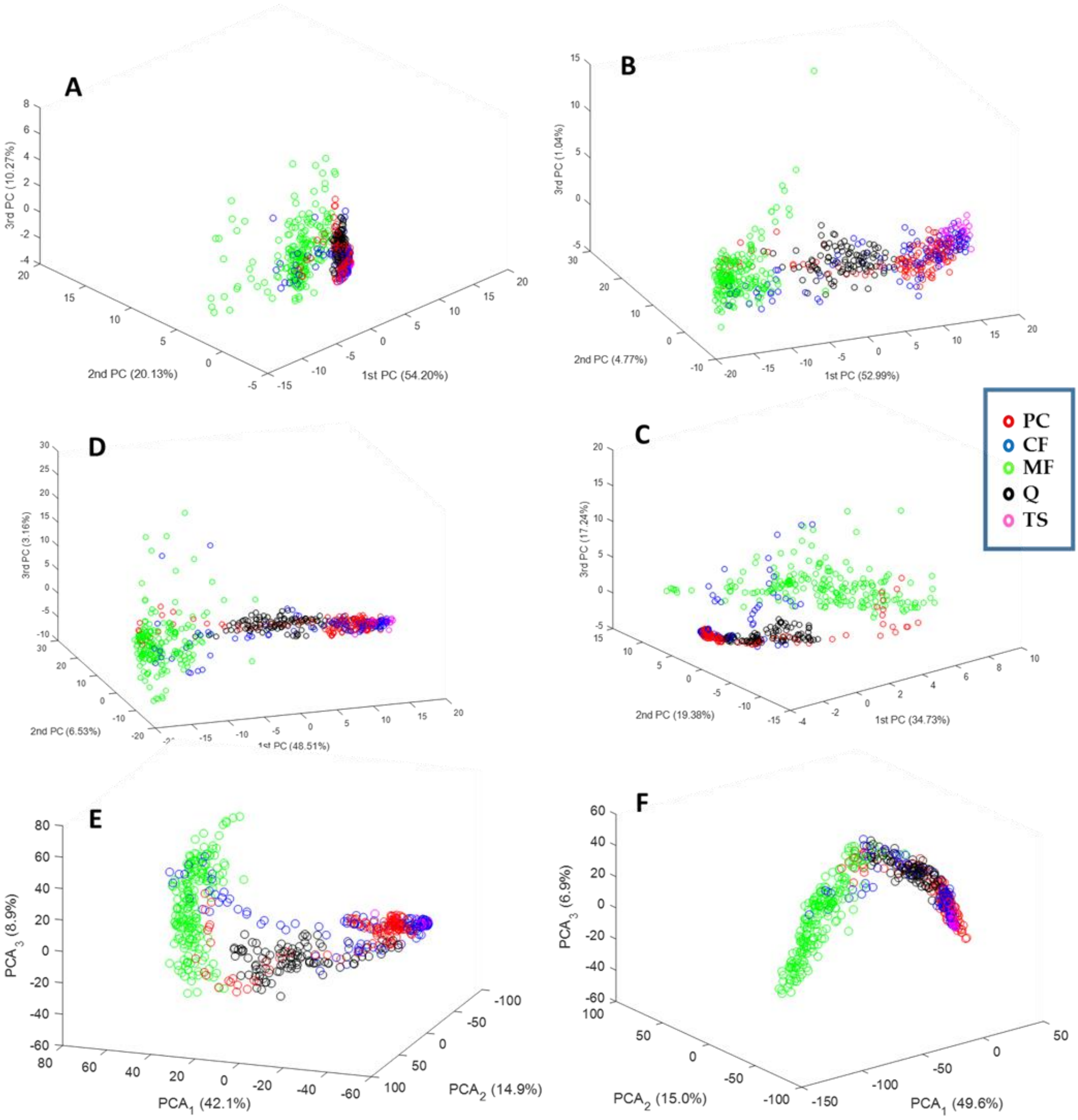


Figure 6-19. Textural features of (A) GLCM and wavelet, (B) LBP, (C) texton, and (D) combined GLCM, wavelet, LBP and texton, (E) AlexNet, and (F) VGG16 as projected onto 3-D principal component subspace

To further study these distance matrices, its textural features were extracted. Six texture matrices were obtained, coming from the considered textural extraction and CNN algorithms. In this paper, however, the GLCM and wavelet features were combined into a single feature matrix due to their lesser dimensionality relative to other features. In addition, the GLCM, wavelet, Texton and LBP feature matrices were combined to a single matrix for comparison. These were analysed thoroughly, by visualisation of data and by classification models. The features are visualised by projecting the features into first 3-D principal component subspace, as presented in Figure 6-19.

From Figure 6-19, although the clusters of portland cement, cake flour, and table salt are difficult to identify due to overlapping, it can still be inferred that distinct clusters are apparent to maize flour (green) and quartz (black). Remarkably, the maize flour are well separated from the other groups for all features. The cluster of quartz, on the other hand, is hard to localise in GLCM + Wavelet features but are distinct and quite separable to other clusters (with some overlapping) in other features. It is interesting to note that these observations supported the visual interpretation of the features of distance matrices, that the maize flour, has the coarsest distance matrix, followed by quartz, while the other three have finer textures and quite similar to one another.

Another stimulating inference in these plots is the separability of the clusters in each plot. In particular, the texton and AlexNet are said to have provided better visualisation of data due to enhanced separations of clusters present in the plot. The clusters of maize flour are very distinct in this case. For cake flour, although some of the points overlapped to other clusters, its cluster is still identifiable. This observation is also the same to Portland cement. On the contrary, the cluster of table salt is difficult to identify. Thorough inspection revealed that a well-formed cluster for this powder is existed as well, but was found in the cluster of Portland cement, this it was hard to see in the plot. It could be, however, be seen in higher dimensional space (e.g. 4-D) but visualisation of data is then limited in this case. As an alternative, the use of feature selection that maximises inter-class discrimination can be performed such as Linear Discriminant Analysis (LDA). In this case, LDA was used here in visualising the data. It should be noted, however, that LDA or any techniques that accentuate maximising inter-class discrimination (e.g.

Eigenvalue-based Mutual Information (EMI)) should be used with caution especially when dealing classification of high dimensional image data as it could result to overfitting of data (R. Liu & Gillies, 2016). Overfitting of data is normally occurred when the sample size is not large enough relative to its dimensionality. In this study, there are 580 segments that were analysed, big enough to minimise overfitting. Furthermore, the paper only used LDA as supplemental tool for visualisation of data. As presented in Figure 6-20, distinct clusters are observed, with a varying extent of separability among the clusters for each feature set.

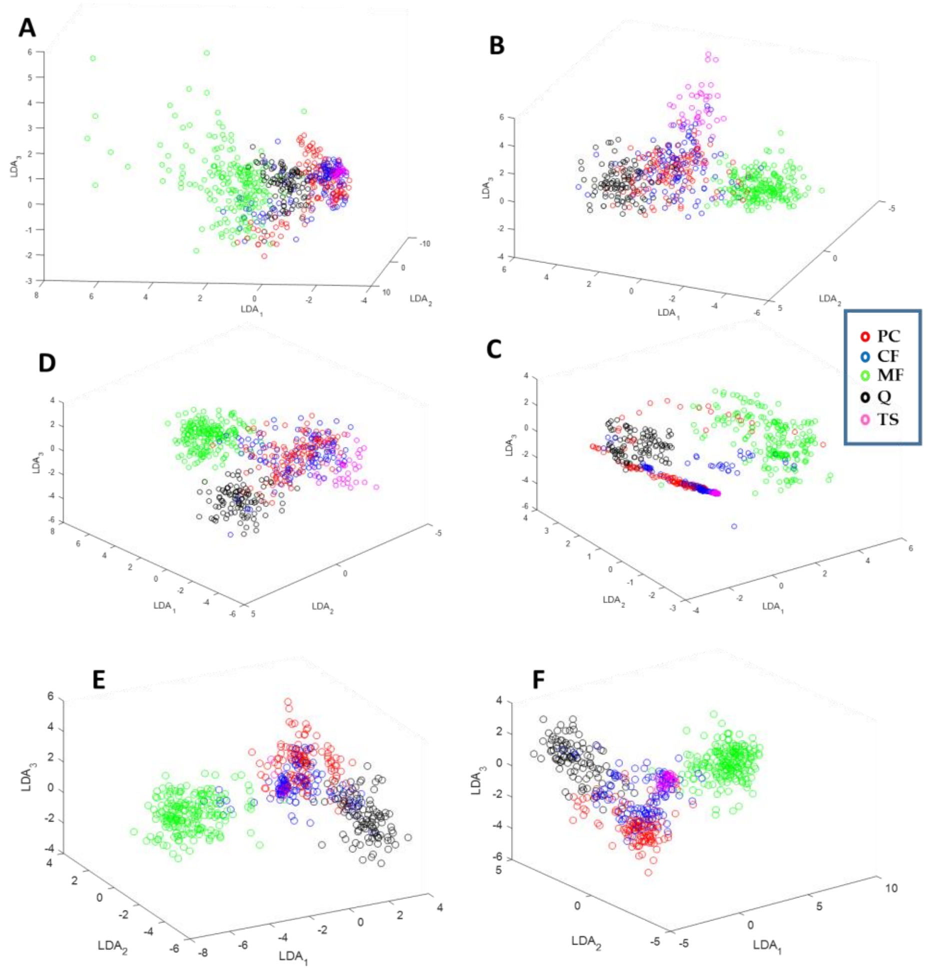


Figure 6-20. Linear discriminant projection of RTA features namely (A) GLCM and wavelet, (B) LBP, (C) textons, and (D) all features (GLCM, wavelet, LBP, textons), showing maximum separability between the powders.

Table 6-8. Classification performance of the feature sets (highlighted the highest classification accuracy in the test dataset)

	GLCM + Wavelet	LBP	Texton	GLCM + Wavelet + LBP + Texton	AlexNet	VGG16
No. of dimensions	17	256	40	313	4096	4096
No. of segments, *N*	580	580	580	580	580	580
No. of classes	5	5	5	5	5	5
Classification Accuracy, %	81.6%	77.0%	**94.8%**	79.3%	93.3%	94.6%

To characterise the powder flow behaviour, a linear-SVM classification model is developed. The analysis is performed using five classes that correspond to the five powders, with the SVM as classifier. The classification accuracies are determined for each run using the sets of the features using holdout method. In this process, the datasets are randomly divided into training (70% of the datasets) and testing (remaining 30% of the datasets). Ten-fold cross validation is implemented in the validation stage to minimise overfitting. The trained models are then employed to predict the classes of the test data.

As presented in Table 7-5, it can be understood that, although all the features provided reliable results by getting more than 70% classification accuracy, the texton, and 2 CNN features have seen to outperform the other features including the combination of GLCM, wavelet, texton and LBP features. In other words, these three features gave the most discriminative features for each powder by obtaining more than 90% classification accuracy.

Comparison to Recurrence Quantification Analysis

To supplement the study, the RQA is also employed in this case and the results are compared to the RTA. As shown in Figure 6-21, the recurrence plots of Portland cement, cake flour and table salt are apparently comparable to each other. In contrast, the recurrence plot of maize flour can easily be differentiate to other recurrence plots. In the analysis, 10 RQA features were extracted from these recurrence plots. Figure 6-22 visualised these features in 3-D principal component subspace. As seen in the figure, the maize flour can be distinguished easily from other powders. The RQA features of quarts, on the other hand, are fairly scattered in the plot which were seen overlapping to most RQA features of other powders.

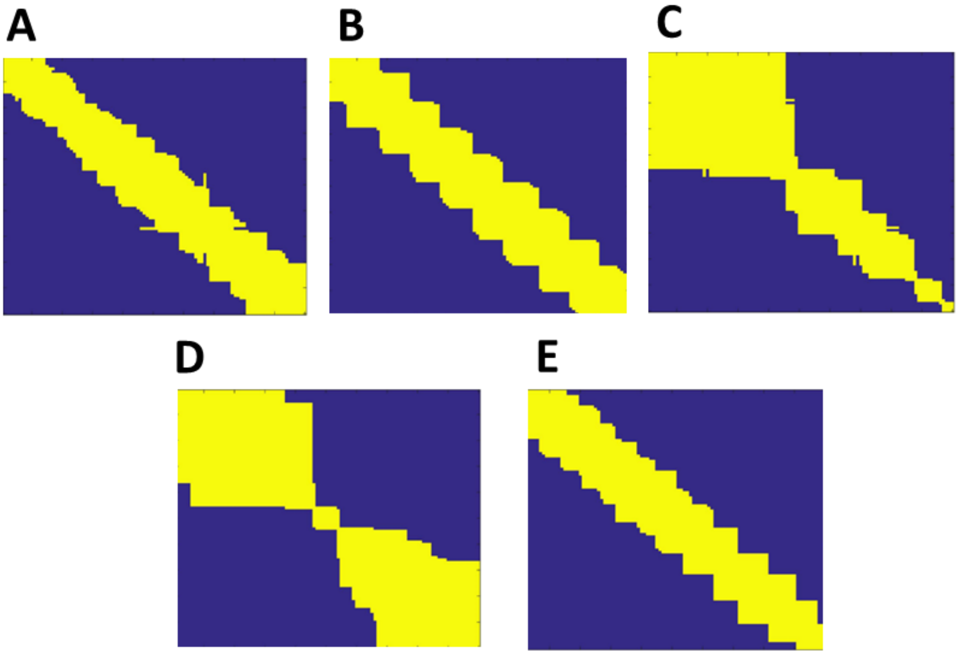


Figure 6-21. Exemplary samples of recurrence plots of (A) Portland cement, (B) cake flour, (C) maize flour, (D) quartz, and (E) table salt.

These RQA features were treated as predictors in SVM classification model. Using the same procedure with RTA in the training and testing, the model successfully classified the classes of the powders using RQA features with accuracy of 81.1%. This result showed that RQA features could also be used to discriminate the classes of the powder which can be used in the development of the powder flowability.

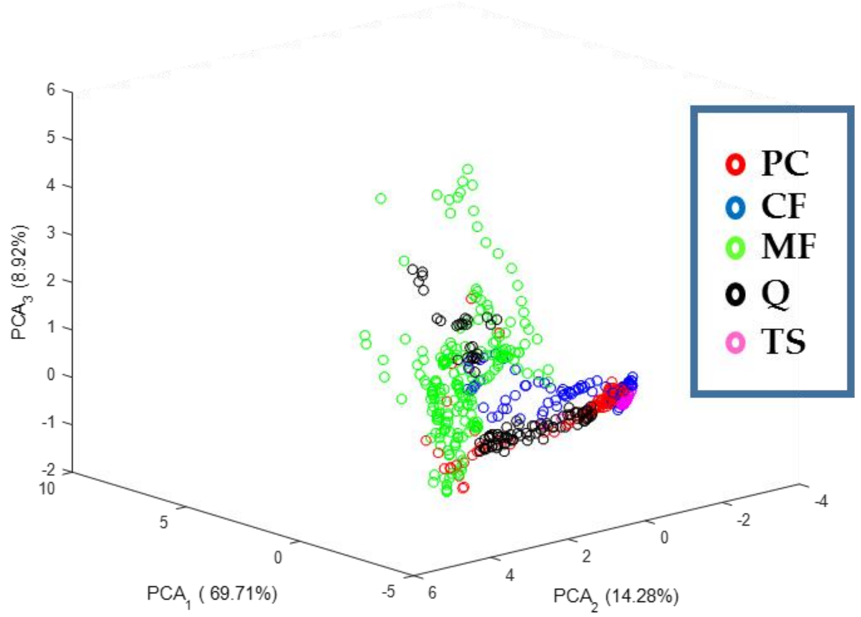


Figure 6-22. 3-D plot of the RQA features using the first 3 principal component scores.

In this comparative study, it has seen that the RTA-texton, RTA-AlexNet and RTA-VGG16 performed the best in classification with less than 7% error rate. Furthermore, the combined GLCM and wavelet also performed well, which has seen to be at par to the RQA features, which have both obtained 81% accuracy. More importantly, this comparative study displayed the robustness of RTA over RQA in characterising the dynamic behaviour of the process data.

6.5 Conclusion and Recommendations

The RTA was employed in this study to characterise the dynamic behaviour of the three sets of solids processing time series data sets, which constituted the three case studies. From the analysis, in general, it was inferred that the RTA demonstrated a robust and reliable technique in capturing the dynamics of these solids processing data. In other words, the case studies showed the applicability of RTA in characterising the dynamic behaviour of the processes. Specifically, the following insights are gathered:

- As suggested with the results, the mill load could potentially be used as a sole parameter for monitoring the conditions of autogenous mill. Conversely, with the encouraging outcome in the first case study, it could also be discerned that RTA has a huge opportunity for possible application in process control and monitoring. The proposed method successfully classified the mill states using the mill load data.
- The results of the second study proposed that the combined power draw and temperature, with incorporation of other mill parameters is a reliable parameters for possible estimation of feed particle size. The use of the method revealed that the use of CNN-VGG16 features could consistently classify the groups of the sizes of the feed which yielded the best performance overall. The models presented in this work, could in principle be used and implemented online once calibrated, where real-time measurement of particle size is generated. Ultimately, with this study, it proved that RTA could form the basis for more development of models for applications in autogenous mill and in minerals processing, in general.
- The general approach compared favourably with a traditional approach in the identification of product size in a horizontal stirred mill, although only marginally so. Although the particle sizes could not be identified with a high degree of accuracy in this case, the models could likely be improved substantially with the availability of more plant data.

- The third case study showed the applicability of the RTA to characterise the dynamic behaviour of the powder flow. Once suitably calibrated, the method could then be applied as a complement in powder flow measurements available in the market. Additionally, in this case study, it was showcased that the mass time series is a reliable variable for possible monitoring and measurement of the powder flow, which gave some ideas that the mass can be incorporated in the development of powder flow indices. Moreover, the reliability of RTA over RQA in the classification of process data was clearly demonstrated in this part.
- Finally, it should be noted that the methodology could be used for time series analysis in general, including forecasting, classification, and clustering, and this should be confirmed by further investigation on larger and more general data sets and systems.

7. APPLICATION: DYNAMIC PROCESS MONITORING

7.1 Introduction

Multivariate statistical process monitoring is key to enhanced process safety, process efficiency, and meeting product quality criteria by controlling process variation. To this end, large volumes of logged plant data are used to build models to enable fault detection and identification in process operations. Models based on principal component analysis (PCA) and partial least squares (PLS) are commonly used (Groenewald, Coetzer, & Aldrich, 2006; Joe Qin, 2003). However, since PCA and PLS models can give misleading information when applied to highly nonlinear systems exhibiting non-Gaussian behavior, a variety of nonlinear extensions to these models have been proposed.

These include principal curves (Dong & McAvoy, 1996; Harkat, Mourot, & Ragot, 2003; Shi, Lv, Fei, & Liang, 2013), independent component analysis (Hsu, Chen, & Liu, 2010; Kano, Tanaka, Hasebe, Hashimoto, & Ohno, 2003; L. Wang & Shi, 2010), kernel methods (Deng, Tian, & Chen, 2013; Jemwa & Aldrich, 2005; Zhang & Qin, 2007), neural networks (Aradhye, Davis, & Bakshi, 2002; Groenewald & Aldrich, 2015; Jia, Martin, & Morris, 1998), etc. Despite the improvements obtainable by these approaches, they are not necessarily designed to deal with the non-stationary and dynamic character of many process systems and can fail when measurements are strongly autocorrelated (Ku, Storer, & Georgakis, 1995).

Some of the methods designed to handle these continuous dynamic processes attempt to capture the structure of the time series data by embedding the data into a phase space of a pseudophase space, such as is the case with dynamic principal component analysis, multidimensional scaling

(Auret & Aldrich 2010; Zhang, 2014), singular spectrum analysis (Krishnannair, Aldrich, & Jemwa, 2016), etc.

In this chapter, the Recurrence Texture Analysis is used to capture the structure of the time series data that subsequently used as a basis in the development of a dynamic process monitoring scheme based on principal component model. To validate the applicability of the method, two case studies are presented using two simulated and benchmark data sets: the Lotka-Volterra predator-prey system and the Tennessee Eastman problem. The dynamic process monitoring system and the results of this study are described in details in this chapter.

7.2 Dynamic Process Monitoring System

The process monitoring system proposed here was based on a principal component model of the feature matrix that was extracted using the method. The general methodology of the dynamic process monitoring scheme is presented in Figure 7-1. As an overview, the Recurrence Texture Analysis was first employed in time series data, which comprised of dividing the time series into a set of contiguous, time series segments and calculating the Euclidean distance matrix for each time series segment. These distance matrices are subsequently treated as images and algorithms used in textural image analysis are used to derive a set of descriptors for each time series segment. These descriptors or variables were then collected into a data matrix which were to construct principal component models using principal component analysis. Process diagnostics were then derived using the models. Both off-line calibration and on-line application of the models were carried out as part of the dynamic process monitoring system.

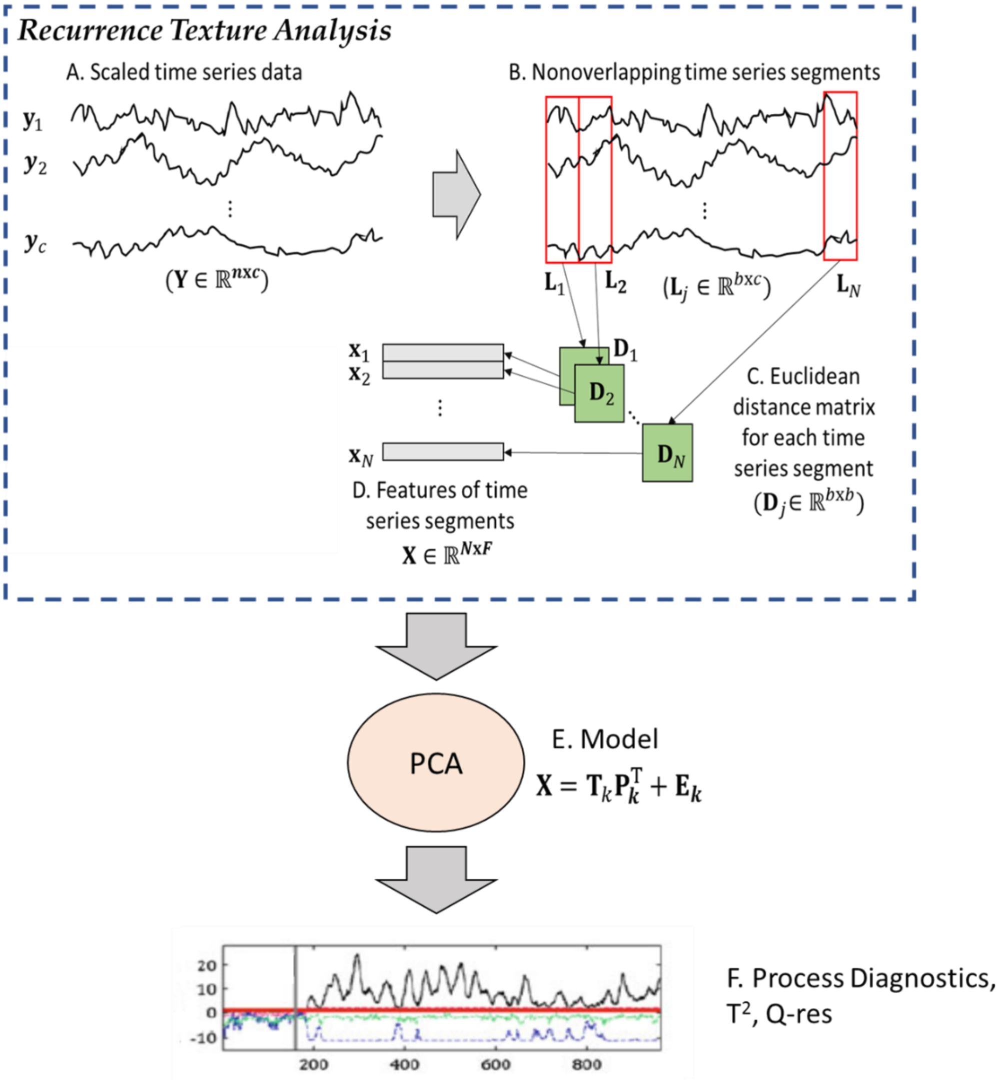

Recurrence Texture Analysis
A. Scaled time series data
y_1
y_2
y_c
($\mathbf{Y} \in \mathbb{R}^{nxc}$)
B. Nonoverlapping time series segments
$\mathbf{L}_1$
$\mathbf{L}_2$
($\mathbf{L}_j \in \mathbb{R}^{bxc}$)
$\mathbf{L}_N$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_N$
$\mathbf{D}_1$
$\mathbf{D}_2$
$\mathbf{D}_N$
C. Euclidean distance matrix for each time series segment
($\mathbf{D}_j \in \mathbb{R}^{bxb}$)
D. Features of time series segments
$\mathbf{X} \in \mathbb{R}^{NxF}$
PCA
E. Model
$\mathbf{X} = \mathbf{T}_k\mathbf{P}_k^{\mathrm{T}} + \mathbf{E}_k$
20
10
0
-10
200
400
600
800
F. Process Diagnostics,
T², Q-res

Figure 7-1. Extraction of features from time series data (A), by segmentation into nonoverlapping segments (B), derivation of Euclidean matrices for all the segments (C) and extraction of features from these distance matrices (D). These features or variables can subsequently be used to build models (E) and to derive diagnostics (F) for process monitoring.

7.2.1 Composition of Feature matrix

Two texture extraction algorithms, i.e. GLCM and wavelet, were considered in this undertaking. These GLCM and wavelet features or textural descriptor variables that were extracted from the distance matrices of the time series segments were aggregated to form a feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$, where N is the number of time series segments, and F is the number of GLCM features

(C) and wavelet features (W) combined, i.e. $F = C + W$. In other words, the time series matrix $\mathbf{Y} \in \mathbb{R}^{n \times c}$ is converted to a data matrix with dimensions $N \ll n$ and $F \gg c$.

The parameters used during GLCM and wavelet feature extraction are summarised in Table 7-1.

Table 7-1. Hyperparameters used in GLCM and wavelet feature extraction.

Algorithm	Hyperparameter	Value
GLCM	Number of grey levels	32
	Distance	1
	Directions (degrees)	[0, 45, 90, 135]
Wavelet	Type of wavelet	Daubechies (db4)
	Level of wavelet decomposition	3

7.2.2 Off-line Calibration of the Principal Component Model

Given that $\mathbf{X} \in \mathbb{R}^{N \times F}$ is the feature matrix representative of the dynamic process, $\mathbf{S} \in \mathbb{R}^{F \times F}$ is the covariance matrix of the feature variables scaled to zero mean and unit variance, $\mathbf{P}_k \in \mathbb{R}^{F \times k}$ is the loading matrix of the first $k < F$ principal components, $\mathbf{\Lambda}_k \in \mathbb{R}^{k \times k}$ is a diagonal matrix containing the k eigenvalues of the decomposition, $\widetilde{\mathbf{P}}_k \in \mathbb{R}^{F \times k}$ is the loading matrix of the F-k remaining principal components and $\widetilde{\mathbf{\Lambda}}_k \in \mathbb{R}^{k \times k}$ is a diagonal matrix containing the F-k remaining eigenvalues of the decomposition. Off-line model calibration is done with data ($\mathbf{X}$) associated with normal operating conditions.

$$\mathbf{S} = \frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{n-1} = \mathbf{P}_k\mathbf{\Lambda}_k\mathbf{P}_k^{\mathbf{T}} + \widetilde{\mathbf{P}}_k\widetilde{\mathbf{\Lambda}}_k\widetilde{\mathbf{P}}_k^{\mathbf{T}} \tag{7-1}$$

$$\mathbf{\Lambda}_k = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_k \end{bmatrix} \tag{7-2}$$

$$\mathbf{T}_k = \mathbf{XP}_k \tag{7-3}$$

In eqn (7-1), $\mathbf{T}_k \in \mathbb{R}^{b \times k}$ is the score matrix of the principal component model consisting of the first k principal components (PCs). The number of PCs retained (k) is determined using cumulative percent variance (CPV) method which was described in Chapter 4.

7.2.3 Process Diagnostics and Control Limits

Using Hotelling's T² and Q-statistics

The Hotelling's ${T_k}^2$ and the Q_k statistics are most often used with principal component models, i.e.

$$Q_k = (\mathbf{x} - \hat{\mathbf{x}}_k)^{\mathrm{T}}(\mathbf{x} - \hat{\mathbf{x}}_k) = \mathbf{x}^{\mathrm{T}}\mathbf{Cx}, \text{ where } \mathbf{C} = \mathbf{I}_M - \mathbf{P}_k{\mathbf{P}_k}^{\mathrm{T}} \tag{7-4}$$

$${T_k}^2 = {\mathbf{t}_k}^{\mathrm{T}}\mathbf{\Lambda}^{-1}\mathbf{t}_k = \mathbf{x}^{\mathrm{T}}\mathbf{Dx}\,, \text{ where } \mathbf{D} = \mathbf{P}_k{\mathbf{\Lambda}_k}^{-1}{\mathbf{P}_k}^{\mathrm{T}} \tag{7-5}$$

where $k \leq 2$, the Q_k diagnostic can be combined with the first two principal component score vectors (i.e. the first two columns of $\mathbf{T}_k$), $\mathbf{t}_1$ and $\mathbf{t}_2$, either in a three-dimensional map ($\mathbf{t}_1$, $\mathbf{t}_2$, Q_2) or in two separate maps, viz. a scatterplot of $\mathbf{t}_1$ versus $\mathbf{t}_2$, as well as Q_2 versus time or sample index.

Using Gaussian Mixture Models

Where applicable, the Gaussian mixture models (GMMs), which were previously described in Chapter 5, were also fitted to the NOC data in the $\mathbf{t}_1$-$\mathbf{t}_2$ score space.

In one-class GMM, the training set is used to define the Gaussian mixture model $p(x|\Theta)$ with the optimised K Gaussians. This creates a decision criteria and decision boundary onto the dataset. The object is to be considered as target if the trained Gaussian mixture model $p(x|\Theta)$ is equal or greater than probability distribution parameter of the i-th Gaussian component, θ_i. Otherwise, the object is classified as outlier or fault (Tax, 2013).

7.2.4 On-line Application of the Model

During process monitoring, newly measured data $\mathbf{Y}_{new} \in \mathbb{R}^{n_{new} \times c}$ are also analysed RTA. In other words, the data are also segmented to yield $\mathbf{L}_{new,i} \in \mathbb{R}^{\boldsymbol{b} \times \boldsymbol{c}}$, for i = 1, 2, … N_{new} from which N_{new} distance matrices $\mathbf{D}_{new,i} \in \mathbb{R}^{b \times b}$ are calculated. Features are subsequently extracted from the distance matrices, to yield a new feature matrix $\mathbf{X}_{new} \in \mathbb{R}^{b_{new} \times F}$. These data are projected onto the principal component model, to generate the principal component scores, $\mathbf{t}_{new,i}$ associated with the new observations for $i = 1, 2, \ldots b_{new}$, i.e.

$$\mathbf{t}_{new,i} = \mathbf{x}_{new,i}\mathbf{P}_k \qquad (7\text{-}6)$$

If this observation falls outside the normal operating condition limits of the model, it is flagged as a process deviation or fault condition. A flowchart illustrating the general approach is shown in

Figure *7-2*.

7.2.5 Performance Metrics

To determine the performance of the proposed monitoring system, several metrics are measured during the testing stage. These were the false alarm rate (FAR), missing alarm rate (MAR), alarm run lengths (ARL), and an area-under-the-curve criterion, as discussed below. For comparison purposes, the diagnostic thresholds were set to correspond to a 5% false alarm rates during the validation stage for all diagnostic statistics.

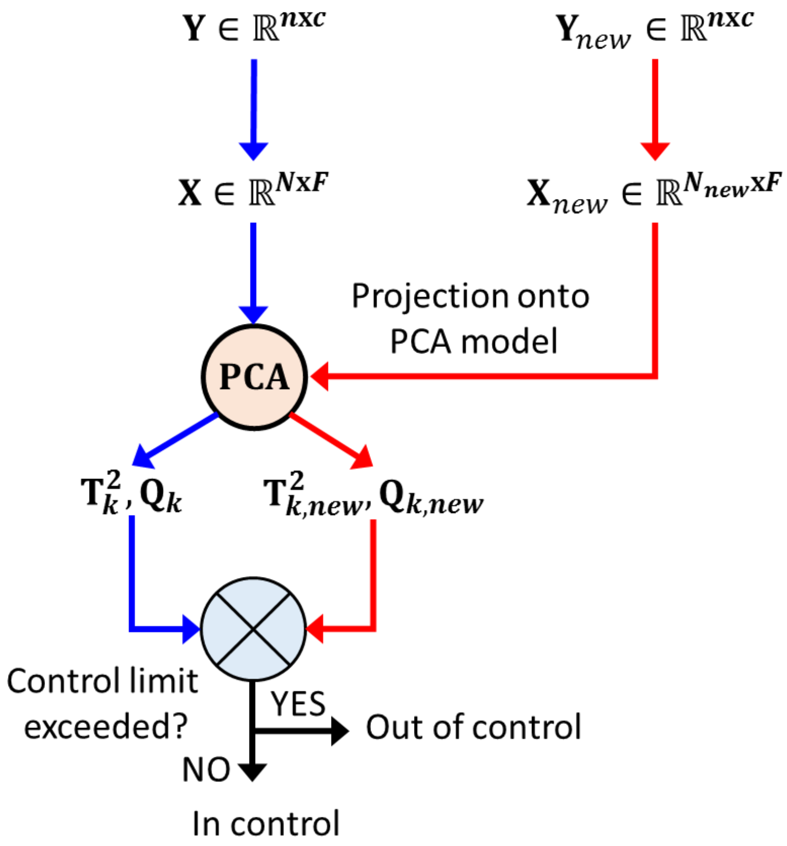

$\mathbf{Y} \in \mathbb{R}^{nxc}$
$\mathbf{Y}_{new} \in \mathbb{R}^{nxc}$
$\mathbf{X} \in \mathbb{R}^{NxF}$
$\mathbf{X}_{new} \in \mathbb{R}^{N_{new}xF}$
Projection onto PCA model
PCA
$\mathbf{T}_k^2, \mathbf{Q}_k$
$\mathbf{T}_{k,new}^2, \mathbf{Q}_{k,new}$
Control limit exceeded?
YES
Out of control
NO
In control

Figure 7-2. General flowchart showing the online application of the dynamic process monitoring model

False Alarm Rate (FAR)

The FAR is the ratio of NOC samples that are identified as faulty (NF) to the total number of NOC samples (TN), i.e.

$$FAR\ (\%) = \frac{NF}{TN}\ x\ 100\% \tag{7-7}$$

Missing Alarm Rate (MAR

The MAR, is the ratio of the known fault samples (FN) that are not detected to the total number of faulty data (TF).

$$MAR\ (\%) = \frac{FN}{TF} x\ 100\% \tag{7-8}$$

Alarm Run Lengths (ARL)

To better understand the FAR and MAR, the alarm run lengths are also determined. Typically, the ARL is computed for both false alarm and true alarm. The ARL for false alarm is the number of consecutive observations before an alarm is triggered for NOC data (TN). Generally, it is desirable to get a larger ARL for false alarm.

The ARL for true alarm, on the other hand, refers to the number of consecutive samples before an alarm is made for faulty data (TF). In this sense, the smaller ARL (true) is desired for an alarm system.

Receiver Operating Characteristics (ROC) and the Area-Under-The-Curve (AUC) criterion

The ROC curve illustrates the performance of classifiers by showing the relationships between the number of NOC data and the faulty data that were accepted in the calculation. It is a parameterized curve, where a one-sided detection threshold is the curve parameter. Normally, the ROC curve is a plot of true alarm rates (TAR) against the false alarm rate (FAR). An ideal classifier is found in the top left corner of the ROC space with a TAR value of 1 and a FAR value of 0. Thus, the closer a certain classifier to the top left of the ROC space, the better its performance (Aldrich & Auret, 2013).

More often than not, the ROC curve is challenging to interpret, as the metric is only determined qualitatively. To quantify it, the AUC can be calculated. The higher the AUC value, the better the classifier (Tax, 2013).

7.3 Case 1: Lotka-Volterra Predator-prey system

The Lotka-Volterra model explains the interaction between predator and prey species, as exemplified in eqns (4-1) and (4-2). In this study, a univariate time series of the system was simulated with the ODE45 subroutine in MATLAB R2015a by considering the change in $\dot{x}$ with time.

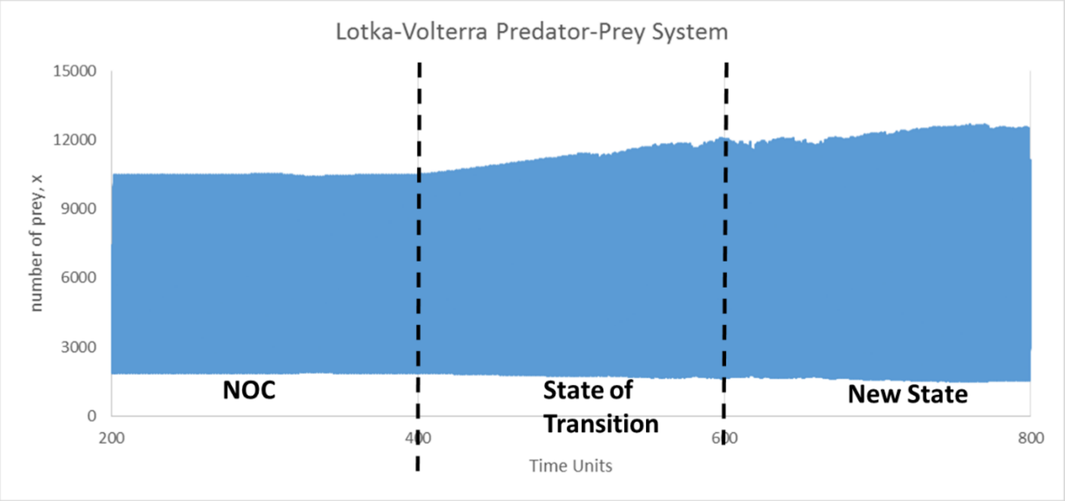


Figure 7-3. Simulated observations of the Lotka-Volterra predator-prey model.

The time series used here, as shown in Figure 7-3, are similar to the one used in Chapter 5 wherein the simulation was done with equal time steps of 0.02 time units resulting to 40,000 samples that spread across three state systems, i.e. NOC, state of transition and new state. However, the first 10,000 are not considered in the analysis since the system is said to have unstable state in those time frame. The parameters used are summarised in Table 7-2.

Parameters	NOC	State of Transition	New State
ẋ	5,000	5,000	5,000
ẏ	100	100	100
k1	2	2 to 3	3

k2	10	10 to 11	11
$\hat{C}$	0.0010	0.0010 to 0.0011	0.0011
$\hat{D}$	0.0020	0.0020 to 0.0021	0.0021
Time units	0 to 400	400 to 600	600 to 800

Table 7-2. Parameter variations used in three condition states (NOC, transition and new).

7.3.1 Results

Representative plots of the distance matrices are shown in Figure 7-4. The coarser structure of the distance matrix associated with normal operating conditions (NOC) compared to that of the new state on the right, is apparent.

Table 7-3. Summary of data and model parameters used in the LVPP

Parameter	Description	Values
n	Number of samples in original data set	10000
c	Number of variables in original and new data set	1
b	Length of time series segment	250
N	Number of time series segments	40
F	Number of features	17
k	Number of PCs retained in model	2
n_{new}	Number of samples in new data set	20000
N_{new}	Number of new time series segments	80

Process monitoring with two principal components is shown in Figure 7-5. Collectively, the two components could explain approximately 66.7% of the variance in the features. The 95% confidence limit generated with a 2-GMM (magenta line) is shown enclosing the NOC data (red circles). As can be expected, some of the samples from the transition state ('blue asterisk') are inside the control limit, but the new state ('black stars') are completely flagged as a new or faulty condition.

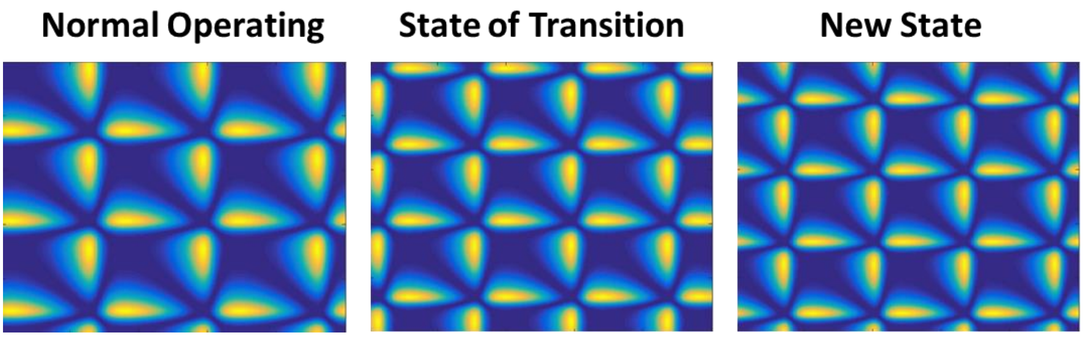


Figure 7-4. Exemplary plots of distance matrices of the three conditional states of the Lotka-Volterra predator-prey model

The corresponding T^2 and Q diagnostics with 95% control limits are shown in Figure 7-6. Both diagnostics could easily discern the new state ('black stars'). Based on the nominal values of the MAR and AUC in *Table 7-6,* the proposed methods performed very similarly to the RQA, SSA and RF methods and better than the NLPCA and 1-SVM approaches. To place the results into context, the single (upper) limit of ROC curves were considered here for the proposed method, SSA, RF, NLPCA and 1-SVM. The RQA, on the other hand, is monitored using both upper and lower limits but was simplified to give a one-dimensional curve to allow comparison to other approaches. The simplification entails enforcing the limits to be symmetrical around the mean of the diagnostic statistic distribution. The AUC was used to quantify the ROC curves which interpreted as a probabilistic value when a classifier ranks a randomly chosen true alarm above a randomly chosen false alarm.

Table 7-6 gives a summary of the performance of the proposed method in terms of the FAR, MAR, AUC, along with other dynamic process monitoring methods.

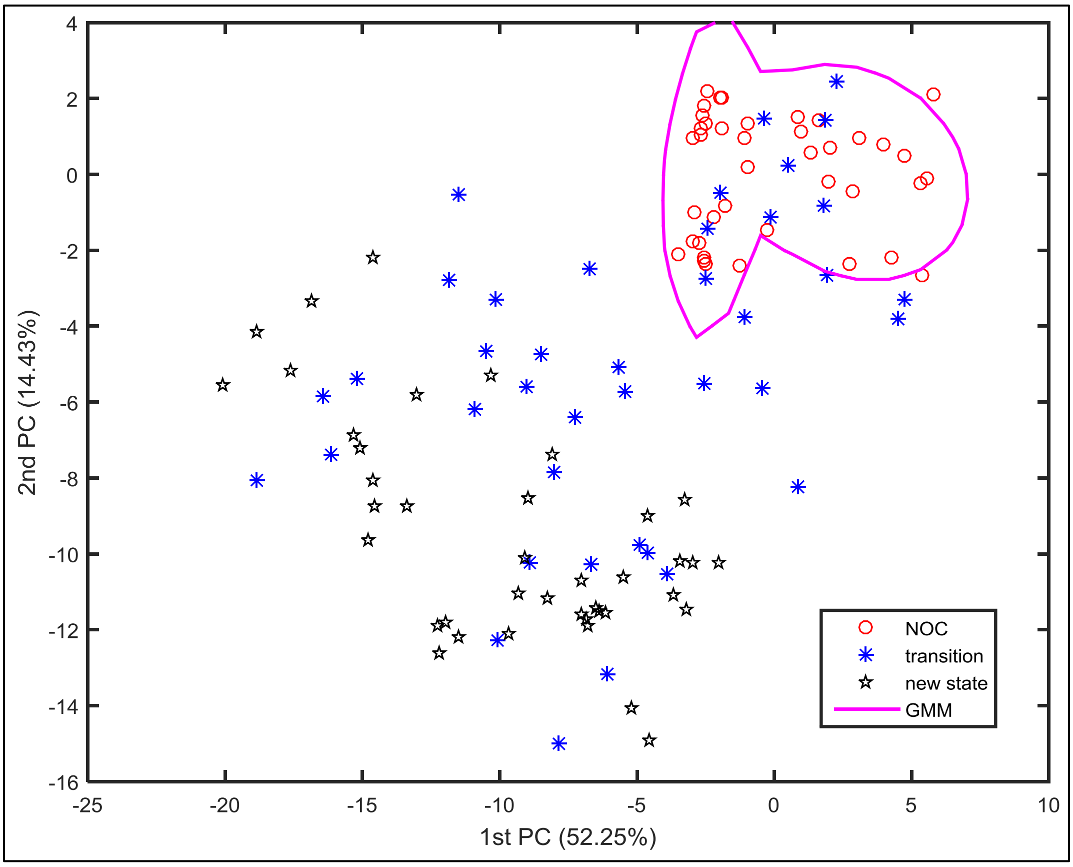


Figure 7-5. Plots of the first two principal components scores of the extracted features (CPV=66.68%) along with the trained 1-class GMM decision boundary (K=2), and its decision criterion plot.

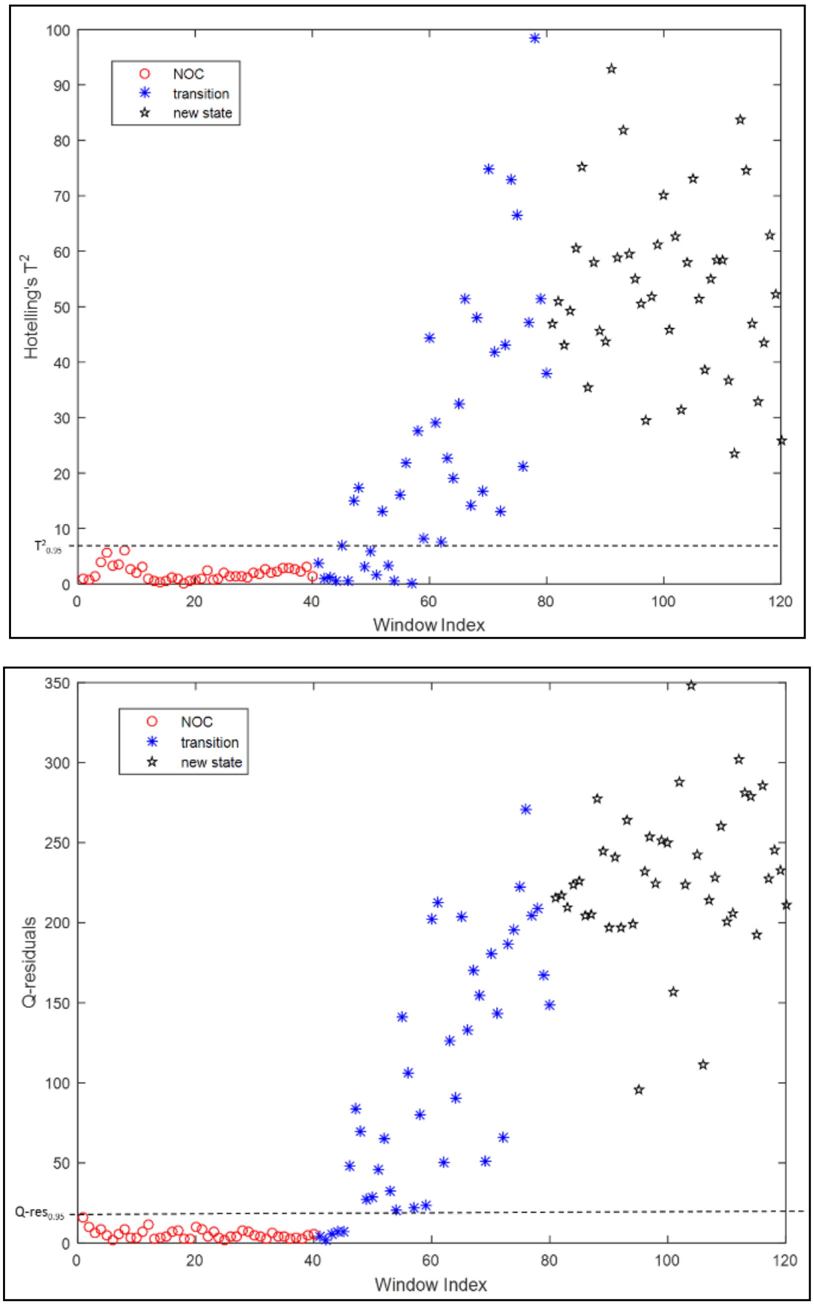


Figure 7-6. The Hotelling's T² (top) and Q-res (bottom) plots of extracted features.

7.3.2 Comparison to other approaches

The proposed method is compared with five other methods for dynamic process monitoring, as described in detail in the work of Aldrich and Auret (2013). These are the nonlinear principal component analysis (NLPCA), one-class support vector machine (1-SVM), recurrence quantification analysis (RQA), singular spectrum analysis (SSA), and random forests (RF). In the first three approaches (NLPCA, 1-SVM, and RQA), the time series data were embedded with a lag time *(k)* of 5 and embedding dimension *(m)* of 2. The lag time and embedding dimension parameters were determined with an average mutual information and false nearest neighbour approach, respectively. However, with the SSA and RF methods, the time series was embedded with a unity lag ($\boldsymbol{\tau}$ = 1) and an embedding dimension of 500, prior to reduction of the embedding dimension through principal component analysis and multidimensional scaling.

Singular Spectrum Analysis

SSA uses data-adaptive basis functions based on the information in the spectral coefficients of the data for decomposition of the time series to multiple components (Krishnannair et al., 2016). In SSA, the phase space characterisation model is defined by embedding parameters, reconstruction distances and eigenvectors. The reconstruction distance from the lag-trajectory vector is then used as a diagnostic statistic for SSA dynamic monitoring. As described elsewhere, SSA uses embedding parameters which subsequently used as a feature extractive step that automatically determines the relevant weights of lag variables for optimal phase space characterisation. SSA works by getting the singular value decomposition of the lag-trajectory matrix. The threshold for diagnostic statistic, $\hat{e}$ is calculated from the validation data by getting the quotient between the reconstruction distance e of test and NOC data as shown in eqn (7-9).

$$\hat{e} = \frac{e^{(test)}/m}{e^{NOC}} \tag{7-9}$$

Random Forest Model

The random forest (RF) model is considered as a nonlinear counterpart of SSA. The RF features are extracted by applying classical multidimensional scaling to the embedded data. Similar to SSA, the RF phase space characterisation model is also defined through scaling, embedding, and reconstruction of the data. The method also requires mapping and demapping models using the random forest with setting parameters defined in Table 7-4.

Table 7-4. Setting parameters used in Random forest model

Parameters	Value/Method
Number of randomly selected variables	N/3
Minimum leaf node size	5
Number of trees	100

Like SSA, the RF monitoring is also facilitated by the average and scaled reconstruction distance as a diagnostic statistic (Aldrich & Auret, 2013). In this case, all random forest models are carried out using the randomForest package (Liaw & Wiener, 2002) in the R statistical environment (Team, 2010).

Non-linear Principal Component Analysis

The nonlinear principal component analysis (NLPCA) is a nonlinear feature extraction approach based on the use of an autoassociative neural network (AANN). In particular, the circular inverse NLPCA was considered, wherein the characterisation of periodic behaviour in an input space is allowed in circular nodes in neural network (Scholz, 2007). In dynamic process monitoring, the use of circular INLPCA is used to extract the closed-curve one-dimensional manifold in the phase space which eventually represent a dense attractor (Aldrich & Auret, 2013).

The training of the network of circular inverse NLPCA is carried out according to the setting parameters described in Table 7-5. The trained networks denoted both the mapping and demapping models of this method.

Table 7-5. Setting parameters used in circular inverse NLPCA

Setting	Value/method
Number of hidden layers	1
Number of nodes in hidden layer	6
Activation function of hidden layer nodes	Hyperbolic tangent
Pre-scaling	To give maximum standard deviation of 0.1
Weight decay for regularization	Weight decay coefficient of 0.001

Similar to SSA and RF, the circular inverse NLPCA phase space characterisation model can be carried out to test data through scaling, embedding, and reconstruction utilizing the nonlinear PCA toolbox for MATLAB (Scholz, 2011). Moreover, the diagnostic statistic is also determined using eqn (7-9) from the validation data.

One class – Support Vector Machine

The One class-support vector machine (1-SVM) model characterises the phase space by data support estimation approach. In this approach, the versatility of kernel functions and the rigorous statistical learning framework of support vectors are combined. To apply this model to dynamic monitoring, the NOC data are used to train the model in order to determine the support of the phase space using the optimised kernel width σ. From there, the new data in the phase space could then be compared with the 1-SVM NOC support (Aldrich & Auret, 2013). Similar with the GMM in the proposed approach, new data that found outside the 1-SVM support of NOC could mean that a change has happened.

The kernel width, σ is searched by grid method with evenly-spaced grid with ten values obtained from the distribution of interpoint distances of lag-trajectory matrix (Belousov, Verzakov, & Von Frese, 2002) along with fivefold cross-validation The optimal σ is chosen through getting the largest value upon trading off between the minimized mean fraction of false alarms and mean fraction of support vectors below 0.1. For determining the diagnostic statistic threshold, an approach similar to other methods using the feature extraction and reconstruction method is used. In this case, the diagnostic statistic threshold was set as the ((1 – α) x 100)-th percentile of validation statistic distribution, where α is the design selection for expected false alarm rates.

Recurrence Quantification Analysis

As previously discussed in Chapter 3, the RQA is a phase space characterisation approach that focuses on the feature space for diagnostic purpose. It works by determining the recurrence or the repeated occurrences of points in the same neighbourhood. The recurrence can be quantified by calculating the recurrence rate which could also be used as a diagnostic statistic of this approach. The trend of recurrence rate is a useful indicator of the dynamics of the data (Aldrich & Auret, 2013). This method requires the definition of the neighbourhood size threshold ε to determine the recurrence matrices (Schinkel et al., 2008). In this instance, this parameter is estimated by obtaining the mean of the interpoint distances of lag-trajectory matrix. As a result, the recurrence rate of the lag-trajectory matrix using the obtained neighbourhood size ε is then considered the diagnostic statistic. However, since both the decrease and increase of the recurrence rate could indicate a change, the upper and lower limits of RQA diagnostics were first defined. Using a user-defined expected false alarm rate, α, the upper and lower limits were defined as the ((1 – α/2) x 100)-th percentile and ((α/2) x 100)-th percentile of the diagnostic statistic distribution, respectively.

7.3.3 Discussions

Based on the nominal values of the MAR and AUC in *Table 7-6*, the proposed methods performed very similarly to the RQA, SSA and RF methods and better than the NLPCA and 1-SVM approaches. To place the results into context, the single (upper) limit of ROC curves were considered here for the proposed method, SSA, RF, NLPCA and 1-SVM. The RQA, on the other hand, is monitored using both upper and lower limits but was simplified to give a one-dimensional curve to allow comparison to other approaches. The simplification entails enforcing the limits to be symmetrical around the mean of the diagnostic statistic distribution. The AUC was used to quantify the ROC curves which interpreted as a probabilistic value when a classifier ranks a randomly chosen true alarm above a randomly chosen false alarm.

Table 7-6. Summary of the performance metrics of the two fault detection methods on the Lotka-Volterra predator-prey model time series.

Performance Metrics		Proposed Method	NLPCA	1-SVM	RQA	SSA	RF
Alarm Rates	**FAR (%)**	5%	1%	3%	35%	1%	4%
	MAR (%)	11.3%	91%	24%	4%	4%	2%
	RL	1000	61	33	43	273	206
	AUC	0.96	0.63	0.84	0.97	0.98	0.99

In terms of the RL, it compared less favourably with the other methods. Decreasing the window size from $b = 250$ to $b = 100$, yielded interesting results, as indicated in Figure 7-7. In this figure, it can be seen that the onset of the change or transition state is rapidly detected on the Q-index plot in particular.

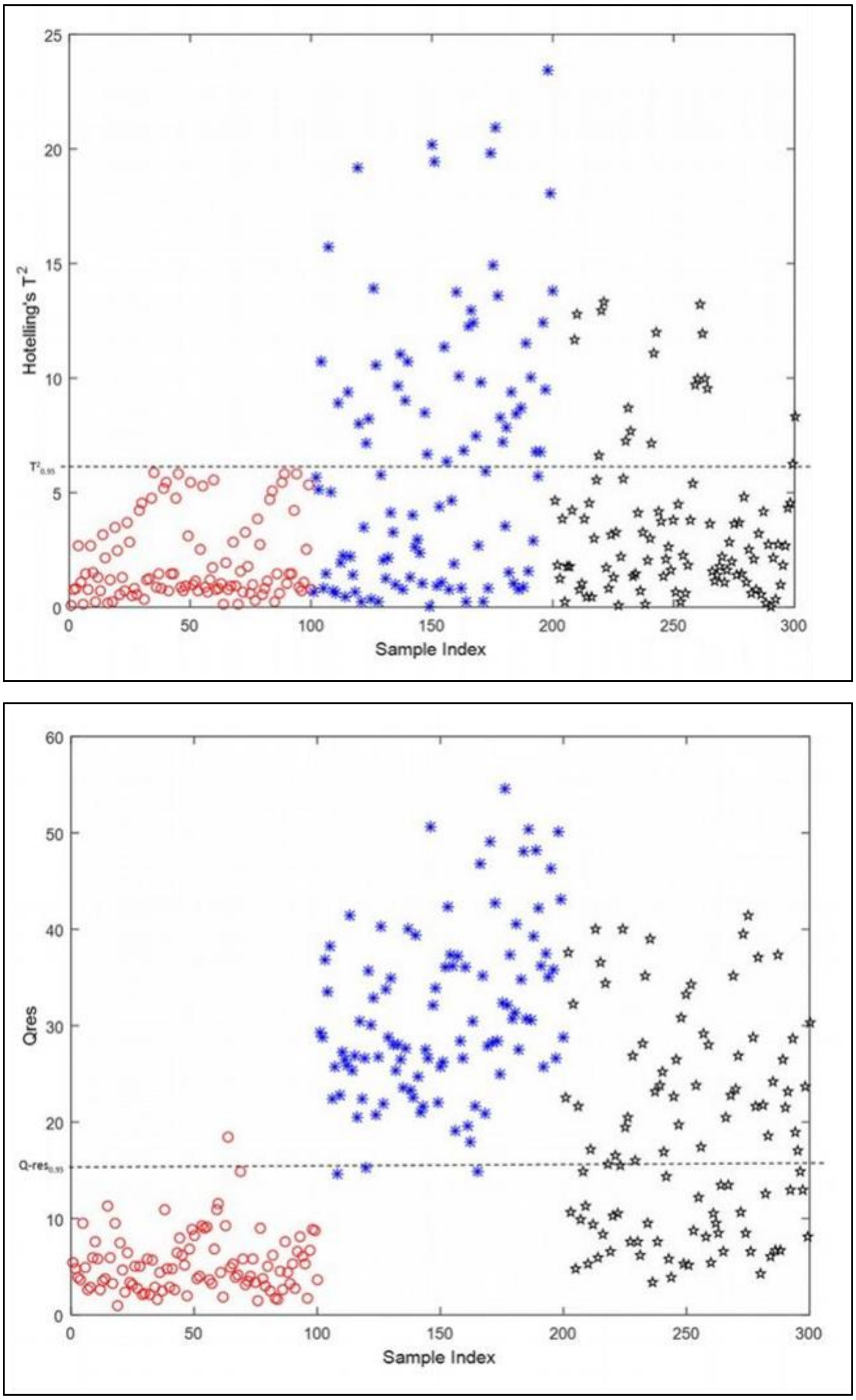


Figure 7-7. The Hotelling's T^2 (top) and Q-index plot (bottom) of extracted features with a window width of $b = 100$.

Interestingly, once the new state is reached, it becomes somewhat less distinguishable. With a shorter window, the method performs as well as any of the other methods in Table 7-6, if not better. Rigorous comparison was not further pursued, since it should be treated as a multi-objective optimisation problem, which was considered beyond the scope of this proof-of-concept study.

7.4 Case Study 2: Tennessee Eastman Problem

The Tennessee Eastman (TE) process, as described in Chapter 5, has been used as a benchmark in numerous process control and monitoring studies. This process is based on an actual chemical process, as initially proposed by Downs and Vogel (1993).

The TE data sets considered here are similar to those used by Aldrich and Auret (2013). A total of 22 data sets were used, i.e. one data associated with NOC and the remaining 21 associated with different faults conditions. The TE data sets comprised 52 variables - 22 of which are continuous process measurements, 19 of which are composition measurements and the remaining 11, which are manipulated variables. Descriptions of the variables are presented in Table 7-7. The simulation is sampled at 3 min intervals. Each simulated data set (NOC and 21 faults) consists of 960 samples.

These NOC samples were used to construct the off-line process monitoring model. Moreover, an additional 500 samples were generated to serve as a validation set in the study. For the 21 faults, simulations were carried out by introducing the fault conditions after 160 samples, thus giving 800 samples related to fault conditions in each data set. The list of faults, as well as their descriptions and associated variables are given in Table 7-8.

Table 7-7. Description of variables in Tennessee Eastman process.

Process Measurement	Compositional Measurement	Manipulative Variable
Variable No. - Description	*Variable No. - Description*	*Variable No. - Description*
1 - A Feed	23 - Reactor feed component A	42 - D feed flow
2 - D Feed	24 - Reactor feed component B	43 - E feed flow
3 - E Feed	25 - Reactor feed component C	44 - A feed flow
4 - Total Feed	26 - Reactor feed component D	45 - Total feed flow
5 - Recycle flow	27 - Reactor feed component E	46 - Compressor recycle valve
6 - Reactor feed rate	28 - Reactor feed component F	47 - Purge valve
7 - Reactor pressure	29 - Purge component A	48 - Separator product liquid flow
8 - Reactor level	30 - Purge component B	49 - Stripper product liquid flow
9 - Reactor temperature	31 - Purge component C	50 - Stripper steam valve
10 - Purge rate	32 - Purge component D	51 - Reactor cooling water flow
11 - Separator temperature	33 - Purge component E	52 - Condenser cooling water flow
12 - Separator level	34 - Purge component F	
13 - Separator pressure	35 - Purge component G	
14 - Separator underflow	36 - Purge component H	
15 - Stripper level	37 - Product component D	
16 - Stripper pressure	38 - Product component E	
17 - Stripper underflow	39 - Product component F	
18 - Stripper temperature	40 - Product component G	
19 - Stripper steam flow	41 - Product component H	
20 - Compressor work		
21 - Reactor cooling water outlet temperature		
22 - Separator cooling water outlet temperature		

Table 7-8. Description of faults in the Tennessee Eastman process.

Fault No.	Description	Type	Variable Directly Involved	Additional Variables Affected
1	Stripper feed A/C feed ratio, B composition constant	Step change	1, 6, 23, 43, 45	Most
2	Stripper feed B composition, A/C feed ratio constant	Step change	47	Most
3	D feed temperature	Step change	-	-
4	Reactor cooling water inlet temperature	Step change	9, 21, 51	None
5	Condenser cooling water inlet temperature	Step change	11, 22, 52	Most
6	Reactor feed A loss	Step change	44	Most
7	Stripper feed C header pressure loss, reduced availability	Step change	-	Most
8	A, B, and C stripper feed composition	Random variation	-	Most
9	Reactor feed D temperature	Random variation	-	-
10	Stripper feed C temperature	Random variation	-	-
11	Reactor cooling water inlet temperature	Random variation	9, 21, 51	None
12	Condenser cooling water inlet temperature	Random variation	22	-
13	Reaction kinetics	Slow drift	-	-
14	Reactor cooling water valve	Sticking	51	Most
15	Condenser cooling water valve	Sticking	-	-
16-20	Unknown		-	-
21	Stripper feed valve fixed at steady-state position	Constant position	45	-

7.4.1 Results

Table 7-9. Summary of data and model parameters used in the Tennessee Eastman (TE) process case study.

Parameter	Description	Values
n	Number of samples in original data set	960
c	Number of variables in original and new data set	52
b	Length of time series segment	80
N	Number of time series segments	12
F	Number of features	17
k	Number of PCs retained in model	2
n_{new}	Number of samples in new data set	20,160
N_{new}	Number of new time series segments	252

A window width b = 80 was used for all the NOC and faulty datasets of the TE process. Examples of images of the distance matrices of the different data sets are shown in Figure 7-8. The NOC distance matrix appears similar to the distance matrices of faults 1, 2, 3, 4, 9, 15, 19 and 21, suggesting that these faults may be more difficult to detect than faults 11, 12, 14 and 17, for example, that appear distinctly different from the NOC matrix.

As before, process monitoring was done with two principal components, since collectively, the first two components could explain approximately 96.9% of the variance in the features derived from the NOC data. The results for the T^2 and Q-statistics, as well as principal component score plots are shown graphically in Appendices A – C. In the score plots, the 95% confidence limit generated with a 2-GMM is again shown enclosing the NOC data (red '*'), while most of the faults (blue '+' markers) can be seen lying outside the GMM decision boundary.

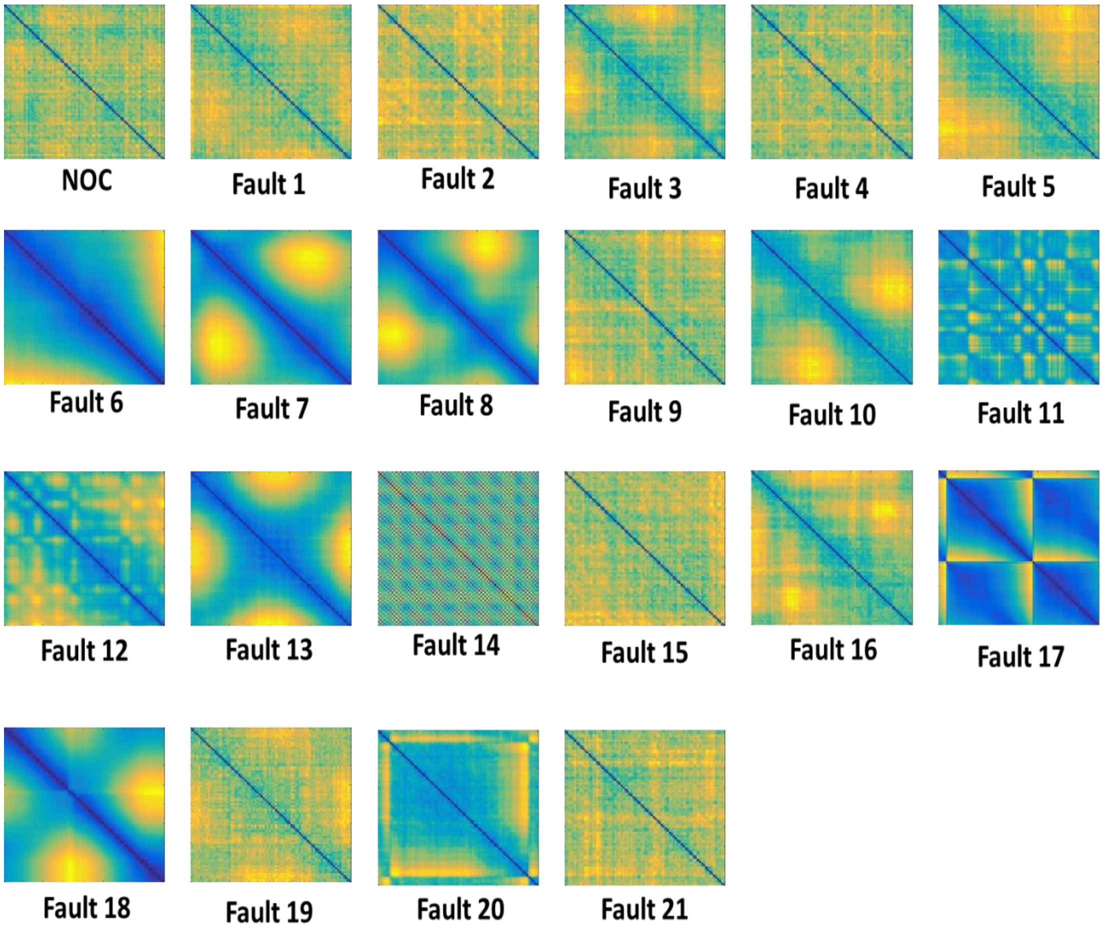


Figure 7-8. Representative plots of distance matrices (80-by-80 matrix) of the NOC and faults datasets of TE problem

Table 7-10. Results of the performance of the process monitoring method to the faults of TE process using the Hotelling's T², Q-residuals, and GMM monitoring statistics.

FAULT No.	***FAULT DETECTION RATES***									
	Rato and Reis (2013)						*Proposed Method*			
	PCA		*DPCA*		*DPCA-DR*					
	T^2	*Q*	*T^2*	*Q*	*T^2*	*Q*	*T^2*	*Q*	GMM	AUC
1	0.991	0.995	0.990	0.994	0.996	**0.998**	0.5	0.4	0.8	0.85
2	**0.985**	0.984	0.984	0.981	**0.985**	0.983	1	0.8	1	0.957
3	**0.036**	0.006	0.035	0.010	0.021	0.016	0.9	1	1	0.914
4	0.218	0.980	0.165	**0.999**	0.998	**0.999**	1	1	1	0.943
5	0.257	0.217	0.293	0.228	**0.999**	**0.999**	0.8	1	1	0.943
6	0.989	**0.999**	0.989	**0.999**	**0.999**	**0.999**	1	0.5	1	1

7	**0.999**	**0.999**	0.986	**0.999**	**0.999**	**0.999**	1	0.9	1	1
8	0.974	0.968	0.973	0.974	**0.985**	0.981	1	0.5	1	1
9	**0.034**	0.010	0.030	0.002	0.020	0.010	0.9	1	1	0.936
10	0.367	0.154	0.439	0.172	**0.956**	0.933	0.9	0.7	1	0.993
11	0.414	0.638	0.340	0.829	**0.965**	0.865	0.7	0.7	0.9	0.915
12	0.985	0.925	0.990	0.964	**0.998**	**0.998**	1	1	1	1
13	0.943	0.950	0.943	0.950	**0.958**	0.956	1	0.9	1	1
14	0.988	**0.999**	0.990	**0.999**	0.998	**0.999**	1	1	1	1
15	0.035	0.007	0.059	0.009	**0.385**	0.047	1	0.9	1	0.986
16	0.174	0.137	0.217	0.145	**0.976**	0.945	1	0.9	1	0.977
17	0.787	0.905	0.790	0.953	**0.976**	0.975	1	0.7	1	1
18	0.893	0.901	0.890	0.898	**0.905**	0.900	1	0.9	1	0.971
19	0.115	0.059	0.046	0.298	**0.971**	0.843	0.9	1	1	0.957
20	0.340	0.423	0.408	0.493	0.908	**0.916**	0.7	0.7	1	0.921
21	0.362	0.414	0.429	0.409	0.539	**0.577**	1	1	1	1

7.4.2 Comparison to other approaches

Table 7-10 summarizes the results in Appendices 1-3 and also compare them to results reported by Rato and Reis (2013). In their study, the use of principal components (PCA), dynamic principal components (DPCA), as well as an enhanced version of DPCA (DPCA-DR) were considered. In the DPCA-DR approach, autocorrelation in the T^2 and Q-statistics is accounted for, to improve fault detection.

In the comparison of the results in Table 7-10, it should be borne in mind that in the proposed approach, chunks of data were mapped, owing to the use of contiguous windows, while individual data points were mapped with the methods considered in the study of Rato and Reis (2013). For this reason, the detection rates of the proposed method are reported to an accuracy of one decimal only. On the whole, the proposed method performed as well as, if not better, than the DPCA-DR method. However, it is interesting to note that it performed considerably better on Faults 3, 9 and 15 that are generally considered to be difficult to detect (Yin et al., 2012).

7.5 Summary and Conclusions

In this chapter, the monitoring of dynamic process systems is proposed, based on Recurrence Texture Analysis, which has seen as a flexible method in drawing on a rich set of descriptors of the structure of the time series data. Only two such approaches, co-occurrence matrices and wavelets were considered in this study.

Conventional statistical process monitoring methods can subsequently be used to detect changes in the dynamic system. In the two case studies presented here, the potential of the RTA as a solid basis for dynamic process monitoring system was clearly demonstrated. It was seen that the Principal component models of RTA-extracted descriptors (i.e. GLCM and wavelet features) obtained from two simulated systems were able to reliably detect most of the fault conditions in two simulated case studies. Moreover, better results were obtained when comparing the approach with other methods proposed for dynamic process monitoring.

Although this study provided some respectable results, some recommendations for future works are still proposed. These include the optimisation of the approach in terms of different (non-Euclidean) distance matrices, different approaches to extract textural features from these matrices, such as textons, steerable pyramids, local binary patterns, fractal descriptors, as well as variants of these and other approaches.

8. CONCLUSION AND RECOMMENDATIONS

8.1 Summary

This thesis presents a principled approach to analysing the structures of time series, as represented by distance or proximity matrices, by use of textural extraction algorithms that are commonly used on image data. It can be seen as an extension or generalisation of recurrence quantification analysis, which is a powerful framework for nonlinear time series analysis proposed in the 1980s. More specifically, the method proposed in this thesis, referred to as "Recurrence Texture Analysis" uses textural analysis to extract features from distance matrices. Six algorithms are considered here: based on the use of grey level co-occurrence matrices, wavelets, local binary patterns, textons and two pretrained convolutional neural networks, namely AlexNet and VGG16. The use of the features extracted from overlapping or non-overlapping segments of the time series, or possibly the entire time series depends on the analysis. This could be time series classification, cluster analysis, dynamic process monitoring or even the development of predictive models.

Over the course of this thesis, RTA showed robust applicability in capturing the structures of data that are relevant in the holistic understanding of dynamical behaviour of the system. It has showed substantial advantage in the characterisation of the complex and dynamic properties of simulated and actual process system data over its counterparts including RQA. In fact, the state-of-the-art deep learning algorithms namely AlexNet and VGG16, along with RTA-textons were seen superior over RQA and highly competitive against other published methods. Additionally, this thesis also demonstrated the reliability of the method in dealing with other applications such as time series analysis and dynamic process monitoring

As a summary, the thesis primarily focused on the evaluation of the method on capturing the dynamic behaviour of the system. Chapter 2 critically reviewed the related literature on the concept of recurrence along with time series analysis using the recurrence plots. In this task, it was found out that only limited researches have considered the use of textural analysis in extracting textural features in the recurrence plots, specifically in the distance matrices, and no research has been made on its application to minerals processing. Moreover, as far as the literature review is concerned, the use of distance matrices alone in analysing the dynamics has not been considered mainly due to the absence of robust method that could quantify the recurrence patterns contained into these plots. In addition, the application of deep neural network or deep learning has not been considered. These research gaps were bridged in Chapter 3 by presenting the RTA as an alternative approach to address these problems. The major steps involved in RTA are also described in detail.

Preliminary study pertaining to the influence of windowing parameters and distance metrics were presented in Chapter 4. Additionally, the method was also applied in the recurrence plots and the results were compared to RQA. In Chapter 5, the method was applied to deal with time series analysis, particularly in time series classification. RTA was employed to represent or encode some of the public time series datasets into RTA features which subsequently used as predictors in the classification. Moreover, the optimisation of the hyperparameters of the RTA algorithms was carried out in this chapter.

In chapter 6, the method was applied to a number of real process datasets in grinding circuit and solids processing for the primary goal of characterising the dynamic behaviour of the data and subsequently providing robust framework model for soft sensor online application. The analysis was carried out by looking the cluster behaviour of the data and further quantifying this by classification. Lastly, the method was used as a strong base in the development of the principal component model-based dynamic process monitoring system. This was clearly presented in Chapter 7.

8.2 Conclusion

With the main objective of evaluating the applicability and robustness of the Recurrence Texture Analysis in characterising the dynamic process systems, that is, to extract the textural features contained in the distance matrices via texture analysis, several tasks were then accomplished. The conclusions are fully outlined below.

- Within the limited scope of the study, it was found that the type of distance metrics did not significantly affect the performance of the method. On the contrary, the windowing parameters, especially the window width *b*, is an important parameter that needs to be selected appropriately as the influence of this parameter is quite significant to the overall outcome of the problem being addressed. In general, the choice of *b* generally depends on the applications of RTA being considered and the kind of structural changes of the time series the researchers are being investigated. In this sense, the autocorrelation function could be used as a guide in estimating this parameter. The acceptable rule is that the minimum value of lagged length where ACF is at the minimum should be used as the minimum value of *b* in segmentation of the time series.
- Throughout the evaluation of the method, it was found out that the RTA-VGG16 consistently achieved highest performance over other considered RTA feature extraction algorithms. Furthermore, it also attained highly competitive results compared to other similar

methods considered in this thesis. To some extent, this can be expected as VGG16 and AlexNet are supervised approaches to functions, as opposed to the other algorithms that are unsupervised.

- For other RTA features, it was demonstrated that the RTA-GLCM, RTA-wavelet, and RTA-LBP also showed satisfactory classification performance, which was reliably on par with other similar methods that deal with time series analysis. Moreover, the RTA-texton and RTA-AlexNet also reliably displayed outstanding classification, outperforming RQA and other similar methods.
- The varying levels of success on the application of RTA to three actual solid processing time series data clearly showed that the RTA is a reliable method to characterise the dynamic behaviour of dynamic process system. Moreover, specific conclusions were made in these studies:
    - It was showed that the mill load could potentially be used as a sole parameter for monitoring the conditions of autogenous mill. Further, it was discerned that the RTA has a huge opportunity for possible application in process control and monitoring. The proposed method successfully classified the mill states using the mill load data.
    - It was also revealed that the RTA features from the combined power draw and temperature of the mill, with possible incorporation of other mill parameters, could provide reliable information in estimating the particle sizes of the feed. Moreover, it was seen that the RTA reliably performed better over traditional approaches such as lagged trajectory matrix. More importantly, the method showed a lot of potential for advanced development of real-time monitoring and estimation of particle sizes.

  - The RTA was found reliable in capturing the powder flow behaviour. It was also concluded that the use of the time series of the powder mass could potentially be used as monitoring parameter and subsequent measurement of the powder flow. Interestingly, the RTA features performed better than RQA.

- The RTA showed a reliable method in drawing on a rich set of descriptors of the structure of the time series data in the application for dynamic process monitoring system. Conversely, the method proved its versatility by demonstrating that the RTA features could be used as a strong foundation in the development of dynamic process monitoring scheme. From a couple of case studies, it was seen that the principal component models of RTA features were able to reliably detect most of the faults conditions. Moreover, better results were obtained when comparing RTA with other methods proposed for dynamic process monitoring.
- It is also noted the VGG16 consistently performed significantly better than AlexNet over all considered datasets, which is in keeping with other comparative studies between these two algorithms. This observation is consistent with the premise that deeper neural networks can lead to better classification models in complex decision spaces.
- The use of pretrained CNNs (VGG16 and AlexNet) is found effective enough to capture the dynamic behaviour of time series encoded through distance matrices. This consequently confirmed that even though distance matrix plots are not part in the trained datasets, it can be extended to other applications such as in recurrence texture analysis through transfer learning.

8.3 Recommendations

In this thesis, it was demonstrated that Recurrence Texture Analysis is a reliable method in characterising the dynamic behaviour of dynamic process system. Moreover, it was also found out that the method is effective in capturing the structures of the time series data. With this, its utilisation to a number of applications can be extended for future study. Specifically, further works are:

- Partial retraining of VGG16 and AlexNet networks as part of parameter optimisation of these CNNs.
- Improvement of the performance of RTA by considering other textural features like fractal descriptors, and variety of these, e.g. use of dynamic feature extraction algorithms, steerable pyramids. Optimisation of the hyperparameters of the textural extraction algorithms considered in this thesis should also be taken into consideration.
- Clearly, further optimisation of the methodology needs to be considered with respect to guidelines for the selection of the optimal distance metric, window width and step size, depending on the case study.
- Development and subsequent implementation of a real-time measurement and monitoring scheme and using the fundamentals of RTA for possible application in minerals processing, particularly autogenous mill.
- Exploration and possible application of the RTA to other time series data mining tasks (e.g. segmentation, indexing, forecasting).
- In-depth comparative study of RTA with other methods of nonlinear time series analysis (e.g. Random Forest, surrogate data).

- Possible application of the method to several scientific fields that deal with dynamic systems and time series data (e.g. stock market analysis, ECG classification).

9. REFERENCES

Abbasi, S., & Tajeripour, F. (2017). Detection of brain tumor in 3D MRI images using local binary patterns and histogram orientation gradient. *Neurocomputing, 219*, 526-535. doi: http://doi.org/10.1016/j.neucom.2016.09.051

Acuña-González, N., García-Ochoa, E., & González-Sánchez, J. (2008). Assessment of the dynamics of corrosion fatigue crack initiation applying recurrence plots to the analysis of electrochemical noise data. *International Journal of Fatigue, 30*(7), 1211-1219. doi: https://doi.org/10.1016/j.ijfatigue.2007.09.002

Addo, P. M., Billio, M., & Guégan, D. (2013). Nonlinear dynamics and recurrence plots for detecting financial crisis. *The North American Journal of Economics and Finance, 26*, 416-435. doi: https://doi.org/10.1016/j.najef.2013.02.014

Afridi, M. J., Ross, A., & Shapiro, E. M. (2018). On automated source selection for transfer learning in convolutional neural networks. *Pattern Recognition, 73*(Supplement C), 65-75. doi: https://doi.org/10.1016/j.patcog.2017.07.019

Aguila-Camacho, N., Le Roux, J. D., Duarte-Mermoud, M. A., & Orchard, M. E. (2017). Control of a grinding mill circuit using fractional order controllers. *Journal of Process Control, 53*, 80-94. doi: https://doi.org/10.1016/j.jprocont.2017.02.012

Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on, 19*(6), 716-723. doi: 10.1109/TAC.1974.1100705

Aldrich, C., & Auret, L. (2013). *Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods* Advances in Computer Vision and Pattern Recognition, (pp. 374). doi:10.1007/978-1-4471-5185-2

Aldrich, C., Burchell, J. J., de V. Groenewald, J. W., & Yzelle, C. (2014). Visualization of the controller states of an autogenous mill from time series data. *Minerals Engineering, 56*, 1-9. doi: http://dx.doi.org/10.1016/j.mineng.2013.10.018

Aldrich, C., Marais, C., Shean, B. J., & Cilliers, J. J. (2010). Online monitoring and control of froth flotation systems with machine vision: A review. *International Journal of Mineral Processing, 96*(1–4), 1-13. doi: http://doi.org/10.1016/j.minpro.2010.04.005

Ambrose, R. P. K., Jan, S., & Siliveru, K. (2016). A review on flow characterization methods for cereal grain-based powders. *Journal of the Science of Food and Agriculture, 96*(2), 359-364. doi: 10.1002/jsfa.7305

Aradhye, H. B., Davis, J. F., & Bakshi, B. R. (2002). ART-2 and multiscale ART-2 for on-line process fault detection - Validation via industrial case studies and Monte Carlo simulation. *Annual Reviews in Control, 26*, 113-127. doi: 10.1016/S1367-5788(02)80020-7

Atay, F. M. (2010). *Complex Time-Delay Systems : Theory and Applications / edited by Fatihcan M. Atay*. Berlin, Heidelberg: Berlin, Heidelberg : Springer Berlin Heidelberg.

Auret, L., & Aldrich, C. (2010). Change point detection in time series data with random forests. *Control Engineering Practice, 18*(8), 990-1002. doi: http://dx.doi.org/10.1016/j.conengprac.2010.04.005

Babinec, P., Kucera, M., & Babincova, M. (2005). Global characterization of time series using fractal dimension of corresponding recurrence plots: From dynamical systems to heart physiology. *Harmonic and Fractal Image Analysis, 1*(1), 87-93.

Ballabio, D. (2015). A MATLAB toolbox for Principal Component Analysis and unsupervised exploration of data structure. *Chemometrics and Intelligent Laboratory Systems, 149, Part B*, 1-9. doi: http://dx.doi.org/10.1016/j.chemolab.2015.10.003

Barnard, J. P., Aldrich, C., & Gerber, M. (2001). Identification of dynamic process systems with surrogate data methods. *AIChE Journal, 47*(9), 2064-2075. doi: 10.1002/aic.690470917

Bela, J. (1981). Textons, the elements of texture perception, and their interactions. *Nature, 290*(5802), 91. doi: 10.1038/290091a0

Belousov, A. I., Verzakov, S. A., & Von Frese, J. (2002). Applicational aspects of support vector machines. *Journal of Chemometrics, 16*(8-10), 482-489. doi: 10.1002/cem.744

Bradley, E., & Mantilla, R. (2002). Recurrence plots and unstable periodic orbits. *Chaos: An Interdisciplinary Journal of Nonlinear Science, 12*(3), 596-600. doi: 10.1063/1.1488255

Caraiani, P., & Haven, E. (2013). The Role of Recurrence Plots in Characterizing the Output-Unemployment Relationship: An Analysis. *PLoS One, 8*(2). doi: 10.1371/journal.pone.0056767

Cazares-Ibáñez, E., Vázquez-Coutiño, G. A., & García-Ochoa, E. (2005). Application of recurrence plots as a new tool in the analysis of electrochemical oscillations of copper. *Journal of Electroanalytical Chemistry, 583*(1), 17-33. doi: 10.1016/j.jelechem.2005.03.046

Chaovalitwongse, W. A., Pottenger, R. S., Wang, S., Fan, Y. J., & Iasemidis, L. D. (2011). Pattern- and Network-Based Classification Techniques for Multichannel Medical Data Signals to Improve Brain Diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 41*(5), 977-988. doi: 10.1109/TSMCA.2011.2106118

Chen, W.-S. (2011). Use of recurrence plot and recurrence quantification analysis in Taiwan unemployment rate time series. *Physica A: Statistical Mechanics and its Applications, 390*(7), 1332-1342. doi: https://doi.org/10.1016/j.physa.2010.12.020

Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., & Batista, G. (2015). *The UCR Time Series Classification Archive.*

Clausi, D. A. (2002). An analysis of co-occurrence texture statistics as a function of grey level quantization. *Canadian Journal of Remote Sensing, 28*(1), 45-62. doi: 10.5589/m02-004

Cotta, C., & Schaefer, R. (2017). Bioinspired algorithms and complex systems *Journal of Computational Science* (Vol. 23, pp. 192-194).

Croisier, A. (1976). Perfect channel splitting by use of interpolation/decimation/tree decomposition techniques. *Proc. Int. Symp. on Info., Circuits and Systems, (Patras, Greece)*.

Deng, X., Tian, X., & Chen, S. (2013). Modified kernel principal component analysis based on local structure analysis and its application to nonlinear process fault diagnosis. *Chemometrics and Intelligent Laboratory Systems, 127*, 195-209. doi: 10.1016/j.chemolab.2013.07.001

Dong, D., & McAvoy, T. J. (1996). Nonlinear principal component analysis—Based on principal curves and neural networks. *Computers and Chemical Engineering, 20*(1), 65-78. doi: 10.1016/0098-1354(95)00003-K

Downs, J. J., & Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers & Chemical Engineering, 17*(3), 245-255. doi: http://dx.doi.org/10.1016/0098-1354(93)80018-I

Eckmann, J. P. (1987). Recurrence plots of dynamical systems. *EPL (Europhysics Letters), 4*(9), 973-977. doi: 10.1209/0295-5075/4/9/004

Eichkitz, C. G., Amtmann, J., & Schreilechner, M. G. (2013). Calculation of grey level co-occurrence matrix-based seismic attributes in three dimensions. *Computers & Geosciences, 60*, 176-183. doi: https://doi.org/10.1016/j.cageo.2013.07.006

Esling, P., & Agon, C. (2012). Time-series data mining. *ACM Comput. Surv., 45*(1), 1-34. doi: 10.1145/2379776.2379788

Farinella, G. M., Allegra, D., Moltisanti, M., Stanco, F., & Battiato, S. (2016). Retrieval and classification of food images. *Computers in Biology and*

Medicine, 77, 23-39. doi: https://doi.org/10.1016/j.compbiomed.2016.07.006

Fernández-Carrobles, M. M., Bueno, G., Déniz, O., Salido, J., García-Rojo, M., & Gonzández-López, L. (2015). Frequential versus spatial colour textons for breast TMA classification. *Computerized Medical Imaging and Graphics, 42*, 25-37. doi: https://doi.org/10.1016/j.compmedimag.2014.11.009

Ferreira, A., & Giraldi, G. (2017). Convolutional Neural Network approaches to granite tiles classification. *Expert Systems with Applications, 84*(Supplement C), 1-11. doi: https://doi.org/10.1016/j.eswa.2017.04.053

Fu, T.-c. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence, 24*(1), 164-181. doi: http://doi.org/10.1016/j.engappai.2010.09.007

Fu, T.-c., Law, C.-w., Chan, K.-k., Chung, F.-l., & Ng, C.-m. (2006). Stock Time Series Categorization and Clustering Via SB-Tree Optimization. In L. Wang, L. Jiao, G. Shi, X. Li, & J. Liu (Eds.), *Fuzzy Systems and Knowledge Discovery: Third International Conference, FSKD 2006, Xi'an, China, September 24-28, 2006. Proceedings* (pp. 1130-1139). Berlin, Heidelberg: Springer Berlin Heidelberg.

Fu, Y., & Aldrich, C. (2018). Froth image analysis by use of transfer learning and convolutional neural networks. *Minerals Engineering, 115*(Supplement C), 68-78. doi: https://doi.org/10.1016/j.mineng.2017.10.005

Fuerstenau, D. W., & Abouzeid, A. Z. M. (2002). The energy efficiency of ball milling in comminution. *International Journal of Mineral Processing, 67*(1), 161-185. doi: https://doi.org/10.1016/S0301-7516(02)00039-X

Gao, J., & Cai, H. (2000). On the structures and quantification of recurrence plots. *Physics Letters A, 270*(1–2), 75-87. doi: http://dx.doi.org/10.1016/S0375-9601(00)00304-2

Ge, L., & Ge, L.-J. (2016). Feature extraction of time series classification based on multi-method integration. *Optik - International Journal for Light and Electron Optics, 127*(23), 11070-11074. doi: http://doi.org/10.1016/j.ijleo.2016.08.089

Gopalakrishnan, K., Khaitan, S. K., Choudhary, A., & Agrawal, A. (2017). Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and Building Materials, 157*(Supplement C), 322-330. doi: https://doi.org/10.1016/j.conbuildmat.2017.09.110

Groenewald, J. W., & Aldrich, C. (2015). Root cause analysis of process fault conditions on an industrial concentrator circuit by use of causality maps and extreme learning machines. *Minerals Engineering, 74*, 30-40. doi: http://dx.doi.org/10.1016/j.mineng.2014.12.006

Groenewald, J. W., Coetzer, L. P., & Aldrich, C. (2006). Statistical monitoring of a grinding circuit: An industrial case study. *Minerals Engineering, 19*(11), 1138-1148. doi: 10.1016/j.mineng.2006.05.009

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., . . . Chen, T. (2017). Recent advances in convolutional neural networks. *Pattern Recognition*. doi: https://doi.org/10.1016/j.patcog.2017.10.013

Guangliang, Y., Xiang, L., Dawei, S., Xiaozhao, Z., Peng, Z., Yuexian, H., & Bin, H. (2016). Encoding physiological signals as images for affective state recognition using convolutional neural networks. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference, 2016*, 812. doi: 10.1109/EMBC.2016.7590825

Haralick, R. M. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE, 67*(5), 786-804. doi: 10.1109/PROC.1979.11328

Harkat, M. F., Mourot, G., & Ragot, J. (2003). Nonlinear PCA Combining Principal Curves and RBF-Networks for Process Monitoring (Vol. 2, pp. 1956-1961).

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.): Springer New York.

Hatami, N., Gavet, Y., & Debayle, J. (2017). Classification of Time-Series Images Using Deep Convolutional Neural Networks.

Holden, J. G., Riley, M. A., Gao, J., & Torre, K. (2013). Fractal analyses: Statistical and methodological innovations and best practices. *Frontiers in Physiology, 4*. doi: 10.3389/fphys.2013.00097

Hou, Y., Aldrich, C., Lepkova, K., Machuca, L. L., & Kinsella, B. (2016). Monitoring of carbon steel corrosion by use of electrochemical noise and recurrence quantification analysis. *Corrosion Science, 112*, 63-72. doi: http://dx.doi.org/10.1016/j.corsci.2016.07.009

Hsu, C.-C., Chen, L.-S., & Liu, C.-H. (2010). A process monitoring scheme based on independent component analysis and adjusted outliers. *International Journal of Production Research, 48*(6), 1727-1743. doi: 10.1080/00207540802552683

Hurvich, C. M., & Tsai, C.-L. (1989). Regression and Time Series Model Selection in Small Samples. *Biometrika, 76*(2), 297-307. doi: 10.2307/2336663

Iwanski, J. S., & Bradley, E. (1998). Recurrence plots of experimental data: To embed or not to embed? *Chaos: An Interdisciplinary Journal of Nonlinear Science, 8*(4), 861-871. doi: 10.1063/1.166372

Javorka, M., Turianikova, Z., Tonhajzerova, I., Javorka, K., & Baumert, M. (2009). The effect of orthostasis on recurrence quantification analysis of heart rate and blood pressure dynamics. *Physiological Measurement, 30*(1), 29-41. doi: 10.1088/0967-3334/30/1/003

Jemwa, G. T. (2003). *Multivariate nonlinear time series analysis of dynamic process systems.* (Master of Science in Engineering (Extractive Metallurgical Engineering)), University of Stellenbosch, South Africa.

Jemwa, G. T., & Aldrich, C. (2005). Monitoring of an industrial liquid–liquid extraction system with kernel-based methods. *Hydrometallurgy, 78*(1), 41-51. doi: 10.1016/j.hydromet.2005.03.003

Jemwa, G. T., & Aldrich, C. (2012). Estimating size fraction categories of coal particles on conveyor belts using image texture modeling methods. *Expert Systems with Applications, 39*(9), 7947-7960. doi: http://dx.doi.org/10.1016/j.eswa.2012.01.104

Jia, F., Martin, E. B., & Morris, A. J. (1998). Non-linear principal components analysis for process fault detection. *Computers and Chemical Engineering, 22*(1), S851-S854.

Jiang, Q., & Yan, X. (2014). Just-in-time reorganized PCA integrated with SVDD for chemical process monitoring. *AIChE Journal, 60*(3), 949-965. doi: 10.1002/aic.14335

Joe Qin, S. (2003). Statistical process monitoring: basics and beyond. *Journal of Chemometrics, 17*(8-9), 480-502. doi: 10.1002/cem.800

Johnsson, F., Zijerveld, R. C., Schouten, J. C., van Den Bleek, C. M., & Leckner, B. (2000). Characterization of fluidization regimes by time-series analysis of pressure fluctuations. *International Journal of Multiphase Flow, 26*(4), 663-715. doi: 10.1016/S0301-9322(99)00028-2

Jovic, A., & Jovic, F. (2017). Classification of cardiac arrhythmias based on alphabet entropy of heart rate variability time series. *Biomedical Signal Processing and Control, 31*, 217-230. doi: http://doi.org/10.1016/j.bspc.2016.08.010

Julesz, B. (1984). A brief outline of the texton theory of human vision. *Trends in Neurosciences, 7*(2), 41-45. doi: http://dx.doi.org/10.1016/S0166-2236(84)80275-1

Kam, K. M. (2000). *Simulation and implementation of nonlinear control systems for mineral processes / Kiew M. Kam.* Thesis (Ph.D.)--Curtin University of Technology.

Kano, M., Tanaka, S., Hasebe, S., Hashimoto, I., & Ohno, H. (2003). Monitoring independent components for fault detection. *AIChE Journal, 49*(4), 969-976. doi: 10.1002/aic.690490414

Kantz, H. (2004). *Nonlinear time series analysis / Holger Kantz and Thomas Schreiber* ([2nd ed.].. ed.). New York: New York : Cambridge University Press.

Keska, J. K., Smith, M. D., & Williams, B. E. (1999). Comparison study of a cluster of four dynamic flow pattern discrimination techniques for multi-phase flow. *Flow Measurement and Instrumentation, 10*(2), 65-77. doi: 10.1016/S0955-5986(98)00048-X

Kim, S., Choi, Y., & Lee, M. (2015). Deep learning with support vector data description. *Neurocomputing, 165*, 111-117. doi: http://dx.doi.org/10.1016/j.neucom.2014.09.086

Kistner, M. (2013). *Image texture analysis for inferential sensing in the process industries.* (Master in Engineering), Stellenbosch University, South Africa.

Kistner, M., Jemwa, G., & Aldrich, C. (2013). Monitoring of mineral processing systems by using textural image analysis. *Minerals Engineering, 52*, 169-177. doi: http://dx.doi.org/10.1016/j.mineng.2013.05.022

Kotyrba, M., Volna, E., & Bujok, P. (2015). Unconventional modelling of complex system via cellular automata and differential evolution. *Swarm and Evolutionary Computation, 25*, 52-62. doi: 10.1016/j.swevo.2015.07.005

Kovalev, V. A., Kruggel, F., Gertz, H. J., & Cramon, D. Y. v. (2001). Three-dimensional texture analysis of MRI brain datasets. *IEEE Transactions on Medical Imaging, 20*(5), 424-433. doi: 10.1109/42.925295

Krishnannair, S., Aldrich, C., & Jemwa, G. T. (2016). Detecting faults in process systems with singular spectrum analysis. *Chemical Engineering Research and Design, 113*, 151-168. doi: 10.1016/j.cherd.2016.07.014

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks.* Paper presented at the Advances in Neural Information Processing Systems.

Ku, W., Storer, R. H., & Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems, 30*(1), 179-196. doi: 10.1016/0169-7439(95)00076-3

Lee, J., & Chang, K. S. (1996). Applications of chaos and fractals in process systems engineering *J. Process Control* (Vol. 6, pp. 71-87).

Leung, T., & Malik, J. (2001). Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons. *International Journal of Computer Vision, 43*(1), 29-44. doi: 10.1023/A:1011126920638

Li, X., Ouyang, G., Yao, X., & Guan, X. (2004). Dynamical characteristics of pre-epileptic seizures in rats with recurrence quantification analysis. *Physics Letters A, 333*(1), 164-171. doi: 10.1016/j.physleta.2004.10.028

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News, 2(3),* 18-22.

Lindfield, G. (2012). *Numerical Methods : Using MATLAB* (3rd ed.. ed.). Burlington: Burlington : Elsevier Science.

Liu, L., Fieguth, P., Zhao, G., Pietikäinen, M., & Hu, D. (2016). Extended local binary patterns for face recognition. *Information Sciences, 358–359*, 56-72. doi: http://doi.org/10.1016/j.ins.2016.04.021

Liu, R., & Gillies, D. F. (2016). Overfitting in linear feature extraction for classification of high-dimensional image data. *Pattern Recognition, 53*, 73-86. doi: https://doi.org/10.1016/j.patcog.2015.11.015

Llop, M. F., Gascons, N., & Llauró, F. X. (2015). Recurrence plots to characterize gas–solid fluidization regimes. *International Journal of Multiphase Flow, 73*, 43-56. doi: 10.1016/j.ijmultiphaseflow.2015.03.003

Mandelbrot, B. (1967). How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension. *Science, 156*(3775), 636-638. doi: 10.1126/science.156.3775.636

March, T. K., Chapman, S. C., & Dendy, R. O. (2005). Recurrence plot statistics and the effect of embedding. *Physica D: Nonlinear Phenomena, 200*(1–2), 171-184. doi: http://dx.doi.org/10.1016/j.physd.2004.11.002

Marwan, Carmen, R., Thiel, M., & Kurths, J. (2007). Recurrence plots for the analysis of complex systems. *Physics Reports, 438*(5–6), 237-329. doi: http://dx.doi.org/10.1016/j.physrep.2006.11.001

Marwan, Kurths, J., & Foerster, S. (2015). Analysing spatially extended high-dimensional dynamics by recurrence plots. *Physics Letters A, 379*(10–11), 894-900. doi: http://dx.doi.org/10.1016/j.physleta.2015.01.013

Marwan, N. (2013). Cross recurrence plot toolbox for MATLAB (Version 5.18 (R29.3)). Retrieved from http://tocsy.pik-potsdam.de/CRPtoolbox/

Mehdipour Ghazi, M., Yanikoglu, B., & Aptoula, E. (2017). Plant identification using deep neural networks via optimization of transfer learning parameters. *Neurocomputing, 235*(Supplement C), 228-235. doi: https://doi.org/10.1016/j.neucom.2017.01.018

Mei, S., Zarrabi, N., Lees, M., & Sloot, P. M. A. (2015). Complex agent networks: An emerging approach for modeling complex systems. *Applied Soft Computing, 37*(C), 311-321. doi: 10.1016/j.asoc.2015.08.010

Mindlin, G. M., & Gilmore, R. (1992). Topological analysis and synthesis of chaotic time series. *Physica D: Nonlinear Phenomena, 58*(1), 229-242. doi: 10.1016/0167-2789(92)90111-Y

Moskvina, V., & Zhigljavsky, A. (2003). An Algorithm Based on Singular Spectrum Analysis for Change-Point Detection. *Communications in*

Statistics - Simulation and Computation, 32(2), 319-352. doi: 10.1081/SAC-120017494

Ojala, T., Pietikainen, M., & Harwood, D. (1994). Performance evaluation of texture measures with classification based on Kullback discrimination of distributions (Vol. 1, pp. 582-585).

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24*(7), 971-987. doi: 10.1109/TPAMI.2002.1017623

Ou, X., Pan, W., & Xiao, P. (2014). In vivo skin capacitive imaging analysis by using grey level co-occurrence matrix (GLCM). *International Journal of Pharmaceutics, 460*(1–2), 28-32. doi: http://dx.doi.org/10.1016/j.ijpharm.2013.10.024

Pain, C. C., Mansoorzadeh, S., Gomes, J. L. M., & de Oliveira, C. R. E. (2002). A numerical investigation of bubbling gas–solid fluidized bed dynamics in 2-D geometries. *Powder Technology, 128*(1), 56-77. doi: 10.1016/S0032-5910(02)00167-5

Rampun, A., Zheng, L., & Malcolm, P., Zwiggelaar, Reyer. (2015). *Classifying Benign and Malignant Tissues within the Prostate Peripheral Zone using Textons.* Paper presented at the Medical Image Understanding and Analysis (MIUA 2015).

Rana, B., Juneja, A., Saxena, M., Gudwani, S., Kumaran, S. S., Behari, M., & Agrawal, R. K. (2017). Relevant 3D local binary pattern based features from fused feature descriptor for differential diagnosis of Parkinson's disease using structural MRI. *Biomedical Signal Processing and Control, 34*, 134-143. doi: http://doi.org/10.1016/j.bspc.2017.01.007

Rato, T. J., & Reis, M. S. (2013). Fault detection in the Tennessee Eastman benchmark process using dynamic principal components analysis based on decorrelated residuals (DPCA-DR). *Chemometrics and*

Intelligent Laboratory Systems, 125, 101-108. doi: 10.1016/j.chemolab.2013.04.002

Rohde, G. K., Nichols, J. M., Dissinger, B. M., & Bucholtz, F. (2008). Stochastic analysis of recurrence plots with applications to the detection of deterministic signals. *Physica D: Nonlinear Phenomena, 237*(5), 619-629. doi: https://doi.org/10.1016/j.physd.2007.10.008

Ruiz de la Hermosa González-Carrato, R., García Márquez, F. P., Dimlaye, V., & Ruiz-Hernández, D. (2014). Pattern recognition by wavelet transforms using macro fibre composites transducers. *Mechanical Systems and Signal Processing, 48*(1–2), 339-350. doi: http://dx.doi.org/10.1016/j.ymssp.2014.04.002

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In E. R. David, L. M. James, & C. P. R. Group (Eds.), *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1* (pp. 318-362): MIT Press.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., . . . Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision, 115*(3), 211-252. doi: 10.1007/s11263-015-0816-y

Safo, S. E., & Ahn, J. (2016). General sparse multi-class linear discriminant analysis. *Computational Statistics & Data Analysis, 99*, 81-90. doi: http://dx.doi.org/10.1016/j.csda.2016.01.011

Savari, C., Kulah, G., Koksal, M., Sotudeh-Gharebagh, R., Zarghami, R., & Mostoufi, N. (2017). Monitoring of liquid sprayed conical spouted beds by recurrence plots. *Powder Technology, 316,* 148-156. doi: https://doi.org/10.1016/j.powtec.2016.12.075

Schinkel, S., Dimigen, O., & Marwan, N. (2008). Selection of recurrence threshold for signal detection. *The European Physical Journal Special Topics, 164*(1), 45-53. doi: 10.1140/epjst/e2008-00833-5

Schmid, C. (2001). Constructing models for content-based image retrieval (Vol. 2, pp. II39-II45).

Schmid, C. (2001, 2001). *Constructing models for content-based image retrieval.* Paper presented at the Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on.

Scholz, M. (2007). Analyzing periodic phenomena by circular PCABioinformatics Research & Development (pp. 38-47). Berline Heidelberg: Springer.

Scholz, M. (2011). Nonlinear PCA toolbox for MATLAB. Retrieved from http://www.nlpca.de/matlab.html

Schultz, A., Spiegel, S., Marwan, N., & Albayrak, S. (2015). Approximation of diagonal line based measures in recurrence quantification analysis. *Physics Letters A, 379*(14), 997-1011. doi: https://doi.org/10.1016/j.physleta.2015.01.033

Schultz, A., Zou, Y., Marwan, N., & Turvey, M. (2011). LOCAL MINIMA-BASED RECURRENCE PLOTS FOR CONTINUOUS DYNAMICAL SYSTEMS. *International Journal of Bifurcation and Chaos, 21*(04), 1065-1075. doi: 10.1142/S0218127411029045

Shao, Y. L. C. (2005). APPLICATION OF GREY RELATION ANALYSIS AND RBF NETWORK ON GRINDING-CONCENTRATION'S SOFT SENSING. *IFAC Proceedings Volumes, 38*(1), 74-79. doi: https://doi.org/10.3182/20050703-6-CZ-1902.01691

Shi, X., Lv, Y., Fei, Z., & Liang, J. (2013). A multivariable statistical process monitoring method based on multiscale analysis and principal curves. *International Journal of Innovative Computing, Information and Control, 9*(4), 1781-1800.

Silva, D. F., Souza, V. M. A. D., & Batista, G. E. A. P. A. (2013, 7-10 Dec. 2013). *Time Series Classification Using Compression Distance of Recurrence Plots.*

Paper presented at the 2013 IEEE 13th International Conference on Data Mining.

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556, 2014*.

Singha, M., Wu, B., & Zhang, M. (2017). Object-Based Paddy Rice Mapping Using HJ-1A/B Data and Temporal Features Extracted from Time Series MODIS NDVI Data. *Sensors (Basel, Switzerland), 17*(1), 10. doi: 10.3390/s17010010

Sipers, A., Borm, P., & Peeters, R. (2011). On the unique reconstruction of a signal from its unthresholded recurrence plot. *Physics Letters A, 375*(24), 2309-2321. doi: https://doi.org/10.1016/j.physleta.2011.04.040

Sipers, A., Borm, P., & Peeters, R. (2017). Robust reconstruction of a signal from its unthresholded recurrence plot subject to disturbances. *Physics Letters A, 381*(6), 604-615. doi: https://doi.org/10.1016/j.physleta.2016.12.028

Sivakumar, B. (2004). Chaos theory in geophysics: past, present and future. *Chaos, Solitons and Fractals: the interdisciplinary journal of Nonlinear Science, and Nonequilibrium and Complex Phenomena, 19*(2), 441-462. doi: 10.1016/S0960-0779(03)00055-9

Souza, V. M. A., Silva, D. F., & Batista, G. E. A. P. A. (2014, 24-28 Aug. 2014). *Extracting Texture Features for Time Series Classification.* Paper presented at the 2014 22nd International Conference on Pattern Recognition.

Spierings, A., Voegtlin, M., Bauer, T., & Wegener, K. (2016). Powder flowability characterisation methodology for powder-bed-based metal additive manufacturing. *Progress in Additive Manufacturing, 1*(1), 9-20.

Sprott, J. C. (2003). *Chaos and time-series analysis / Julien Clinton Sprott*. Oxford ; New York: Oxford ; New York : Oxford University Press.

Sudarshan, V. K., Acharya, U. R., Ng, E. Y. K., Tan, R. S., Chou, S. M., & Ghista, D. N. (2016). An integrated index for automated detection of infarcted myocardium from cross-sectional echocardiograms using texton-based features (Part 1). *Computers in Biology and Medicine, 71*, 231-240. doi: https://doi.org/10.1016/j.compbiomed.2016.01.028

Tang, J., Chai, T., Yu, W., & Zhao, L. (2012). Feature extraction and selection based on vibration spectrum with application to estimating the load parameters of ball mill in grinding process. *Control Engineering Practice, 20*(10), 991-1004. doi: https://doi.org/10.1016/j.conengprac.2012.03.020

Tatsumi, K., Yamashiki, Y., Canales Torres, M. A., & Taipe, C. L. R. (2015). Crop classification of upland fields using Random forest of time-series Landsat 7 ETM+ data. *Computers and Electronics in Agriculture, 115*, 171-179. doi: http://doi.org/10.1016/j.compag.2015.05.001

Tax, D. M. J. (2013). DDtools, the Data Description Toolbox for Matlab, 2013 (version 2.0.0).

Tax, D. M. J., & Duin, R. P. W. (2004). Support Vector Data Description. *Machine Learning, 54*(1), 45-66. doi: 10.1023/B:MACH.0000008084.60811.49

Team, R. D. C. (2010). R: A language and environment for statistical computing. Vienna: R Foundation for Statistical Computing. Retrieved from http://www.R-project.org

Terrill, P. I., Wilson, S. J., Suresh, S., Cooper, D. M., & Dakin, C. (2013). Characterising non-linear dynamics in nocturnal breathing patterns of healthy infants using recurrence quantification analysis. *Computers in Biology and Medicine, 43*(4), 231-239. doi: https://doi.org/10.1016/j.compbiomed.2013.01.005

Thiel, M., Romano, M. C., Kurths, J., Meucci, R., Allaria, E., & Arecchi, F. T. (2002). Influence of observational noise on the recurrence quantification analysis. *Physica D: Nonlinear Phenomena, 171*(3), 138-152. doi: 10.1016/S0167-2789(02)00586-9

Varma, M., & Zisserman, A. (2005). A statistical approach to texture classification from single images. *International Journal of Computer Vision, 62*, 61-81.

Vatsavai, R. R., Cheriyadat, A., & Gleason, S. (2010, 13-13 Dec. 2010). *Unsupervised Semantic Labeling Framework for Identification of Complex Facilities in High-Resolution Remote Sensing Images.* Paper presented at the 2010 IEEE International Conference on Data Mining Workshops.

Wang, L., & Shi, H. (2010). Multivariate statistical process monitoring using an improved independent component analysis. *Chemical Engineering Research and Design, 88*(4), 403-414. doi: 10.1016/j.cherd.2009.09.002

Wang, Z., & Oates, T. (2015). Imaging Time-Series to Improve Classification and Imputation.

Webber Jr, C. L., Marwan, N., Facchini, A., & Giuliani, A. (2009). Simpler methods do it better: Success of Recurrence Quantification Analysis as a general purpose data analysis tool. *Physics Letters A, 373*(41), 3753-3756. doi: http://dx.doi.org/10.1016/j.physleta.2009.08.052

Webber Jr, C. L., & Zbilut, J. P. (2003). Recurrence Quantification Analysis of Nonlinear Dynamical Systems *Recurrence Quantification Analysis*.

Yanhua, H., Carmona, J., & Murphy, R. F. (2006, 6-9 April 2006). *Application of temporal texture features to automated analysis of protein subcellular locations in time series fluorescence microscope images.* Paper presented at the 3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2006.

Yu, J. (2012). A nonlinear kernel Gaussian mixture model based inferential monitoring approach for fault detection and diagnosis of chemical processes. *Chemical Engineering Science, 68*(1), 506-519. doi: http://dx.doi.org/10.1016/j.ces.2011.10.011

Zbilut, J. P., & Webber, C. L. (1992). Embeddings and delays as derived from quantification of recurrence plots. *Physics Letters A, 171*(3), 199-203. doi: 10.1016/0375-9601(92)90426-M

Zbilut, J. P., Zaldivar-Comenges, J.-M., & Strozzi, F. (2002). Recurrence quantification based Liapunov exponents for monitoring divergence in experimental data. *Physics Letters A, 297*(3), 173-181. doi: 10.1016/S0375-9601(02)00436-X

Zeki Yalniz, I., & Aksoy, S. (2010). Unsupervised detection and localization of structural textures using projection profiles. *Pattern Recognition, 43*(10), 3324-3337. doi: https://doi.org/10.1016/j.patcog.2010.04.016

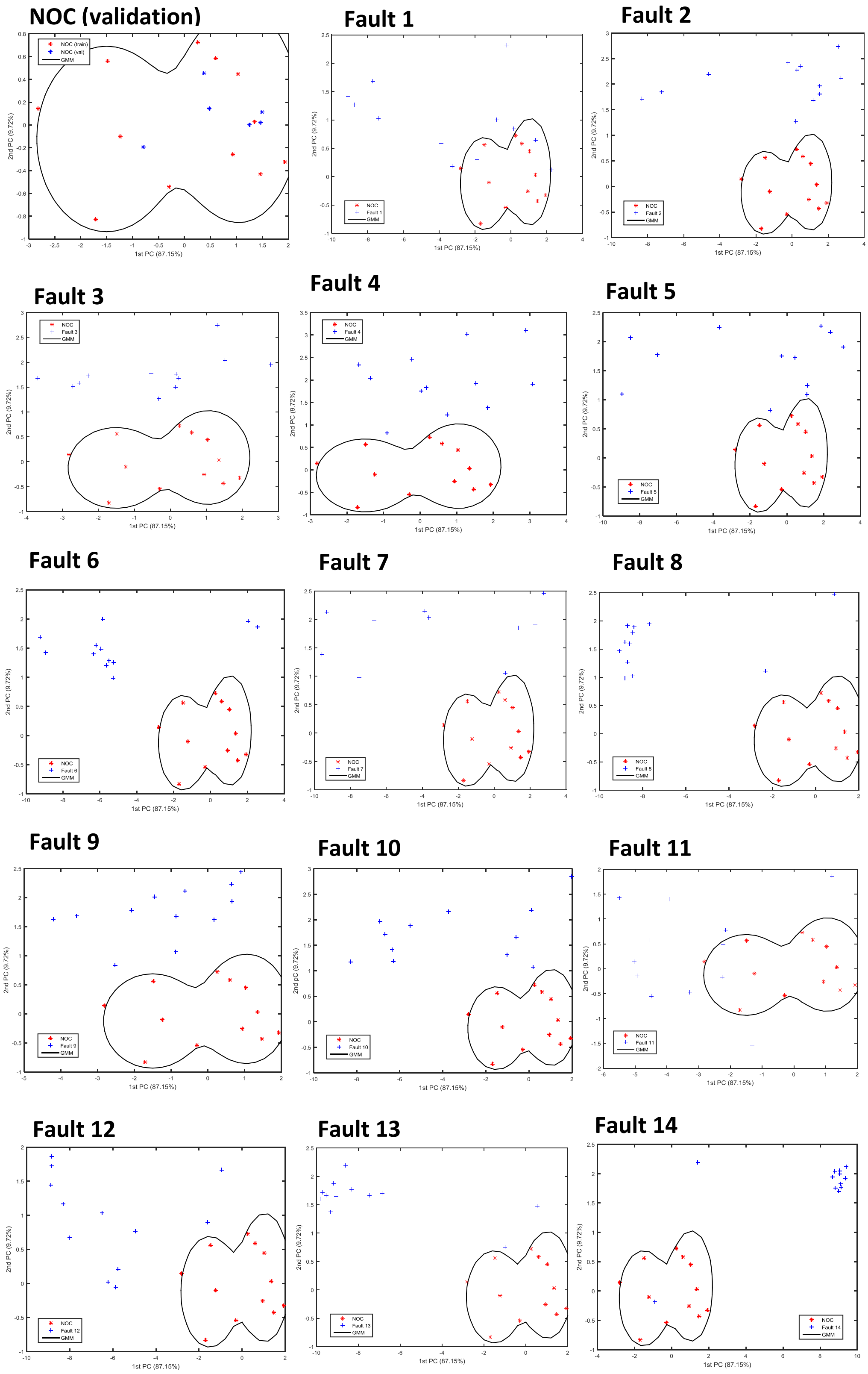
Zeng, Y., & Forssberg, E. (1994). Monitoring grinding parameters by vibration signal measurement - a primary application. *Minerals Engineering, 7*(4), 495-501. doi: https://doi.org/10.1016/0892-6875(94)90162-7

Zeng, Y., & Forssberg, K. S. E. (1996). Vibration signal emission from mono-size particle breakage. *International Journal of Mineral Processing, 44-45*, 59-69. doi: https://doi.org/10.1016/0301-7516(95)00018-6

Zhang, Y., & Qin, S. J. (2007). Fault detection of nonlinear processes using multiway kernel independent component analysis. *Industrial and Engineering Chemistry Research, 46*(23), 7780-7787. doi: 10.1021/ie070381q

10. APPENDICES

Appendix A. Plots of NOC and faulty data samples of TE process onto 2D PCA subspace along with the trained 1-class GMM (K=2) decision boundary



NOC (validation)

Fault 1

Fault 2

Fault 3

Fault 4

Fault 5

Fault 6

Fault 7

Fault 8

Fault 9

Fault 10

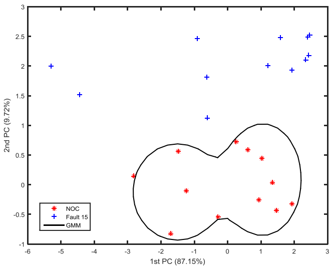
Fault 11

Fault 12

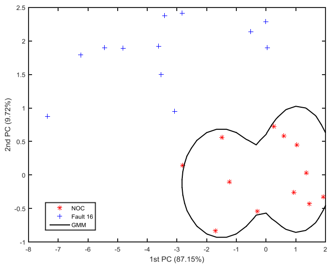
Fault 13

Fault 14

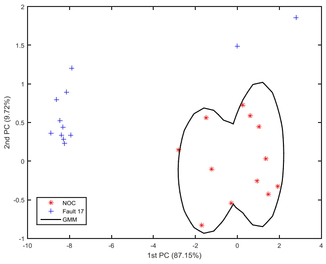
Fault 15



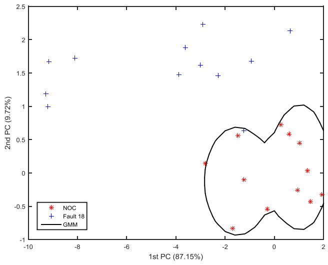
Fault 16



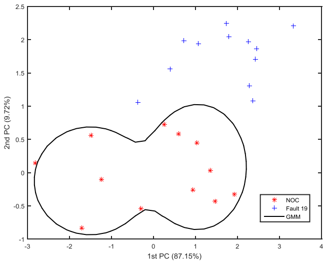
Fault 17



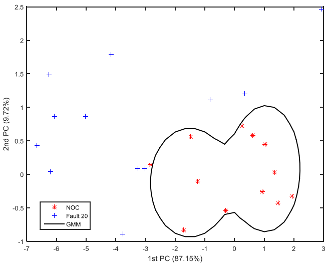
Fault 18



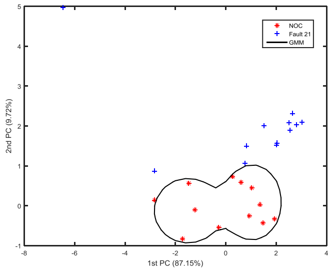
Fault 19



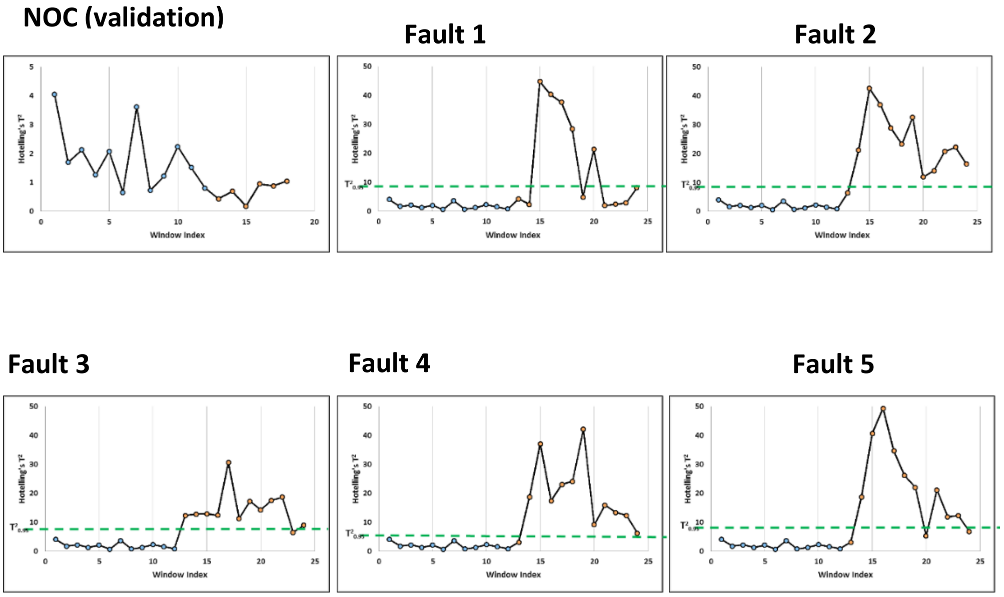
Fault 20



Fault 21



Appendix B. Plots of Hotelling's T2 of the NOC (blue) and faulty (orange) data samples of the Tennessee-Eastman process.


NOC (validation)

Fault 1

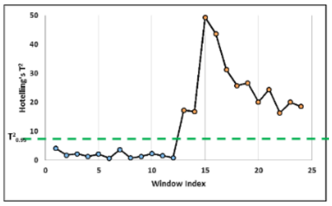
Fault 2

Fault 3

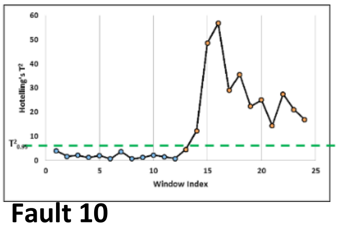
Fault 4

Fault 5

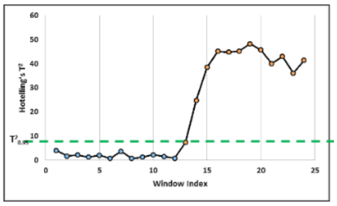
Fault 6



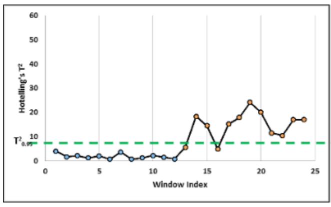
Fault 7



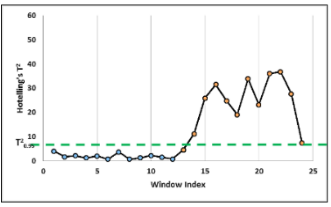
Fault 8



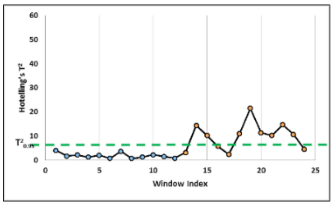
Fault 9



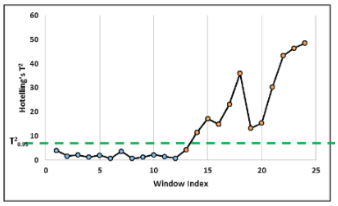
Fault 10



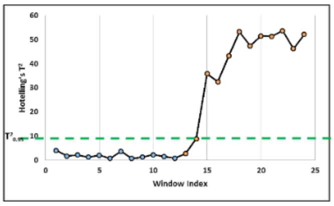
Fault 11



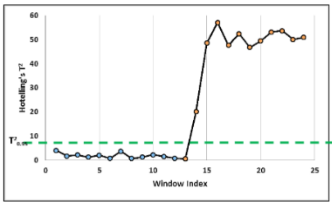
Fault 12



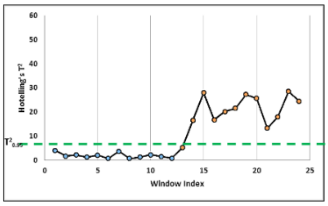
Fault 13



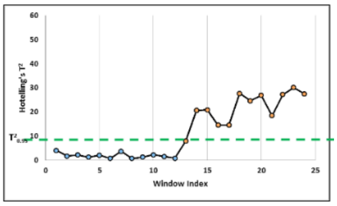
Fault 14



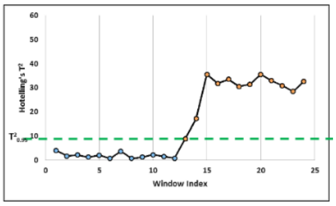
Fault 15



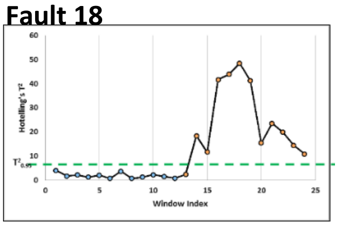
Fault 16



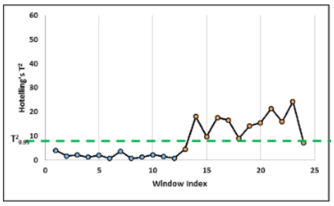
Fault 17



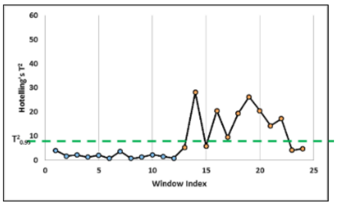
Fault 18



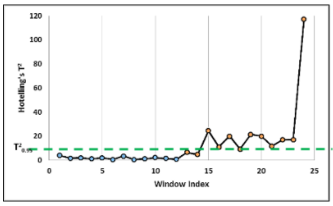
Fault 19



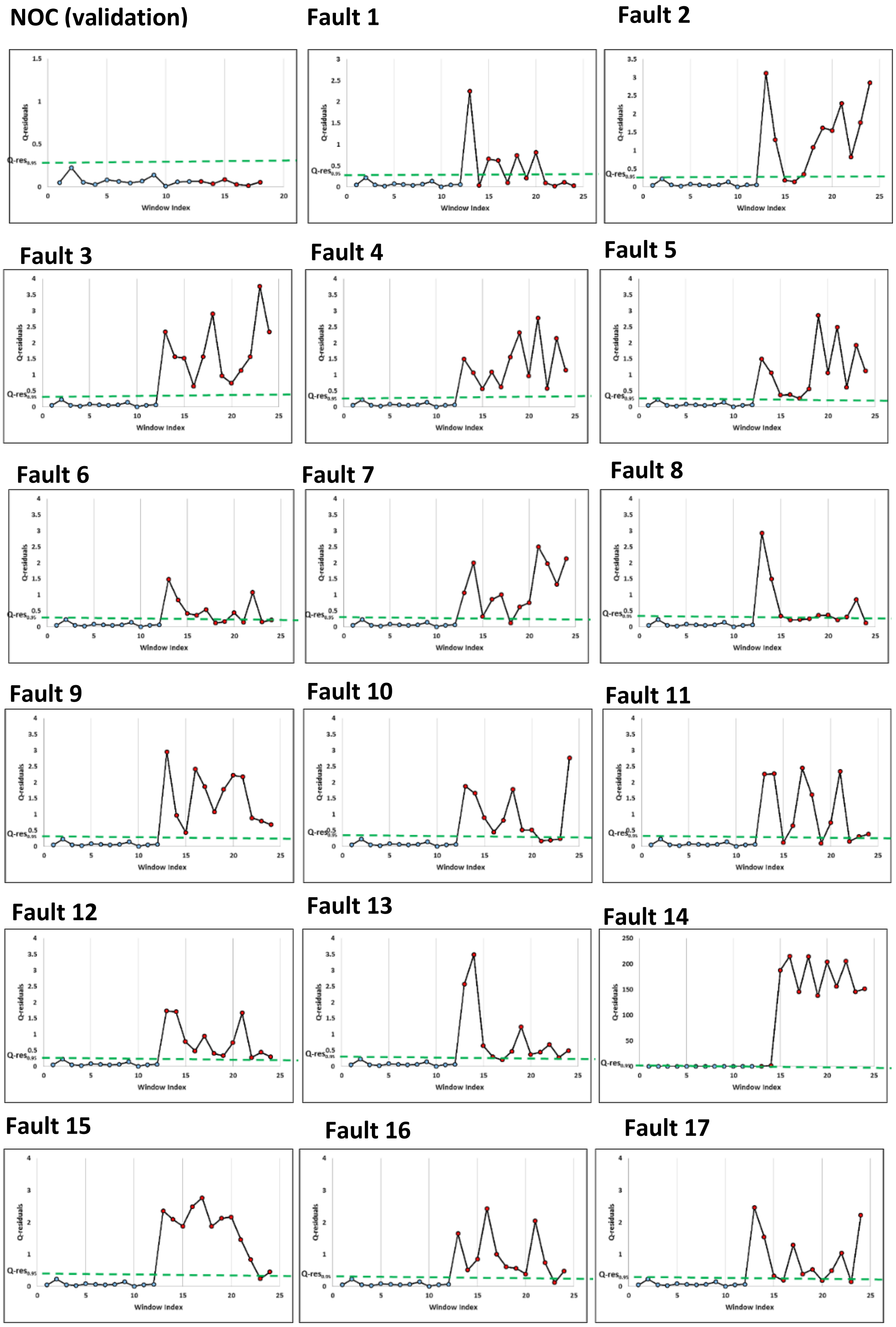
Fault 20



Fault 21



Appendix C. Plots of Q-residuals of the NOC (blue) and faulty (red) data samples of the Tennessee-Eastman process.



NOC (validation)

Fault 1

Fault 2

Fault 3

Fault 4

Fault 5

Fault 6

Fault 7

Fault 8

Fault 9

Fault 10

Fault 11

Fault 12

Fault 13

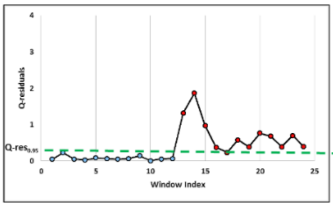
Fault 14

Fault 15

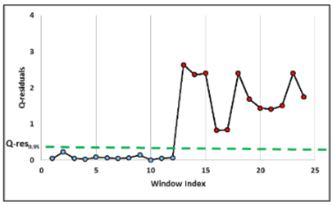
Fault 16

Fault 17

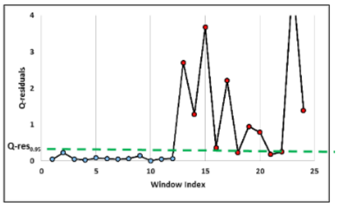
Fault 18



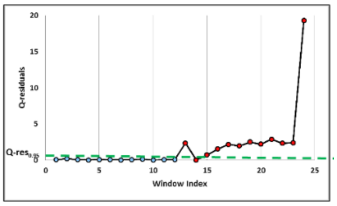
Fault 19



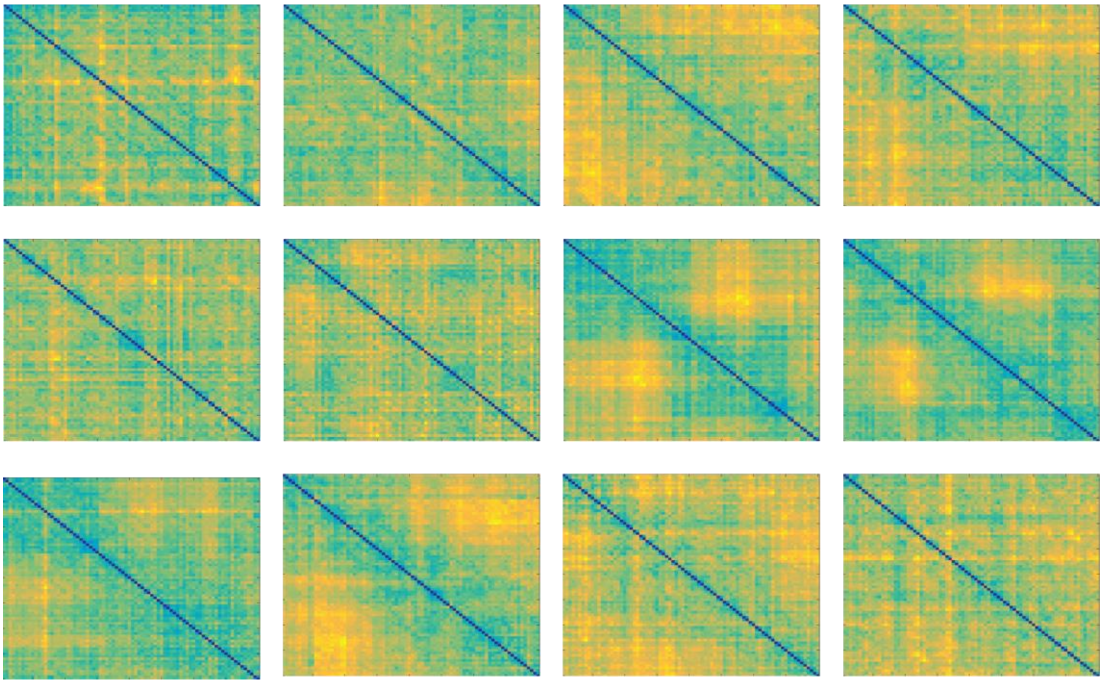
Fault 20



Fault 21



Appendix D. Distance matrix plots of NOC in Tennessee Eastman Problem



Appendix E. Pseudo-code used in this thesis (performed in MATLAB® R2015a)

DISTANCE MATRIX CALCULATION

```
%default: Euclidean distance “Euclidean”, use other distance
metrics, e.g. “chebychev”, “mahalanobis”
%Example:
%for i=1:N
%q=squareform(pdist(Y(:,i, “chebychev”))); DD{i}=q; end

for i=1:N
q=squareform(pdist(Y(:,i)));
DD{i}=q;
end
```

SEGMENTATION OF TIME SERIES

```
function [Y,N] = Segment(X,n,k)
%Parameters:
%INPUTS:
%X--> time series data (normalised, axb matrix, a samples,
b %dimensions (b=1 for univariate) )
%n--> window width (estimated using ACF)
%k--> sliding step
%OUTPUTS:
%Y--> segmented time series (nxN matrix)
%N-->no. of segments
n>=2;
k>=1;
ind = bsxfun(@plus, 1:n, (0:k:numel(X)-n).');
out = X(ind);
Y=out';
sz=size(Y);
N=sz(2);
```

FOR NON-OVERLAPPING SEGMENTS

```
[Y,DD,RP,CRQA] = RP_DD_RQA(X,n,m,t,r,index)

% Inputs are :
% X - time-series data
% n - window width/sliding step (non-overlapping segments)
% m - embedding dimension (1 for this study)
% t - delay time (1 for this study)
% r - neighbourhood size
% index - (0 or 1) - 0 for Euclidean, 1 for maximum distance

% Outputs are:
% Y - segmented data matrix
% DD - distance matrix
% RP - Recurrence Plot matrix
%CRQA – RQA matrix (13 features)

n>=2;
sz=size(X);
N=fix(sz(1)/n); %N - no. of segments
%% DATA SEGMENTATION

for i=1:N
Y(1:n,i)=X(n*i-(n-1):n*i)
end
%% DD and RP CALCULATION (based from RPplot.m of (Li, Ouyang,
Yao, & Guan, 2004))

for i=1:N
[rp,dd]=RPplot(Y(:,i)',m,t,r,index);
RP(n*i-(n-1):n*i,:)=rp;
DD(n*i-(n-1):n*i,:)=dd;
nd


%% CRQA Feature Extraction ((N. Marwan, 2013))
for i=1:N
r=crqa(Y(:,i),m,t,r,n,1);
CRQA(i,:)=r;
end
```

DISTANCE MATRIX PLOTTING

```
function savedd(DD)
for i=1:N
figure(i)
imagesc(DD(n*i-(n-1):n*i,:));
saveas(figure(i),sprintf('DD%05d.jpg',i));
set(gcf, 'Visible', 'off')
end
```