Application of Machine Learning in Fault Detection Using Control Chart Pattern Recognition

Talayeh Razzaghi, Ph.D.

Assistant Professor
Department of Industrial Engineering
New Mexico State University
Outline

Introduction

Control Charts

Control Chart Patterns

Imbalanced Classification

Proposed Methodology: Cost-sensitive learning based algorithm

Computational Results and Discussion

Conclusion
Introduction

Types of Analytical Models:

1. Descriptive Models
   - Shaping the questions and data into a structured problem
Introduction

Types of Analytical Models:

1. Descriptive Models
   • Shaping the questions and data into a structured problem

2. Predictive Models
   • Understanding the data and predicting the future
Introduction

Types of Analytical Models:

1. Descriptive Models
   - Shaping the questions and data into a structured problem

2. Predictive Models
   - Understanding the data and predicting the future

3. Prescriptive Models
   - Seeking optimal decisions to alter the future
Introduction

Types of Analytical Models:
1. Descriptive Models
   ▶ Shaping the questions and data into a structured problem

2. Predictive Models
   ▶ Understanding the data and predicting the future

3. Prescriptive Models
   ▶ Seeking optimal decisions to alter the future
Introduction

- Predictive models are of interest to statisticians, computer scientists, and us (industrial engineers)!

- They are referred to with terms such as statistical learning, machine learning, and data mining.

- They have been applied to several applications.
  - Image Recognition
  - Manufacturing
  - Health Informatics
  - Cybersecurity
Introduction

- Predictive models are of interest to statisticians, computer scientists, and us (industrial engineers)!

- They are referred to with terms such as statistical learning, machine learning, and data mining.

- They have been applied to several applications.
  - Image Recognition
    Why has Image Recognition been at the center of attention for predictive analytics?
  - Manufacturing

- Health Informatics

- Cybersecurity
Introduction

![Image of a dog and a cat lying together]
Introduction
Introduction

The Bad and Good news:

- Not all applications offer a set of clean, perfect, and problem-free data to work.
- It is challenging to recognize and “treat” the issues that appear in real-world datasets.
- Examples of issues: imbalanced-ness, outliers, missing values, and massive size datasets
Control Charts

- Control charts are used for monitoring the behavior of a process.
- Control charts, also known as Shewhart charts (Walter A. Shewhart, 1920) or process-behavior charts.
- Control charts are a statistical process control tool used to determine if a manufacturing, chemical or business process is in a state of control.
Control charts are useful to identify not only out-of-control points but also the type of patterns:

- Up trend
- Down trend
- Cyclic
- Systematic
- Up shift
- Down shift
- Stratification
Imbalanced Classification

- An application in Quality Control (Control Chart Pattern Recognition) *
  - Trend patterns
    - Stamping tonnage
    - Abnormal signals
  - Shift patterns
    - Variations of machine, material/operator
  - Cyclic Patterns
    - Voltage variability
    - Automotive body assembly
  - Systematic Patterns
    - Automotive body assembly

Western Electric Company (1958)
Control Chart Pattern Recognition (CCPR)

- However, one important parameter has been neglected!!
  - Abnormal patterns are rare but important to detect
  - Normal patterns are common
- CCPR belongs to the category of imbalanced classification
Imbalanced Data

Applications:
- Breast cancer detection (Verma et al., 2010)
- Credit card fraud detection (Wei et al., 2012)
- Oil spills detection in satellite radar images (Kubat et al., 1998)
- Network intrusion detection (Xu et al., 2011)
- Control chart pattern recognition (Xanthopolous & Razzaghi, 2014)

Binary Classification Problem Definition

Preliminaries:
- Data represented by \((x_i, y_i) \in \mathbb{R}^m \times \{-1, 1\}\)
  - \(x_i\): actual data
  - \(y_i\): corresponding label (binary case)

Classification Problem:
- Find a classifier function \(f : \mathbb{R}^m \mapsto \{-1, 1\}\)
- It can be used to predict the labels \(y_i^{test}\) of a group of data samples \(x_i^{test}\)
- Classification performance is evaluated through performance measures such as Accuracy, Sensitivity, Specificity and G-mean

Support Vector Machines (Vapnik, 2000):
- Classifier is obtained from solution of a Quadratic Optimization problem (Computationally tractable)
- Less over fitting in practice (unlike Artificial Neural Networks)
- Nice optimization problem structure
Proposed Methodology

- **Hard Margin Support Vector Machines**

\[ \begin{align*} 
\text{Maximize} \ (\text{objective}) \ & \ \text{the separation margin } (2/\|w\|) \ \text{subject to} \\
\text{correct classification} \ (\text{constraints}) \\
\min_{w,b} \ & \ \frac{1}{2} \|w\|^2 \\
\text{s.t.} \ & \ y_i (w^T x_i - b) \geq 1, \quad i = 1, \ldots, n 
\end{align*} \]
Dual Formulation

- An arbitrary data sample \( x_u \) is assigned to a class \( y_u \) based on the following rule:
  \[
y_u = \text{sgn}(w^T x_u - b)
\]
where \( \text{sgn}(\cdot) \) is the sign function

- The separation hyperplane can be computed as follows:
  \[
w^* = \sum_{i=1}^{n} y_i \alpha_i^* x_i, \quad b^* = -\frac{\max_{y_i=-1} \langle w^* x_i \rangle + \min_{y_i=1} \langle w^* x_i \rangle}{2}
\]
where \( a_i \) are the dual variables (or Lagrange multipliers associated with the the \( i^{\text{th}} \) constraint of the primal)
Inseparable Case: Soft Margin SVM

\[
\begin{align*}
\min_{w,b,\xi} & \quad \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \xi_i \\
\text{s.t.} & \quad y_i (w^T x_i - b) \geq 1 - \xi_i, \quad i = 1, \ldots, n
\end{align*}
\]  

Parameter $C$ controls misclassification penalty
Inseparable Case: Soft Margin SVM

The dual is calculated by the Karush-Kuhn-Tucker (KKT) conditions.

\[
\begin{align*}
\text{max} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\
\text{s.t.} \quad & \sum_{j=1}^{n} \alpha_i y_i = 0 \\
\quad & 0 \leq \alpha_i \leq C \quad \text{for } i = 1, \ldots, n
\end{align*}
\]
Extension to Nonlinear Classification (Kernels)

- Often the data sets are not linearly separable and the soft margin SVM, while feasible, yields poor performance (Cristianini and Shawe-Taylor, 2000)
Extension to Non Linear Classification (Kernels)

- Embed data from **input space** to a higher dimension **feature space**
- This is done through an embedding function $\phi(x)$
- We denote $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$
- Popular kernel functions include:

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polynomial*</td>
<td>$(a x_i^T x_j + c)^d$</td>
</tr>
<tr>
<td>RBF</td>
<td>$\exp(-\gamma |x_i - x_j|^2)$</td>
</tr>
<tr>
<td>Cauchy</td>
<td>$(1 + \frac{1}{\alpha} |x_i - x_j|^2)^{-1}$</td>
</tr>
<tr>
<td>Inverse multi quadratic</td>
<td>$(|x_i - x_j|^2 + \alpha^2)^{-1/2}$</td>
</tr>
</tbody>
</table>

* For $a = 1, c = 0$ and $d = 1$ it is a linear kernel
Imbalanced Classification

Methods:
- Resampling (Chawla et al., 2002)
- Ensemble Learning (Boosting, bagging, etc.) (Freund and Schapire, 1997)
- **Cost-sensitive Learning** (Veropoulos et al., 1999)
Cost-Sensitive SVM

- Penalize misclassification of each class with different coefficient (Veropoulos, 1999)

\[
\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C^+ \sum_{\{i|y_i=+1\}} \xi_i + C^- \sum_{\{i|y_i=-1\}} \xi_i
\]

s.t. \( y_i(w^T \phi(x_i) - b) \geq 1 - \xi_i, \quad i = 1, \ldots, n \)  
\( \xi_i \geq 0, \quad i = 1, \ldots, n \) \hspace{1cm} (6a, 6b, 6c)

- The weights are usually chosen to be inversely proportional to the size of each class \( n^+ \) and \( n^- \):

\[
C^+ = \frac{C}{n^+}, \quad C^- = \frac{C}{n^-} \hspace{1cm} (7)
\]
Proposed Methodology

(a) Linear SVM  
(b) Linear WSVM  
(c) SVM with RBF kernel  
(d) WSVM with RBF kernel
Performance Measures

- **Accuracy**: the percent of the correctly classified examples over the total number of examples

- **Sensitivity**

- **Specificity**

\[
Sensitivity = \frac{TP}{TP + FN}, \quad Specificity = \frac{TN}{TN + FP}
\] (8)

- **G-mean**

\[
G – Mean = \sqrt{Sensitivity \times Specificity}
\] (9)

**Table**: Confusion Matrix

<table>
<thead>
<tr>
<th></th>
<th>Positive class</th>
<th>Negative class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive class</td>
<td>TP</td>
<td>FP</td>
</tr>
<tr>
<td>Negative class</td>
<td>FN</td>
<td>TN</td>
</tr>
</tbody>
</table>
Other Performance Measures for CCPR

- **Average Target Pattern Run Length (ATPRL)** (Hwarng & Hubele, 1991): the average number of samples needed for discovering an abnormal pattern.

- **Average Run Length Index (ARLIDX)** (Hwarng & Hubele, 1991): which equals to the fraction of ATPRL divided by the discovery rate of abnormal patterns.

- The ARL-based measures are important especially for applications where the production of each sample is cost and labor intensive.

- Ultimately one wants to detect an anomaly with the **lower** ATPRL possible.
Experimental Setup

- SVM and WSVM models were solved using LIBSVM-3.12 and LIBSVM-weights-3.12.
- Data processing and further scripting were done in MATLAB.
- Experiments were conducted for highly imbalanced problems where 97.5% of the data belong to the normal class and only 2.5% belong to the abnormal.
- For each classification problem, we generate a total of 1000 data points and for cross validation purposes, 90% of the data was used for training and the rest 10% was used for testing.
- All data are normalized prior to classification, so that they have zero mean and unitary standard deviation.
- Radial basis function (RBF) kernel was used.
LIBSVM -- A Library for Support Vector Machines

Chih-Chung Chang and Chih-Jen Lin

 LIBSVM tools provides many extensions of LIBSVM. Please check it if you need some functions not supported in LIBSVM.
 We now have a nice page LIBSVM data sets providing problems in LIBSVM format.
 A practical guide to SVM classification is available now! (mainly written for beginners)

We now have an easy script (easy.py) for users who know NOTHING about SVM. It makes everything automatic--from data scaling to parameter selection.
The parameter selection tool grid.py generates the following contour of cross-validation accuracy. To use this tool, you also need to install python and gnuplot.
Computational Results

- SVM results in **poor classification performance** for inseparable and partially separable cases
- Our proposed WSVM is **effective** for CCPR in a highly imbalanced environment!
SVMs: more than 2 classes?

- The SVM as defined works for \( K = 2 \) classes. What do we do if we have \( K > 2 \) classes?
  - **One versus All (OVA):** Fit \( K \) different 2-class SVM classifiers \( \hat{f}_k(x) \), \( k = 1, \ldots, K \); each class versus the rest. Classify \( x^* \) to the class for which \( \hat{f}_k(x^*) \) is largest.
  - **One versus One (OVO):** Fit all \( \binom{K}{2} \) pairwise classifiers \( \hat{f}_{kl}(x) \). Classify \( x^* \) to the class that wins the most pairwise competitions.
- Which to choose? If \( K \) is not too large, use OVO.
Multi-class classification

- The weighting Strategy for multi-class WSVM for CCPR

\[ C_i = \frac{C}{n_i} \quad i = 1, 2, \ldots, m \]  

Table: Classification results for multi-class SVM and WSVM for CCPR with window length=10 and highly imbalanced data. Rows are related to predicted class labels and the columns are related to real labels.

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Dt</th>
<th>Ut</th>
<th>S</th>
<th>Ds</th>
<th>Us</th>
<th>C</th>
<th>Str</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.05</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Dt</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Ut</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>S</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.95</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Ds</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Us</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>C</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Str</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SVM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>0.65</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.15</td>
<td>0.19</td>
<td>0.13</td>
<td>0.31</td>
</tr>
<tr>
<td>Dt</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Ut</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>S</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Ds</td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.75</td>
<td>0.00</td>
<td>0.00</td>
<td>0.08</td>
</tr>
<tr>
<td>Us</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.77</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>C</td>
<td>0.07</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.04</td>
<td>0.70</td>
<td>0.00</td>
</tr>
<tr>
<td>Str</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.10</td>
<td>0.00</td>
<td>0.17</td>
<td>0.61</td>
</tr>
</tbody>
</table>
## Multi-class classification

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Dt</th>
<th>Ut</th>
<th>S</th>
<th>Ds</th>
<th>Us</th>
<th>C</th>
<th>Str</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.40</td>
<td>1.00</td>
<td>0.47</td>
<td>1.00</td>
</tr>
<tr>
<td>Dt</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Ut</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SVM</td>
<td></td>
<td></td>
<td></td>
<td>S</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ds</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Us</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Str</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>WSVM</td>
<td>N</td>
<td>0.98</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.20</td>
<td>0.37</td>
<td>0.27</td>
</tr>
<tr>
<td></td>
<td>Dt</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>Ut</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>Ds</td>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.80</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>Us</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.63</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.73</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>Str</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Table: Classification results for multi-class SVM and WSVM for CCPR with window length=50 and highly imbalanced data. Rows are related to predicted class labels and the columns are related to real labels.
Wafer dataset (Adopted from UCR Time Series Classification Archive)

- Electronics manufacturing usually involves a large number of steps (> 250) which can induce defects to the final product.
- Quality control is performed by recording the different frequencies that are emitted by the plasma during the process.
- The data set composed of 1000 training samples (of length 152 each) and 6174 testing samples of the same length (Olszewski, 2001; Keogh et al., 2011). The training samples are imbalanced (903 are majority and 97 minority).

### Table: Performance for the wafer manufacturing industry dataset

<table>
<thead>
<tr>
<th></th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Gmean</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td><strong>0.9996</strong></td>
<td>0.9160</td>
<td>0.9156</td>
<td><strong>0.9913</strong></td>
</tr>
<tr>
<td>WSVM</td>
<td>0.9967</td>
<td><strong>0.9350</strong></td>
<td><strong>0.9319</strong></td>
<td>0.9905</td>
</tr>
<tr>
<td><strong>Testing</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td><strong>0.9971</strong></td>
<td>0.9654</td>
<td>0.9811</td>
<td><strong>0.9937</strong></td>
</tr>
<tr>
<td>WSVM</td>
<td>0.9895</td>
<td><strong>0.9895</strong></td>
<td><strong>0.9895</strong></td>
<td>0.9895</td>
</tr>
</tbody>
</table>
Results (cont’d)

Figure: WSVM training and testing time vs. training size for cyclic pattern
Results (cont’d)

- For all patterns and most problem instances, WSVM has lower ARLIDX

- Lower ARLIDX are obtained compared to the ARLIDX in

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Uptrend</th>
<th>Upshift</th>
<th>Systematic</th>
<th>Cyclic</th>
<th>Stratification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SVM</td>
<td>WSVM</td>
<td>SVM</td>
<td>WSVM</td>
<td>SVM</td>
</tr>
<tr>
<td>0</td>
<td>155.54</td>
<td>155.19</td>
<td>155.06</td>
<td>155.21</td>
<td>155.19</td>
</tr>
<tr>
<td>0.005</td>
<td>16.33</td>
<td>14.60</td>
<td>138.41</td>
<td>111.11</td>
<td>110.83</td>
</tr>
<tr>
<td>0.03</td>
<td>13.04</td>
<td>9.83</td>
<td>129.87</td>
<td>86.07</td>
<td>106.90</td>
</tr>
<tr>
<td>0.055</td>
<td>11.34</td>
<td>6.64</td>
<td>81.49</td>
<td>81.75</td>
<td>94.95</td>
</tr>
<tr>
<td>0.08</td>
<td>9.46</td>
<td>6.61</td>
<td>103.12</td>
<td>66.67</td>
<td>80.73</td>
</tr>
<tr>
<td>0.105</td>
<td>8.37</td>
<td>7.28</td>
<td>106.51</td>
<td>59.09</td>
<td>80.36</td>
</tr>
<tr>
<td>0.13</td>
<td>8.09</td>
<td>7.84</td>
<td>90.50</td>
<td>51.73</td>
<td>62.92</td>
</tr>
<tr>
<td>0.155</td>
<td>7.50</td>
<td>6.90</td>
<td>70.50</td>
<td>53.83</td>
<td>74.21</td>
</tr>
<tr>
<td>0.18</td>
<td>7.10</td>
<td>6.06</td>
<td>73.83</td>
<td>33.55</td>
<td>59.54</td>
</tr>
<tr>
<td>0.205</td>
<td>6.81</td>
<td>6.24</td>
<td>62.91</td>
<td>24.07</td>
<td>67.33</td>
</tr>
<tr>
<td>0.23</td>
<td>7.47</td>
<td>7.12</td>
<td>76.66</td>
<td>20.64</td>
<td>77.87</td>
</tr>
<tr>
<td>0.255</td>
<td>7.01</td>
<td>6.89</td>
<td>37.44</td>
<td>18.59</td>
<td>62.76</td>
</tr>
<tr>
<td>0.28</td>
<td>7.66</td>
<td>6.97</td>
<td>40.74</td>
<td>6.99</td>
<td>79.42</td>
</tr>
<tr>
<td>0.305</td>
<td>7.01</td>
<td>6.75</td>
<td>49.65</td>
<td>7.62</td>
<td>52.66</td>
</tr>
<tr>
<td>0.33</td>
<td>7.27</td>
<td>6.94</td>
<td>60.99</td>
<td>6.82</td>
<td>36.13</td>
</tr>
<tr>
<td>0.355</td>
<td>6.49</td>
<td>6.27</td>
<td>37.28</td>
<td>8.05</td>
<td>49.41</td>
</tr>
<tr>
<td>0.38</td>
<td>7.50</td>
<td>7.37</td>
<td>25.59</td>
<td>6.52</td>
<td>48.56</td>
</tr>
<tr>
<td>0.405</td>
<td>6.67</td>
<td>6.69</td>
<td>24.99</td>
<td>6.57</td>
<td>27.87</td>
</tr>
<tr>
<td>0.43</td>
<td>6.75</td>
<td>6.69</td>
<td>18.74</td>
<td>6.04</td>
<td>22.04</td>
</tr>
<tr>
<td>0.455</td>
<td>6.76</td>
<td>6.76</td>
<td>19.08</td>
<td>6.13</td>
<td>23.24</td>
</tr>
<tr>
<td>0.48</td>
<td>6.59</td>
<td>6.48</td>
<td>15.13</td>
<td>5.59</td>
<td>25.15</td>
</tr>
<tr>
<td>0.505</td>
<td>6.50</td>
<td>6.52</td>
<td>14.86</td>
<td>5.86</td>
<td>16.26</td>
</tr>
<tr>
<td>0.53</td>
<td>6.85</td>
<td>6.84</td>
<td>13.81</td>
<td>6.34</td>
<td>13.82</td>
</tr>
<tr>
<td>0.555</td>
<td>6.69</td>
<td>6.69</td>
<td>13.33</td>
<td>6.40</td>
<td>15.65</td>
</tr>
<tr>
<td>0.58</td>
<td>6.51</td>
<td>6.51</td>
<td>12.59</td>
<td>5.44</td>
<td>22.55</td>
</tr>
<tr>
<td>0.605</td>
<td>6.61</td>
<td>6.65</td>
<td>16.56</td>
<td>6.20</td>
<td>20.11</td>
</tr>
<tr>
<td>0.63</td>
<td>6.44</td>
<td>6.43</td>
<td>11.40</td>
<td>5.85</td>
<td>13.22</td>
</tr>
<tr>
<td>0.655</td>
<td>6.45</td>
<td>6.50</td>
<td>13.81</td>
<td>5.75</td>
<td>12.36</td>
</tr>
<tr>
<td>0.68</td>
<td>6.21</td>
<td>6.23</td>
<td>10.61</td>
<td>5.71</td>
<td>10.90</td>
</tr>
<tr>
<td>0.705</td>
<td>6.29</td>
<td>6.30</td>
<td>12.46</td>
<td>6.32</td>
<td>16.04</td>
</tr>
<tr>
<td>0.73</td>
<td>6.38</td>
<td>6.34</td>
<td>8.33</td>
<td>5.89</td>
<td>10.78</td>
</tr>
<tr>
<td>0.755</td>
<td>6.26</td>
<td>6.23</td>
<td>10.09</td>
<td>5.91</td>
<td>10.19</td>
</tr>
<tr>
<td>0.78</td>
<td>6.18</td>
<td>6.21</td>
<td>8.07</td>
<td>5.93</td>
<td>11.58</td>
</tr>
</tbody>
</table>
Conclusion

- The proposed WSVM is more effective for imbalanced learning in CCPR problem.
- Current study results are encouraging enough in terms of average run length, computational time, and G-mean.
- WSVM multi-class classification helps to detect the abnormal points based on their types which outperforms SVM multi-class classification under a highly imbalanced environment.
- Accuracy might not be a proper performance indicator for imbalanced classification problems.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (Plat, 1999).
- For nonlinear boundaries, kernel SVMs are popular. Can use kernels with LR and LDA as well, but computations are more expensive.
Thank you!