Decision Science Letters 12 (2023) 487–498

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

Fuzzy support vector machine for classification of time series data: A simulation study

Hartayuni Sain[a,b], Heri Kuswanto[a*], Santi Wulan Purnami[a] and Santi Puteri Rahayu[a]

[a]Statistics Department, Faculty of Science and Data Analytics, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia
[b]Statistics Study Program, Mathematics and Natural Sciences Faculty, Tadulako University, Palu 94118, Indonesia

C H R O N I C L E

Article history:
Received: December 12, 2022
Received in revised format: January 22, 2023
Accepted: May 12, 2023
Available online: May 12, 2023

Keywords:
Fuzzy
Support Vector Machine
FSVM
Classification of time series data
Multiclass imbalanced

A B S T R A C T

Support vector machine (SVM) has become one of most developed methods for classification, focusing on cross-sectional analysis. However, classification of time series data is an important issue in statistics and data mining. Classification of time series data using SVMs that focus on cross-sectional data leads to improper classification, and hence, the SVM needs to be extended for handling time series dataset. As with cross-section data, the problem of imbalanced data is also common in time series data. Fuzzy method has been proven to be capable of overcoming the case of imbalanced data. In this paper, we developed a Fuzzy Support Vector Machine (FSVM) model to classify time series data with imbalanced class. The proposed method puts the fuzzy membership function on the constraint function. Through simulation studies, this research aims to assess the performance of the developed FSVM in classifying time series data. Based on the classification accuracy criteria, we prove that the proposed FSVM method outperforms the standard SVM method for the classification of multiclass time series data.

1. Introduction

Classification is a grouping of data which has a class or target label, with the aim that it can be used for predicting the value of the output from a provided input. One of our most powerful classification methods is SVM introduced by Vapnik (1995). The SVM method has proven to be a method that can solve overfitting problems by minimising the upper bound of the generalisation error. The method works by finding the best hyperplane that serves as a separator of any two classes in the given space of inputs. Moreover, SVM is also able to obtain a global optimal solution and always arrive at the same solution for each run (Farquad & Bose, 2012).

The SVM proposed by Vapnik (1995) was initially developed only to solve binary classification problems. Furthermore, in its development SVM has been expanded to handle multi-class classification problems but limited to analyzing cross-sectional. Classification of time series data becomes an important issue in statistics and data mining. Jeong et al. (2011) proposed a new penalty-based dynamic time warping (DWT) distance measure and a modified logistic weight function. It is proven to improve the classification and clustering performance of time series data. Ismail Fawaz et al. (2019) used Deep Neural Networks (DNN) method in the classification of time series data in the form of images, DNN in achieving advanced performance in document classification and speech recognition. While Bostrom and Bagnall (2017), Hills et al. (2014), Lines et al. (2012) used shapelet transformation method for classifying time series data. SVM has also been extended to have capability for making predictions called Support Vector Regression (SVR), and it has been applied to time series dataset. This paper extends the SVM to classify time series dataset, instead of making predictions. The development of

* Corresponding author.
E-mail address: heri_k@statistika.its.ac.id (H. Kuswanto)

doi: 10.5267/dsl.2023.5.002

SVM for classification of time series data is still limited. A study related to the time series data classification was conducted by Zhang et al. (2010) who used the SVM method and proposed the use of Gaussian elastic metric kernel (GEMK).

Recent developments in data mining classification have attempted to incorporate the concept of a fuzzy method with several machine learning methods such as SVM, Neural Network (NN) and some others. Lin and Wang (2002), the combination of fuzzy and SVM methods due to the robustness of SVM in handling several issues such as outlier, noise and imbalanced class response. Whereas fuzzy method has been proven to be capable of overcoming the presence of outlier, noise and imbalanced data. Therefore, combining fuzzy with SVM (hereafter denoted as FSVM) is expected to be a powerful approach to deal with those issues. Among studies that use hybrid methods are a study conducted by Abe and Inoue (2002) who used fuzzy-SVM to overcome the problem of unclassifiable data. Another study conducted by Mohammadi and Sarmad (2019), used the FSVM method to solve data classification with outlier problems. Similar research was also conducted by Lee et al. (2006) who used the FSVM method and outlier detection algorithm (ODA).

Another problem is unbalanced data, which arises where the size of data in one or more classes is far smaller than the size of data in another class, or vice versa. Various methods have been developed to overcome this problem. As in Batuwita and Palade's (2010) research, using FSVM in the case of class imbalance learning. Then research conducted by Gu, Ni and Wang (2014) used FSVM for class imbalance problems. Research by Wu, Shen and Zhang (2017) proposed FSVM for imbalanced data problems. Similar research was conducted by Sain and Purnami (2015) by combining the use of sampling methods and SVM for imbalance data classification problems. Fan et al. (2017) proposed the FSVM method to solve imbalanced data and developed entropy-based fuzzy membership.

The research mentioned before applies the hybrid concept to cross-section data classification. However, as research on classification in general, the concept is still not much developed for classifying time series data. The use of the FSVM method for cross-section data classification has been very successful, as can be seen from the previous studies described above. However, to our knowledge, so far there are no methods that have been developed to classify time series data using FSVM, in particular with the unbalanced response class.

This research is focused on developing a classification method for time series data using fuzzy support vector machines (FSVM), with a direct multiclass SVM following the work of Cramer and Singer (2002) to handle multiclass problems. The studies by Cramer and Singer (2002) as well as Tsang, Yeung and Chan (2003) used FSVM to classify binary data by placing a fuzzy membership function on the constraint function applied to time series data. Through simulation studies, this research aims to assess the performance of the developed FSVM in classifying time series data.

2. Material and Methods

2.1. Fuzzy

Lotfi Zadeh (1965) introduced fuzzy logic whose basic ideas are inclusion, union, intersection, complement, relation and convexity. Fuzzy logic allows and even exploits tolerance for imprecision, with truth values ranging between 0 and 1. Fuzzy sets are sets without boundaries that are firm and fast (Mandal, Choudhury and Chaudhuri, 2012). Fuzzification is a process of transforming an input into a fuzzy representation, usually in terms of fuzzy sets and their respective membership functions. The next stage is defuzzification, which is the stage of changing fuzzy output to crisp output. The input value for the defuzzification stage is a fuzzy set for which the output will be a single number.

2.1.1. Membership Function

The membership function curve shows the mapping of the entered data points to their membership values. A functional approach can be used to obtain the membership value. Membership in Fuzzy sets has different forms consisting of linear, bell, gaussian, trapezoidal and triangular forms. In the following, one form of membership function is described, namely the triangular membership function. This membership function is basically a combination of two lines (linear).

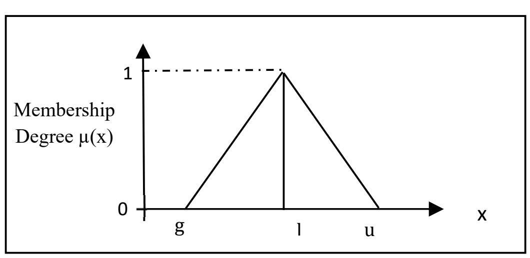


Fig. 1. Illustration for the triangular membership function

From Fig. 1, the membership function is

$$\mu(x)=\begin{cases}0 & ;\ x\le g \text{ atau } x\ge u \\ \dfrac{(x-g)}{(l-g)} & ;\ g\le x\le l \\ \dfrac{(u-x)}{(u-l)} & ;\ l\le x\le u\end{cases} \tag{1}$$

2.2. SVM

SVM was proposed by Vapnik (1995) only to solve binary classification problems but in its development SVM has been expanded to solve multiclass classification. There are two techniques that can be used to solve multiclass problems in classification using SVM i.e., solving optimization problems where all data is used as training data or to construct a binary classification from roots to leaves. However, along with the development of SVM methods to overcome multi-class problems can use the one versus rest and one versus one scheme methods. In addition to these two schemes, it can also use the method SVM multiclass developed by Weston and Watkins (1999) and Crammer and Singer (2002).

2.2.1. SVM Multiclass

According to Xue, Yang and Chen (2014), multiclass classification (k>2) is generally decomposed into a set of binary problems, so that SVM can be directly applied. There are two approaches that can be used, namely the one-versus-all or OVA approach (Vapnik, 1998) and the one-versus-one or OVO approach (Krebel, 1999). The OVA approach constructs $\boldsymbol{m}$ binary classifications separately for $\boldsymbol{m}$-class classifications. If one of the classes is positive (+1) then the remaining $\boldsymbol{m-1}$ class as negative (-1). While the OVO approach is also called the pairwise approach, it evaluates all possible classifications so that there are $\boldsymbol{m(m-1)/2}$ binary classifications.

2.2.2. Support Vector Machine Multiclass Crammer Singer

One of the approaches for direct multiclass SVM is by using the direct II multiclass SVM method developed by Crammer and Singer (2002). Crammer and Singer (2002) represent an "all together" approach to solving multiclass SVM problems,

$$\min_{\mathbf{w}_m\in H,\xi\in\mathbb{R}^l} \frac{1}{2}\sum_{m=1}^{k}\mathbf{w}_m^T\mathbf{w}_m + \sum_{i=1}^{l}\xi_i \tag{2}$$

with the constrained function

$$\mathbf{w}_{y_i}\varphi(\mathbf{x}_i)-\mathbf{w}_m\varphi(\mathbf{x}_i)\ge 1-\delta_{y_i,t}-\xi_i \tag{3}$$

3. Proposed Method

3.1. FSVM Method for Time Series Data

Given a training dataset $\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\cdots,(\mathbf{x}_t,y_t)\}$, $\mathbf{x}_t\in R^d$ indicates an input variable that has a relationship with the target value y_t, $t=1,\ldots,n$ is a measure of training data, and $\mathbf{s}_t$ is a fuzzy vector that represents the membership degree of the sample $\mathbf{x}_t$ of each class with a fuzzy membership value is $0\le s_t\le 1$. The fuzzy membership function according to Eq. (1).

The multiclass time series data classification using FSVM mode:

$$\min_{\mathbf{w}\in\mathbb{R}^{N+1},\xi_t\in\mathbb{R}} \frac{\lambda}{2}\sum_{r=1}^{k}\mathbf{w}_r^T\mathbf{w}_r + \sum_{t=i}^{n}\xi_t \tag{4}$$

with the constrained function

$$\left[\mathbf{w}_{y_t}\mathbf{x}_t-\mathbf{w}_r\mathbf{x}_t\right]s_t\ge 1-\delta_{y_t,r}-\xi_t \tag{5}$$

The optimization problem in Eq (4) solvable using Karush-Kuhn-Tucker (KKT) technique. Lagrange function with KKT procedure, aims to minimize the objective function in Eq. (4) with constraint in Eq. (5). Lagrange function as follows

$$L(\mathbf{w},\xi,\alpha) = \frac{\lambda}{2}\sum_{r=1}^{k}\mathbf{w}_r^T\mathbf{w}_r + \sum_{t=i}^{n}\xi_t - \sum_{t,r}\alpha_{t,r}\left(\mathbf{w}_{y_t}\mathbf{x}_t s_t\right) + \sum_{t,r}\alpha_{t,r}\left(\mathbf{w}_r\mathbf{x}_t s_t\right) - \sum_{t,r}\alpha_{t,r}\left(\delta_{y_t,r}\right) + \sum_{t,r}\alpha_{t,r} - \sum_{t,r}\alpha_{t,r}\,\xi_t \tag{6}$$

The KKT conditions to be fulfilled are

$$\frac{\partial L(\mathbf{w},\xi,\alpha)}{\partial \mathbf{w}_r} = 0 \tag{7}$$

$$\frac{\partial L(\mathbf{w},\xi,\alpha)}{\partial \xi_t} = 0 \tag{8}$$

and hence, Eq. (7) and Eq. (8) become

$$\lambda\sum_{r=1}^{k}\mathbf{w}_r - \sum_{t,y_t=r}\underbrace{\left(\sum_{q}\alpha_{t,q}\right)}_{=1}(\mathbf{x}_t s_t) + \sum_{t,r}\alpha_{t,r}\left(\mathbf{x}_t s_t\right) = 0$$

$$\mathbf{w}_r = \lambda^{-1}\left[\sum_{t,r}\left(\delta_{y_t,r} - \alpha_{t,r}\right)\mathbf{x}_t s_t\right] \tag{9}$$

$$\sum_{t,r}\alpha_{t,r} = 1 \tag{10}$$

by substituting the partial derivatives, namely Eq. (9) and (10), into the Lagrange function Eq. (6) the dual optimization is obtained a s follows:

$$\max_{\alpha} L(\mathbf{w},\xi,\alpha) = \frac{\lambda}{2}\sum_{r=1}^{k}\mathbf{w}_r^T\mathbf{w}_r - \sum_{t,r}\alpha_{t,r}\left(\mathbf{w}_{y_t}\mathbf{x}_t s_t\right) + \sum_{t,r}\alpha_{t,r}\left(\mathbf{w}_r\mathbf{x}_t s_t\right) + \sum_{t,r}\alpha_{t,r}\left(1-\delta_{y_t,r}\right) \tag{11}$$

by defining, then Eq. (11) becomes

$$\max_{\alpha} L(\mathbf{w},\xi,\alpha) = \overbrace{\frac{\lambda}{2}\sum_{r=1}^{k}\mathbf{w}_r^T\mathbf{w}_r}^{\overset{def}{=}Q1} - \overbrace{\sum_{t,r}\alpha_{t,r}\left(\mathbf{w}_{y_t}\mathbf{x}_t s_t\right)}^{\overset{def}{=}Q2} + \overbrace{\sum_{t,r}\alpha_{t,r}\left(\mathbf{w}_r\mathbf{x}_t s_t\right)}^{\overset{def}{=}Q3} + \sum_{t,r}\alpha_{t,r}\,e_{t,r} \tag{12}$$

By substituting Eq. (9) into the above Eq.(12), the following Eq. (12) is obtained;

$$Q1 = \frac{1}{2}\lambda^{-1}\sum_{t(i),t(j)}\left(\mathbf{x}_{t(i)}s_{t(i)}\right)\cdot\left(\mathbf{x}_{t(j)}s_{t(j)}\right)\sum_{r}\left(\delta_{y_{t(i)},r} - \alpha_{t(i),r}\right)\left(\delta_{y_{t(j)},r} - \alpha_{t(j),r}\right) \tag{13}$$

$$Q2 = \lambda^{-1}\sum_{t(i),t(j)}\left[\left(\mathbf{x}_{t(i)}s_{t(i)}\right)\cdot\left(\mathbf{x}_{t(j)}s_{t(j)}\right)\right]\sum_{r}\delta_{y_{t(i)},r}\left(\delta_{y_{t(j)},r} - \alpha_{t(j),r}\right) \tag{14}$$

$$Q3 = \lambda^{-1}\sum_{t(i),t(j)}\left(\mathbf{x}_{t(i)}s_{t(i)}\right)\cdot\left(\mathbf{x}_{t(j)}s_{t(j)}\right)\sum_{r}\alpha_{t(i),r}\left(\delta_{y_{t(j)},r} - \alpha_{t(j),r}\right) \tag{15}$$

Next, substitute the values of Q1, Q2 and Q3 from Eq. (13), Eq. (14) and Eq. (15) into Eq. (12), and we obtain

$$\max_{\alpha} L(\mathbf{w},\xi,\alpha) = -\frac{1}{2}\lambda^{-1}\sum_{t(i),t(j)}\left(\mathbf{x}_{t(i)}s_{t(i)}\right)\cdot\left(\mathbf{x}_{t(j)}s_{t(j)}\right)\sum_{r}\left(\delta_{y_{t(i)},r} - \alpha_{t(i),r}\right)\left(\delta_{y_{t(j)},r} - \alpha_{t(j),r}\right) + \sum_{t,r}\alpha_{t,r}\,e_{t,r} \tag{16}$$

By defining, I_i it is a vector whose components all contain the value 0 except for the *i*-component, which is equal to 1. Then Eq. (16) above can be rewritten into a dual program in the form of the following Eq. (17)

$$\max_{\alpha} L_D(\mathbf{w},\xi,\alpha) = -\frac{1}{2}\lambda^{-1}\sum_{t(i),t(j)}\left[\left(\mathbf{x}_{t(i)}s_{t(i)}\right)\cdot\left(\mathbf{x}_{t(j)}s_{t(j)}\right)\right]\left[\left(I_{y_{t(i)}}-\alpha_{t(i)}\right)\left(I_{y_{t(j)}}-\alpha_{t(j)}\right)\right]+\sum_{t(i)}\alpha_{t(i)}\,e_{t(i)} \tag{17}$$

with the consrained function $\forall t(i) \quad \alpha_{t(i)} \geq 0$, and $\alpha_{t(i)} \cdot I = 1$.

Because the constraint function in Eq. (17) is linear, this proves that $L_D(\mathbf{w},\xi,\alpha)$ right convex in α. The problem of Eq. (17) above has only one optimal solution using the Quadratic Programming (QP). Also, to further simplify Eq. (17), then defied $\tau_{t(i)} = I_{t(i)} - \alpha_{t(i)}$, that Eq. (9) becomes

$$\hat{\mathbf{w}}_r = \lambda^{-1}\sum_t\left(\mathbf{x}_t s_t\right)\tau_{t,r} \tag{18}$$

and Eq. (17) can be rewritten as the following Eq. (19)

$$\max_{\tau} L(\mathbf{w},\xi,\tau) = -\frac{1}{2}\lambda^{-1}\sum_{t(i),t(j)}\left[\left(\mathbf{x}_{t(i)}s_{t(i)}\right)\cdot\left(\mathbf{x}_{t(j)}s_{t(j)}\right)\right]\left(\tau_{t(i)}\cdot\tau_{t(j)}\right)-\sum_{t(i)}\tau_{t(i)}\,e_{t(i)} \tag{19}$$

with the constrained function $\forall t(i) \quad \tau_{t(i)} \leq I_{y(i)}$ and $\tau_{t(i)} \cdot I = 0$.

Then the classification of the new data can be written to the variable τ as follows

$$\begin{aligned}\hat{f}\left(\mathbf{x}_{(new)}\right) &= \underset{r}{\arg\max}\left(\hat{\mathbf{w}}_r\cdot\mathbf{x}_{(new)}\right)\\ &= \underset{r}{\arg\max}\left\{\sum_t \tau_{t,r}\left[\mathbf{x}\cdot\mathbf{x}_t\right]s_t\right\}\end{aligned} \tag{20}$$

Eq. (20) above can be written in general kernel functions as follows

$$\max_{\tau} L(\mathbf{w},\xi,\tau) = -\frac{1}{2}\lambda^{-1}\sum_{t(i),t(j)}K\left[\left(\mathbf{x}_{t(i)}s_{t(i)}\right)\cdot\left(\mathbf{x}_{t(j)}s_{t(j)}\right)\right]\left(\tau_{t(i)}\cdot\tau_{t(j)}\right)-\sum_{t(i)}\tau_{t(i)}\,e_{t(i)} \tag{21}$$

with $\forall t(i) \quad \tau_{t(i)} \leq I_{y(i)}$ and $\tau_{t(i)} \cdot I = 0$.

The new data classifier can be

$$\hat{f}\left(\mathbf{x}_{(new)}\right) = \underset{r}{\arg\max}\left\{\sum_t \tau_{t,r}\,K\left[\mathbf{x}\cdot\mathbf{x}_t\right]s_t\right\} \tag{22}$$

The purpose of the kernel function is to take data as an input and then convert it into the form that is required. Linear, non-linear, polynomial, radial basis function (RBF) and sigmoid are some of the most commonly used types of kernel functions. The RBF kernel function is used in this study.

$$K\left[\mathbf{x}\cdot\mathbf{x}_t\right]=\exp\left(-\frac{\left\|\mathbf{x}-\mathbf{x}_t\right\|^2}{2\sigma^2}\right) \tag{23}$$

3.2. Evaluation of Model Performance

The actual and predicted data from the classification results are shown in Table 1, the confusion matrix of multiclass data for the 3-class case table. Calculation of average classification accuracy and classification error using the following formula (Sokolova & Lapalme, 2009);

Table 1
Confusion Matrix of multiclass data for the 3-class case

F_{kh}		**Prediction Classes (h)**		
		Class 1	**Class 2**	**Class 3**
Actual Classes (k)	**Class 1**	F_{11}	F_{12}	F_{13}
	Class 2	F_{21}	F_{22}	F_{23}
	Class 3	F_{31}	F_{32}	F_{33}

$$Average\ Accuracy\ of\ \ Classification\ = \frac{1}{k}\left(\sum_{i=1}^{k}\frac{tp_i+tn_i}{tp_i+fn_i+fp_i+tn_i}\right)$$

$$=\frac{1}{3}\left(\sum_{i=1}^{3}\frac{tp_i+tn_i}{tp_i+fn_i+fp_i+tn_i}\right)$$

$$=\frac{1}{3}\left[\left(\frac{tp_1+tn_1}{tp_1+fn_1+fp_1+tn_1}\right)+\left(\frac{tp_2+tn_2}{tp_2+fn_2+fp_2+tn_2}\right)+\left(\frac{tp_3+tn_3}{tp_3+fn_3+fp_3+tn_3}\right)\right] \tag{24}$$

where tp (true positive) is the number of correctly classified positive data, tn (true negative) is the number of correctly classified negative data. For fp (false positive) is the amount of positive data that is misclassified, and tn (false negative) is the amount of negative data that is misclassified. For multi-class cases, tp, tn, fp, and tn will be calculated for each class. For example, when we calculate for class 1 then the formula used is $tp_1 = F_{11}$, $tn_1 = F_{22} + F_{23} + F_{32} + F_{33}$, $fp_1 = F_{21} + F_{31}$, and $fn_1 = F_{12} + F_{13}$. Furthermore, the same is true for the other classes 2 and 3.

The Receiver Operating Characteristic (ROC) curve is a curve that shows the false positive rate and true positive rate. Area Under Curve (AUC) is the area under curve, which can also be an indication of the accuracy of the prediction model. Horng (2009) proposed an approach to calculate the AUC of a multiclass classification using Eq. (25).

$$AUC_{average} = \frac{1}{number\ of\ binary\ SVM}\sum_{c_i,c_j\in C} AUC(c_i,c_j) \tag{25}$$

4. Simulation

4.1. Data Description

In this simulation, the generated process is restricted to AR(2). Furthermore, the data used consists of two data schemes with 504 data each. Table 2 summarizes description of generated data for simulation study, where Y consists of 3 classes for each simulation.

Table 2
Data description and class size comparison for Y data

Simulation Data Schema	Total Data	Y		
		Class 1 (0)	Class 2 (1)	Class 3 (2)
Data 1	504	72 (14,29%)	183 (36,31%)	249 (49,40%)
Data 2	504	27 (5,36%)	234 (46,43%)	243 (48,21%)

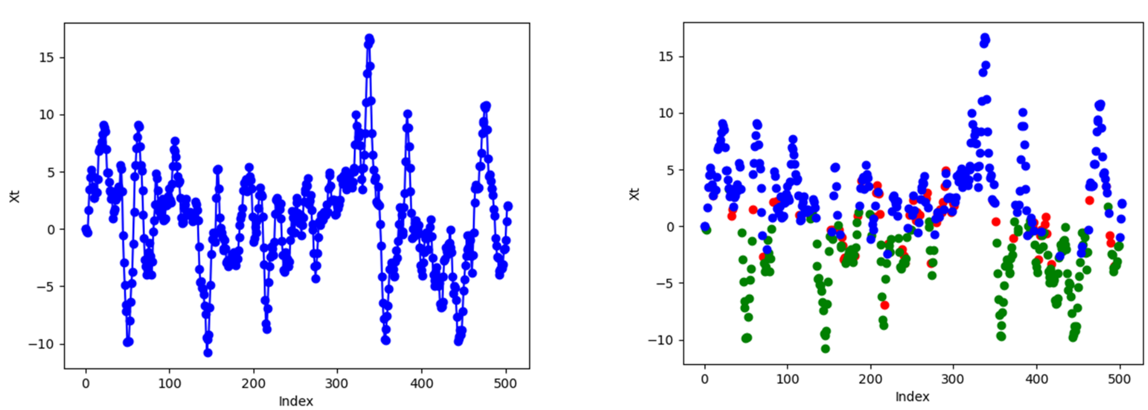


Fig. 2. Plot of data 1

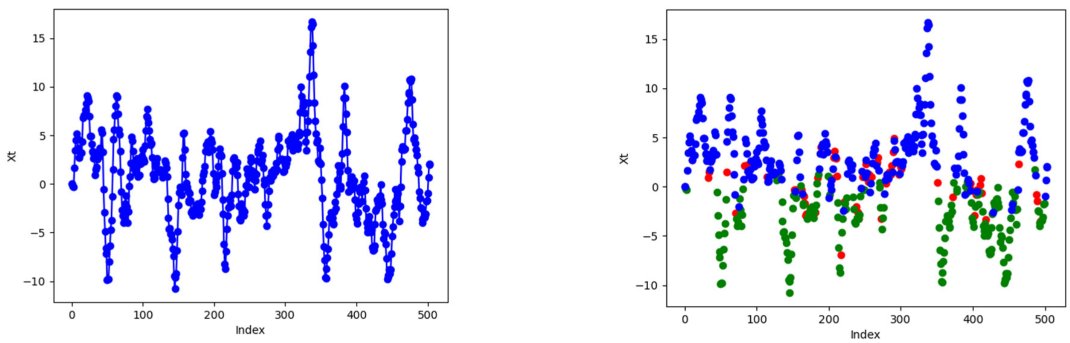


Fig. 3. Plot of data 2

Description of the simulation data used, presented in Table 2. From Table 2, for data 1, the comparison of the amount of data for each class is class 1 as many as 72 (14.29%), class 2 as many as 183 (36.31%) and class 3 as many as 249 (49.40%). As for data 2, the comparison is that class 1 is 27 (5.36%), class 2 is 234 (46.43%) and class 3 is 243 (48.21%). In fig. 2 and fig. 3 there are two plots which are (1) a plot of X data and (2) a plot between X and Y data, where each class of Y data is mapped with a different color.

4.2. Steps of simulation

The simulation steps to assess the accuracy of the FSVM model for classifying time series data with multiclass problems are described as follows;

a) generate time series data, AR(p) with p = 2. The stages are as follows;
   1) generate n=504 data from ε_t that is normally distributed with $\mu = 0$ dan σ^2=25
   2) determine the value of ϕ_1 and ϕ_2 which must fulfil the stationary condition as follows

      $-1 < \phi_2 < 1$

      $-2 < \phi_1 < 2$

      where $(\phi_2 + \phi_1) < 1$ and $\phi_1 - \phi_2 < 1$. So the values of ϕ_1 and ϕ_2 that will be used are ϕ_1= 1,5 and ϕ_2 = -0,56.
   3) calculate the X_t value with

      $X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \varepsilon_t$
   4) generate Y_t with multinomial function.

b) determine the fuzzy membership function s_t using Eq. (1).

c) split data into training and testing data.

d) selecting the kernel function with kernel parameter and λ parameter.

e) selecting the best combination of kernel parameter and cost parameters from training data optimization for testing data classification using the grid search method.

f) classify the new data using the following classification functions

$$\hat{f}\left(\mathbf{x}_{(new)}\right) = \underset{r}{\arg\max} \left\{ \sum_t \tau_{t,r} K[\mathbf{x} \cdot \mathbf{x}_t] s_t \right\}$$

g) determine the accuracy of classification from the accuracy value according to Eq. (24) and the AUC of a multiclass classification using Eq. (25).

5. Simulation Results

In this section, data classification is carried out using SVM and FSVM for two simulated data. The composition of training and testing dataset is examined under four schemes e.g., a) 90%:10%, b) 80%: 20%, c) 70%:30% and d) 60%:40%. Furthermore, we used the radial basis function (RBF) kernel, with kernel parameters σ = 1, 10, and 100, and parameter λ = 1, 10, and 100. Table 3 and Table 4 summarise the classification accuracy results using SVM and FSVM, respectively.

Table 3, for data 1, the compositions of training : testing data of 90%:10% and 80%:20% have the same highest accuracy value at λ = 1 and σ = 1 which is 81,19%. Meanwhile using 70%:30%, the highest accuracy value is obtained for λ = 1 and σ = 1 with the value of 80,13%. With 60%: 40% has the highest accuracy value 85,15% obtained at λ = 100 and σ = 1. For data 2, with the composition of training: testing data 90%:10%, 80%:20% and 70%:30% have the same highest accuracy value of 82% at the combination of λ = 1, 10 and 100 with σ = 1 and 10. Meanwhile, using 60%:10%, the highest accuracy value is 90,59% at the combination of λ = 1, 10 and 100 with σ = 1.

Table 3
Testing accuracy value (%) of simulation data using SVM

Simulation Data Schema	Training : Testing (%)	λ	σ		
			1	10	100
Data 1	90 : 10	1	81.19	79.21	80.2
		10	79.21	75.25	73.27
		100	79.21	76.24	75.25
	80 : 20	1	81.19	79.21	80.2
		10	79.21	74.26	73.27
		100	79.21	77.23	74.26
	70 : 30	1	80.13	76.82	78.15
		10	79.47	77.48	72.85
		100	78.81	77.48	73.51
	60 : 40	1	83.66	82.18	77.72
		10	83.66	81.68	76.73
		100	85.15	79.21	77.72
Data 2	90 :10	1	82	82	80
		10	82	82	80
		100	82	82	80
	80 : 20	1	82	82	80
		10	82	82	80
		100	82	82	80
	70 : 30	1	82	82	80
		10	82	82	80
		100	82	82	80
	60 : 40	1	90.59	88.61	86.14
		10	90.59	88.61	86.14
		100	90.59	88.61	86.14

From Table 3, for data 1, the compositions of training : testing data of 90%:10% and 80%:20% have the same highest accuracy value at λ = 1 and σ = 1 which is 81,19%. Meanwhile using 70%:30%, the highest accuracy value is obtained for λ = 1 and σ = 1 with the value of 80,13%. With 60%: 40% has the highest accuracy value 85,15% obtained at λ = 100 and σ = 1. For data 2, with the composition of training : testing data 90%:10%, 80%:20% and 70%:30% have the same highest accuracy value of 82% at the combination of λ = 1, 10 and 100 with σ = 1 and 10. Meanwhile, using 60%:10%, the highest accuracy value is 90,59% at the combination of λ = 1, 10 and 100 with σ = 1.

Table 4
Testing accuracy value (%) of simulation data using FSVM

Simulation Data Schema	Training : Testing (%)	λ	σ		
			1	10	100
Data 1	90 : 10	1	81.19	81.19	79.21
		10	82.18	80.2	74.26
		100	81.19	77.23	74.26
	80 : 20	1	81.19	82.18	79.21
		10	81.19	79.21	74.26
		100	79.21	76.26	74.26
	70 : 30	1	80.79	78.81	74.17
		10	80.13	77.48	70.2
		100	78.81	77.48	73.51
	60 : 40	1	85.15	83.17	79.21
		10	84.65	82.18	73.76
		100	84.16	82.18	73.27
Data 2	90 : 10	1	90	90	90
		10	88	90	90
		100	88	90	90
	80 : 20	1	90	90	90
		10	88	90	90
		100	88	90	90
	70 : 30	1	90	90	90
		10	88	90	90
		100	88	90	90
	60 : 40	1	94.55	93.07	86.63
		10	92.08	93.07	86.63

100	93.56	93.07	86.63

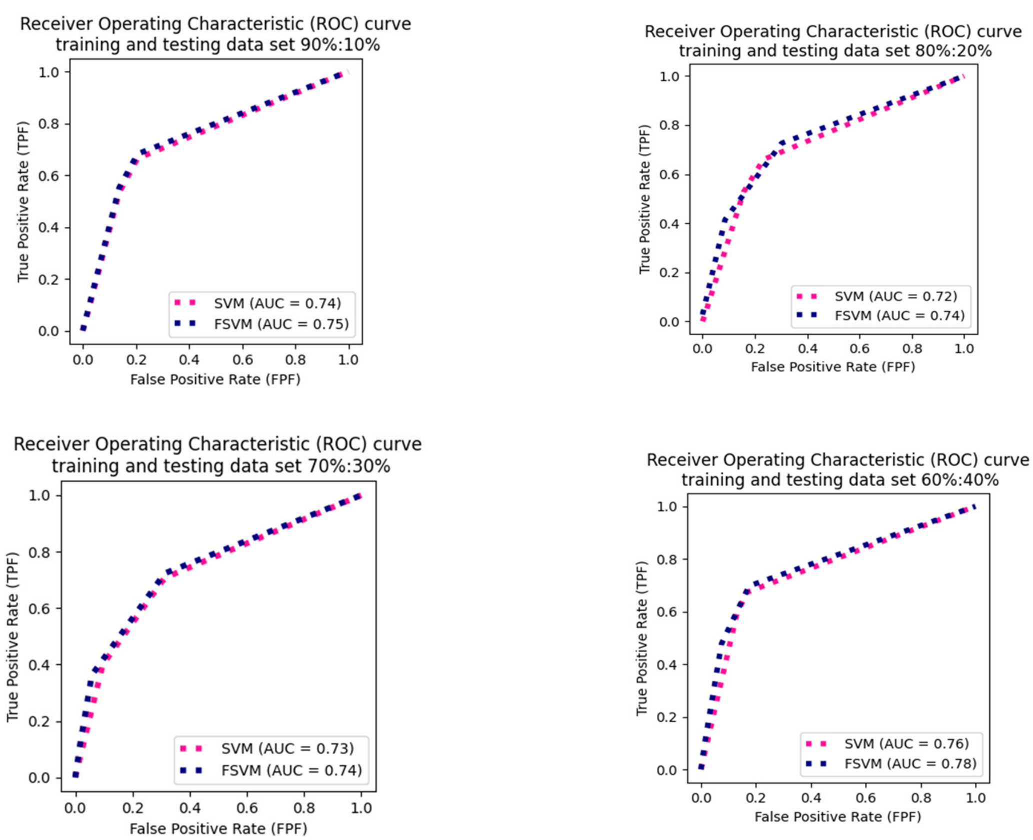
From Table 4, for data 1, the 90%:10% training : testing data composition have the higher accuracy with a value of 82,18% at λ = 10 and σ = 1. With 80%:20% has the highest accuracy value at λ = 1 and σ = 10 which is 82,18%. Meanwhile, using 70%:30%, the highest accuracy value is 80,79% at λ=1 and σ=1. And with 60%: 40% has the highest accuracy value obtained at λ = 1 and σ = 1 which is 85,15%. For data 2, with the composition of training : testing data 90%:10%, 80%:20% and 70%:30% have the same highest accuracy value of 90% obtained at λ=1 and σ=1 and at the combination of λ=1, 10 and 100 with σ=10 and 100. Meanwhile, using 60%:10%, the highest accuracy value is obtained at λ=1 and σ=1 which is 94,55%.

Table 5 summarizes the highest accuracy values of classification using SVM and FSVM for all simulated data schemes. This table shows that classification of simulated data using the developed FSVM method has a better accuracy value than classification using the SVM method. The accuracy value of FSVM classification for all divisions of training and testing data of both simulated data schemes is better than classification with SVM.

Table 5

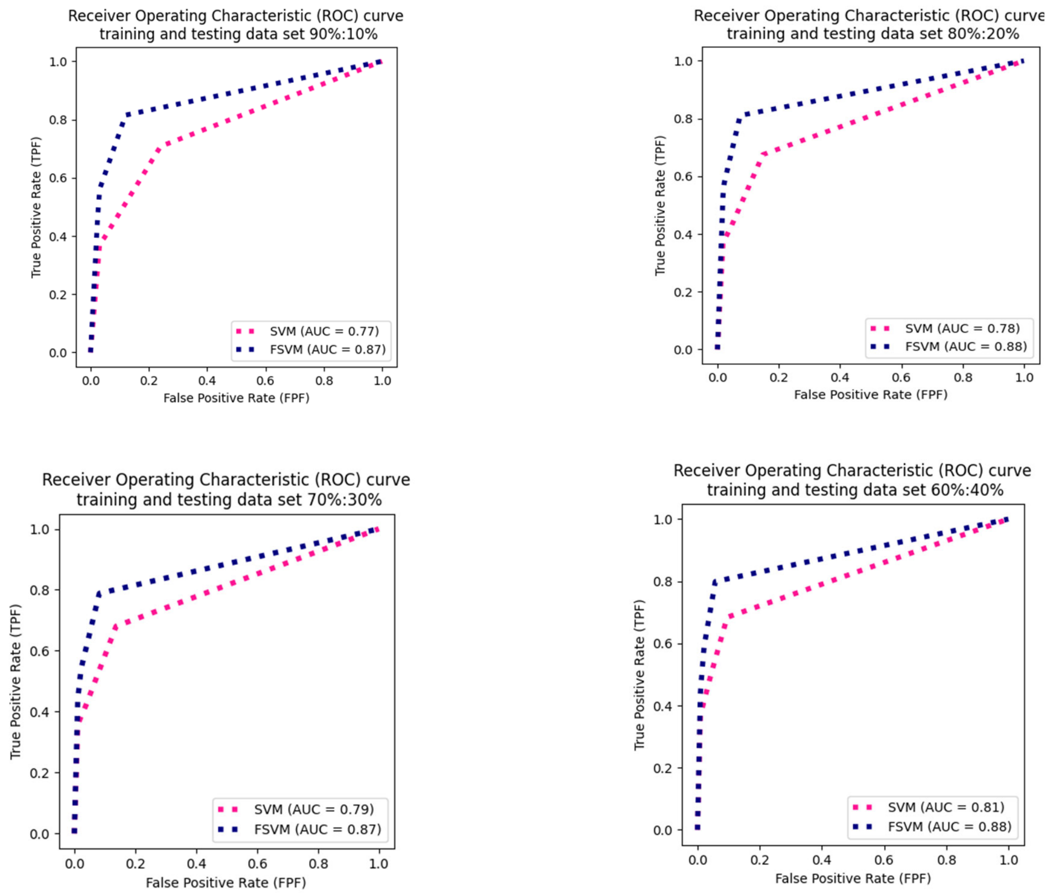
The highest testing accuracy value of classification using SVM and FSVM (%)

Simulation Data Schema	Training : Testing (%)	Class Training (%)			Class Testing (%)			λ	σ	SVM	FSVM
		0	1	2	0	1	2				
Data 1	90 :10	15	37	49	12	32	56	10	1	79.21	**82.18**
	80 : 20	15	31	54	12	57	31	1	10	79.21	**82.18**
	70 : 30	16	27	57	11	58	31	1	1	80.13	**80.79**
	60 : 40	18	31	51	9	44	47	1	1	83.66	**85.15**
Data 2	90 : 10	4	46	50	14	50	36	1	1	82	**90**
	80 : 20	4	46	49	9	47	45	1	1	82	**90**
	70 : 30	4	44	51	7	52	41	1	1	82	**90**
	60 : 40	5	45	50	6	48	46	1	1	90.59	**94.55**



Fiq. 4. The ROC Curves for Data 1

The ROC curves for each simulated data of SVM and FSVM classification results are presented in Fig. 4 for data 1 and Fig. 5 for data 2. From Fig. 4 and Fig. 5, the results obtained from all training : testing data division show that the FSVM graph is closer to the point (0,1) so it can be concluded that the classification performance with FSVM is better than SVM. And for the AUC value of each ROC curve presented in Table 6. From Table 6 it can be concluded that the AUC value of the classification results using FSVM for all divisions of training data: testing data from both simulated data schemes is better than using SVM.



Fiq. 5. The ROC Curves for Data 2

Table 6
AUC value of the ROC curve of classification using SVM and FSVM

Simulation Data Schema	Training : Testing (%)	Class Training (%)			Class Testing (%)			λ	σ	SVM	FSVM
		0	1	2	0	1	2				
Data 1	90 :10	15	37	49	12	32	56	10	1	0.74	**0.75**
	80 : 20	15	31	54	12	57	31	1	10	0.72	**0.74**
	70 : 30	16	27	57	11	58	31	1	1	0.73	**0.74**
	60 : 40	18	31	51	9	44	47	1	1	0.76	**0.78**
Data 2	90 : 10	4	46	50	14	50	36	1	1	0.77	**0.87**
	80 : 20	4	46	49	9	47	45	1	1	0.78	**0.88**
	70 : 30	4	44	51	7	52	41	1	1	0.79	**0.87**
	60 : 40	5	45	50	6	48	46	1	1	0.81	**0.88**

6. Conclusions

The proposed FSVM method for classification of time series data, developed from the crammer singer SVM by adding fuzzy to the constraint function. We have shown a step by step procedure to obtain the parameter estimation by Lagrange Multipliers method and solution of optimization problem using Karush-Kuhn-Tucker method. The application of this method on simulated data gives results that are in accordance with our expectations and desires. This is evident from the accuracy value of simulated data classification using FSVM is better than SVM. And in the imbalanced cases, where data 2 is a more extreme imbalanced case than data 1. The results of the classification accuracy value and AUC value show that the classification performance with FSVM on data 2 is better than FSVM on data 1. So it can be concluded that the developed FSVM can classify extreme imbalanced data better than SVM.

Acknowledgments

All authors are grateful to the editor and reviewers for improving this paper through their critiques and suggestions.

References

Abe, S., & Inoue, T. (2002, April). Fuzzy support vector machines for multiclass problems. In ESANN (pp. 113-118).

Batuwita, R., & Palade, V. (2010). FSVM-CIL: fuzzy support vector machines for class imbalance learning. *IEEE Transactions on Fuzzy Systems*, *18*(3), 558-571.

Bostrom, A., & Bagnall, A. (2017). Binary shapelet transform for multiclass time series classification. *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXII: Special Issue on Big Data Analytics and Knowledge Discovery*, 24-46.

Crammer, K., & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research, 2*(Dec), 265-292.

Fan, Q., Wang, Z., Li, D., Gao, D., & Zha, H. (2017). Entropy-based fuzzy support vector machine for imbalanced datasets. *Knowledge-Based Systems*, *115*, 87-99.

Farquad, M. A. H., & Bose, I. (2012). Preprocessing Imbalanced data using support vector machine. *Decision Support Systems, 53*(1), 226-233.

Hills, J., Lines, J., Baranauskas, E., Mapp, J., & Bagnall, A. (2014). Classification of time series by shapelet transformation. *Data mining and knowledge discovery*, *28*, 851-881.

Horng, M. H. (2009). Multi-class support vector machine for classification of the ultrasonic images of supraspinatus. *Expert Systems with Applications*, *36*(4), 8124-8133.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2019). Deep learning for time series classification: a review. *Data mining and knowledge discovery*, *33*(4), 917-963.

Jeong, Y. S., Jeong, M. K., & Omitaomu, O. A. (2011). Weighted dynamic time warping for time series classification. *Pattern recognition*, *44*(9), 2231-2240.

Kreßel, U. G. (1999). Pairwise classification and support vector machines. Advances in kernel methods: support vector learning, 255-268.

Lee, T. S., Chiu, C. C., Chou, Y. C., & Lu, C. J. (2006). Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. *Computational Statistics & Data Analysis, 50*(4), 1113-1130.

Lin, C. F., & Wang, S. D. (2002). Fuzzy support vector machines. *IEEE transactions on neural networks, 13*(2), 464-471.

Lines, J., Davis, L. M., Hills, J., & Bagnall, A. (2012, August). A shapelet transform for time series classification. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 289-297).

Mandal, S. N., Choudhury, J. P., & Chaudhuri, S. B. (2012). In search of suitable fuzzy membership function in prediction of time series data*. International Journal of Computer Science Issues, 9*(3), 293-302.

Mohammadi, M., & Sarmad, M. (2019). Robustified distance based fuzzy membership function for support vector machine classification. *Iranian Journal of Fuzzy Systems*, *16*(6), 191-204.

Sain, H., & Purnami, S. W. (2015). Combine sampling support vector machine for imbalanced data classification. *Procedia Computer Science*, *72*, 59-66.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, *45*(4), 427-437.

Tsang, E. C., Yeung, D. S., & Chan, P. P. (2003, November). Fuzzy support vector machines for solving two-class problems. *In Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03EX693) (Vol. 2, pp. 1080-1083). IEEE.*

Vapnik, V. N. (1995). *The nature of statistical learning theory*. 840 Springer-Verlag New York. Inc., New York, NY, USA, 841, 842.

Weston, J., & Watkins, C. (1999, April). Support vector machines for multi-class pattern recognition. In Esann (Vol. 99, pp. 219-224).

Wu, Y., Shen, L., & Zhang, S. (2017, May). Fuzzy multiclass support vector machines for unbalanced data. In *2017 29th Chinese Control And Decision Conference (CCDC)* (pp. 2227-2231). IEEE.

Xue, X., Yang, X., & Chen, X. (2014). Application of a support vector machine for prediction of slope stability. *Science China Technological Sciences, 57*, 2379-2386.

Zadeh, L. A. (1965). Fuzzy sets. *Information and control, 8*(3), 338-353.

Zhang, D., Zuo, W., Zhang, D., & Zhang, H. (2010, August). Time series classification using support vector machine with Gaussian elastic metric kernel. *In 2010 20th International Conference on Pattern Recognition (pp. 29-32). IEEE.*