ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DETECTION OF STRUCTURAL DAMAGE USING ONE-CLASS SUPPORT VECTOR MACHINE

M.Sc. THESIS

Efkan OK

Department of Civil Engineering

Structure Engineering Programme

June 2020

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DETECTION OF STRUCTURAL DAMAGE USING ONE-CLASS SUPPORT VECTOR MACHINE

M.Sc. THESIS

Efkan OK
(501161060)

Department of Civil Engineering

Structure Engineering Programme

Thesis Advisor: Dr. Burcu GÜNEŞ

June 2020

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

TEK-SINIF DESTEK VEKTÖRLERİ KULLANARAK YAPISAL HASAR TESPİTİ

YÜKSEK LİSANS TEZİ

Efkan OK
(501161060)

İnşaat Mühendisliği Anabilim Dalı

Yapı Mühendisliği Programı

Tez Danışmanı: Dr. Öğr. Üyesi Burcu GÜNEŞ

Haziran 2020

Efkan Ok, a M.Sc. student of İTU Graduate School of Science Engineering and Technology student ID 501161060, successfully defended the thesis/dissertation entitled “Detection of Structural Damage using One-Class Support Vector Machine”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Dr. Burcu GÜNEŞ** Istanbul Technical University	.....
Jury Members :	**Assoc.Prof.Dr. B.Özden ÇAĞLAYAN** Istanbul Technical University	.....
	Dr. Çağrı Mollamahmutoğlu Yildiz Technical University	.....

Date of Submission : 07 May 2020
Date of Defense : 03 June 2020

To my family,

FOREWORD

I would like to thank my advisor Dr. Burcu GÜNEŞ for her invaluable guidance and support. I would also like to thank my sister and colleague Tuba OK İŞKARA for her help.

June 2020

Efkan OK
(Civil Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

ARX **:** Autoregressive Model with Exogeneus Input
ANN **:** Artificial Neural Network
ERA **:** Eigensystem Realization Algorithm
OC-SVM **:** One-Class Support Vector Machine
RBF **:** Radial Basis Function
SHM **:** Support Health Monitoring
SNR **:** Signal Noise Ratio
SVDD **:** Support Vector Data Description
SVM **:** Support Vector Machine

SYMBOLS

$\boldsymbol{A, B, C, D}$	**:** Resulting Matrices of the Realization
$\boldsymbol{D, CB, CAB \ldots CA^{s-2}B}$	**:** Markov Parameters
$\boldsymbol{H( )}$	**:** Hankel Matrix
$\boldsymbol{I}_{\text{i}}$	**:** Identity Matrix
$\boldsymbol{K}(\boldsymbol{x_i}, \boldsymbol{x_j})$	**:** Kernel Function
$\boldsymbol{L( )}$	**:** Langrage
σ	**:** Gaussian Kernel Width
$\boldsymbol{\varphi}$	**:** Mapping Function
$\boldsymbol{\theta}$	**:** Coefficient Matrices
$\boldsymbol{\Lambda}$	**:** Eigenvalues
$\boldsymbol{\Psi}$	**:** Eigenvector
$\boldsymbol{0_i}$	**:** Null Matrix
$\boldsymbol{V_o}$	**:** Input Data
$\boldsymbol{u}$	**:** Input Vector
$\boldsymbol{k}$	**:** Time Step
$\boldsymbol{p}$	**:** Order of Model
$\boldsymbol{w}$	**:** Natural frequencies
$\boldsymbol{x}$	**:** State Vector
$\boldsymbol{x_i}$	**:** Support Vector
$\boldsymbol{y}$	**:** Output Vector
$\boldsymbol{y_o}$	**:** Output Data

LIST OF TABLES

Page

LIST OF FIGURES

Page

DETECTION OF STRUCTURAL DAMAGE USING ONE-CLASS SUPPORT VECTOR MACHINE

SUMMARY

Support vector machines (SVMs) are a set of supervised machine learning methods that have recently been applied for structural damage detection. In this thesis, the one-class SVM that requires data only from the undamaged state of the structure for which the damaged data is not available a priori is utilized to train a model for damage detection. The proposed one-class kernel classifier utilizes the autoregressive model with exogeneous input (ARX) parameters and modal frequencies of the intact structure as the training feature to detect structural damage. The trained classifier can then be used with the data obtained from the same structure at different states to classify the structure as damaged or undamaged. The proposed methodology is evaluated using simulated acceleration data of one node from each floor via a finite element model of three-story moment resisting frame under various damage cases. In this study the features selected as damage sensitive feature are ARX model coefficients of related nodes, residual mean errors between prediction accelerations and accelerations obtained from finite element model of related nodes, standard deviations of residual errors and natural frequencies of system. The acceleration signals obtained from sensor locations are processed using Eigensystem Realization algorithm to extract the modal properties of the structure. Reduction in elastic modulus of material and reduction in rigidity of six different connections of structural elements are employed as damage case. The results show that success of model in damage detection varies depending on structural feature used and damage case. While highest damage detection ratios were obtained from SVM-model created with modal frequencies, the performance of SVM-model created with ARX model parameters is also satisfactory.

TEK-SINIFLI DESTEK VEKTÖRLERİ KULLANARAK YAPISAL HASAR TESPİTİ

ÖZET

Yapılar belirli bir kullanım süresi gözetilerek tasarlanırlar. Bu kullanım süreleri boyunca da deprem, nem, rüzgar gibi dış etkilere maruz kalırlar. Bu dış etkilerden veya tasarım aşamasında göz önünde bulundurulan sürekli yüklerin yarattığı yorulma gibi etkilerden dolayı bir süre sonra kullanılmaları tehlikeli hale gelir. Bu tehlikeli halin ne zaman olduğunu tam olarak bulmak için de yapının tepkileri ölçülerek durumu analiz edilir. Bu işleme ise yapı sağlığı izleme adı verilmiştir. Asıl olarak yapının dinamik verilerinin takibi ve analizi anlamına gelen yapı sağlığı izleme yapının eğimi, çatlak durumu veya yerdeğiştirmesi gibi statik etkilerinin takibini de kapsamaktadır. Dinamik özelliklerin değişiminden yola çıkarak yapıda meydana gelmiş ya da gelebilecek olan hasarın tespitini, lokalizasyonunu ve şiddetini ölçmeyi amaçlar.

Makine öğrenme yöntemleri ise yapıdan toplanan dinamik verilerin analiz aşamasında kullanılır. Bu yöntemlerin en büyük avantajı oluşturulan analiz modelinin sadece yapıdan toplanan dataya odaklı olması ve her yapı için ayrı bir analiz modeli gereksinimini ortadan kaldırmasıdır. Yapay sinir ağı ve destek vektör makineleri yöntemleri bu alanda en çok kullanılan makine öğrenme yöntemleridir. Bu çalışmada kullanılan destek vektör makineleri ilk olarak çoklu sınıflandırma için geliştirilmiş bir algoritmadır. Tek-sınıflı destek vektör makineleri ise bu yöntemin özel bir halidir. Bu yöntemle tek bir duruma ait verilerle sınıflandırıcı bir model oluşturularak yapıdaki beklenmedik durumların tespiti yapılabilir.

Bu tezde, tek-sınıflı destek vektörleri ile hasar tespiti için geliştirilen modelde yapının sadece hasar görmemiş durumundan gelen veriler kullanılmıştır. Önerilen bu tek-sınıflı kernel sınıflandırıcı modeli, yapısal hasar tespiti için hasarsız yapının ARX (autoregressive model with exogeneous input) parametrelerini ve modal frekanslarını kullanmaktadır. ARX parametreleri herhangi bir sistemde girdiler ve çıktılar arasındaki lineer bağıntıyı ifade etmede kullanılan katsayılardır. Bu çalışmada, yapıya etkitilen yer hareketi ivme kaydı ile bunun sonucunda yapının düğüm noktalarında oluşan ivmeler arasındaki bağıntının katsayıları ARX parametreleri olarak kullanılmıştır. Modal frekansları ise, yapıdan elde edilen ivme verilerinden Özdeğer Sistem Gerçekleştirme Algoritması (Eigensystem Realization Algorithm - ERA) kullanılarak elde edilmiştir. ERA temel olarak yapıya etkiyen girdi verilerini ve yapının tepkisini oluşturan çıktı verilerini kullanarak oluşturduğu numerik modeli kullanarak modal tanımlama işlemidir.

Bu çalışmada numerik veriler elde etmek için sonlu elemanlar yöntemi ile oluşturulmuş iki açıklıklı, üç katlı ve iki boyutlu bir yapı kullanılmıştır. İlk olarak yapının hasarsız durumuna bir deprem yer hareketi etkitilerek her kata ait bir düğüm noktasından ivme kayıtları toplanmıştır. Seçilen deprem yer hareketi ölçeklendirilerek yapının lineer kalması hedeflenmiştir. Elde edilen düğüm noktaları ivmelerinden toplamda altı farklı hasara duyarlı özellik çıkarılmıştır. Bunlar, ilgili düğüm noktasına ait ARX model katsayıları, ilgili düğüm noktası için tahmin edilen ivme değerleri ile sonlu eleman modelinden alınan ivme değerleri arasındaki ortalama hata değeri, hata

değerlerinin standart sapmaları ve yapının doğal frekanslarıdır. Hata değerleri ve hata değerlerinin standart sapmalarının bulunması amacı ile yapıya başka bir deprem yer hareketi de etkitilmiştir. Destek vektör makineleri yöntemi ile sınıflandırıcı bir model oluşturabilmek için elde edilen hasara duyarlı özelliklerin çeşitlendirilmesi gerekmektedir. Sonlu eleman modelinden saptanan ivme kayıtlarına hem çıkarılan özelliklerin bir model oluşturabilmesi hem de sahada yapılan uygulamalara yakınsaması için farklı seviyelerde rastgele gürültü eklenmiştir. Bu sayede her bir hasara duyarlı özellik grubundan eklenen gürültü seviyesine göre birbirinden farklılaşmış 300 simülasyon elde edilmiştir. Elde edilen bu simülasyon verileri MATLAB programı yardımıyla altı farklı tek-sınıflı kernel sınıflandırıcı modeli oluşturmak için kullanılmıştır.

Önerilen metodoloji çeşitli hasar senaryoları altında farklı yapısal özelliklerin hasar tespitindeki başarısının kıyaslanmasını içerir. Buna uygun olarak, hasar durumlarını temsil etmek için bir global hasar ve altı lokal hasar olmak üzere toplam 7 adet hasar senaryosuna ait ivme verileri sonlu elemanlar yöntemiyle bulunmuştur. Hasarlı durum senaryoları oluşturmak için yapıda kullanılan malzemenin elastisite modülünde ve altı farklı eleman birleşiminin rijitliğinde azaltma yapılmıştır. Altı farklı birleşimin ilk üç tanesi kiriş-kolon bağlantısı ile diğerleri kolon-kiriş bağlantısı ile ilgilidir. Elastisite modülünde azaltım işlemi her adımda %1 uygulanarak on dokuzuncu adımda başlangıç elastisite modülünün %80'ine ulaşıldığında tamamlanmıştır. Birleşimin rijitliğinde azaltım işlemi ise her adımda %10 uygulanarak onuncu adımda bağlantı moment aktaramayacak duruma geldiğinde tamamlanmıştır. Tüm hasarlı durumlara ait ivme verilerine de gürültü eklenerek altı farklı hasara duyarlı özellik grubuna ait üç yüzer adet simülasyon verisi elde edilmiştir. Hasarlı duruma ait bu veriler, ilgili sınıflandırıcı model yardımıyla hasarlı veya hasarsız olarak sınıflandırılarak modelin başarısı ölçülmeye çalışılmıştır.

Sonuçlar bize gösteriyor ki sınıflandırıcı modelin hasar tespitindeki başarısı kullanılan yapısal özelliğe ve yapıda meydana gelen hasar tipine göre değişmektedir. En alt kattaki düğüm noktası ivmeleri kullanılarak hesaplanan ARX katsayıları ile oluşturulan model kendisine yakın kolon bağlantısının rijitliğinin azaltılması hariç diğer hasar tiplerini tespit etmede başarılı olamamıştır. Hasara duyarlı özellik olarak orta kattaki düğüm noktaları ivme değerlerinden elde edilen ARX katsayılarının kullanıldığı model ise hasar tespitinde daha başarılı olmuştur. Bu model ile elastisite modülündeki kademeli azalmanın son adımında elde edilen verilerin yaklaşık %70'i hasarlı olarak sınıflandırılabilmiştir. Ek olarak, modelde kullanılan ivmelerin alındığı düğüm noktasına yakın kolonun rijitliğinin azaldığı hasar tipi de yüksek doğruluk oranıyla tespit edilebilmiştir. ARX katsayıları kullanılarak oluşturulan modellerden en başarılısı en üst katın ivme verilerine ait parametrelerin kullanıldığı modeldir. Bu model yardımıyla en alt kattaki kolon bağlantısının rijitliğinin azaltıldığı durum hariç diğer tüm hasar tipleri %80 üzerinde başarı oranıyla tespit edilebilmiştir. Ortalama hata değerleri ve standart sapmanın kullanıldığı sınıflandırıcı modeller ise her ne kadar elastisite modülünün azalması ve kolon-kiriş birleşiminin rijitliğinin azaltılması durumlarında umut verici sonuçlar verse de kiriş-kolon birleşim rijitliğinin azaltıldığı durumda sağlıklı sonuçlar vermediği için güvenilir bir hasar tespit özelliği olarak kabul edilmemiştir. Modal frekansların hasar durumlarının son adımında tüm verileri doğru şekilde sınıflandırdığı ve hasar tespitinde kullanılabilecek en başarılı yapısal özellik olduğu görülmüştür.

Sonuçlara genel olarak bakılırsa, her adımda yapıya uygulanan hasar şiddeti ile hasar tespitinin orantılı şekilde artması ve hasardan daha çok etkilenmesi beklenen üst katlara ait özellik gruplarının hasar tespitinde daha başarılı olması, önerilen

metodolojinin mühendislik bilgileriyle uyumlu olduğunu göstermiştir. Yine de kullanılan tüm veriler sonlu elemanlar yöntemi ile sanal ortamda elde edildiğinden çalışmanın laboratuvar ortamında ve tam ölçekli saha deneyleriyle de test edilmesi gerekmektedir.

1. INTRODUCTION

Under operation conditions, structures can be damaged by loads affecting each day or external influences such as earthquakes, wind, humidity. Detection of damage at the earliest time possible, can prevent an unexpected collapse of the structure or can reduced repair costs.

Structural Health Monitoring (SHM) aims to assure the structural safety by providing warning as soon as the condition of the structure deteriorates using data from sensors placed on the structure (Long and Büyüköztürk, 2016). In other words, the situation of structure whether healthy or damaged is obtained with frequent or continuous SHM.

According to Rytter (1993), damage identification is consisting of 4 stages: (1) detect presence of damage, (2) localization of damage, (3) severity of damage, (4) consequence of damage. By monitoring changes in some physical properties of the structure, first two of four stages of damage identification can be determined. To identified other stages, some helpful tools like structural model should be used (Doebling et al., 1998).

Two general classes of damage detection methods: (1) global detection methods that deal with dynamic characteristic of structure (2) local detection methods like visual test, tap test, acoustic test or ultrasonic methods, magnetic field methods, radiographs, eddy-current methods and thermal field methods need to know location of previously known damage. Local methods can be efficient in small structure, large structures require dynamic study with global methods (Frigui et al., 2018).

1.1 Global Damage Detection Methods

Global damage detection methods also referred to as vibration-based damage detection methods investigate change in dynamic properties of structure like natural frequency, mode shape, damping. These methods provide continous and online or offline damage detection to ensure that an structure remains fully operational with no significant damage.

According to Yan et al. (2007), global damage detection methods can be divided into two classes: (1) Traditional methods based on modal properties and (2) Modern type methods using statistical pattern recognition methods.

1.1.1 Traditional damage detection methods

The change in the stiffness of the structure due to any damage affects its dynamic properties. Traditional methods based on these changes such as natural frequency, mode shape and damping require precise measurement to create model of structure.

Damage detection methods based on shift of natural frequency is related to global response of structure. There are some limitations in methods that use frequency variation, such as the need for very precise measurements from the sensors or the structure under examination must have been severely damaged (Doebling et al., 1996). As an example, if damage occurs in one of the column-beam connections of a steel structure, there may be no change in the fundamental modes of the structure. Hence any damage cannot be detected with using frequency shifted methods. Localization of damage with frequency shift can be described as another limitation (Limongelli, 2019).

Mode shape-based methods investigate differences mode shapes before and after damage. According to Fan and Qiao (2011), there are two advantages of using mode shape-based methods compared to frequency-based methods: (1) multiple damage detection can be obtained from mode shape-based methods, (2) sensitivity of environmental effects of mode shaped-based methods are less than the others. However, generate of mode shape of structure from experimental data require much more sensor data from finding natural frequency of structure.

The damping ratio used as another damage detection feature undergoes a change that can be observed due to damage to the structure. However, it is not used in structural health monitoring as much as other dynamic properties because of complexity of its mechanism (Cao et al., 2017). There are two main disadvantages of using damping as damage sensitive feature. One of them is large standard deviation in calculation of damping ratio (William and Salawu, 1997). The difficulty of finding the source of the change in damping with vibration measurements in real buildings is another disadvantage (Rytter, 1993).

1.1.2 Modern damage detection methods

Modern damage detection methods also called as intelligent damage diagnosis methods is trying to build detection model that is less dependent on structural shape that only evaluate measured data. During process of creating model, vibration responses of the structure, such as natural frequency, damping are used as classification feature.

As compared to traditional methods, modern methods have some advantages: (1) using only vibration responses thus reducing dependence of experiments, (2) creating a universal methodology for damage detection, (3) chance to detect small structural damage using better characteristic information (Yan et al., 2007).

One of the intelligent damage diagnosis methods is artificial neural network (ANN) methods. This method that consist of 4 part: input, net function, transfer function and one output is pattern recognition methods between input and output. After the learning process using initial data, ANN build the weight function to use between input value and net function and activation function to use between transfer function and output value. There are many publications in the literature that detect damage using ANN such as Ghaboussi et al. (1991), Kudva (1992), McCormick and Nandi (1997), Samanta et al. (2003), Mehrjoo et al. (2008).

Support vector machine (SVM) algorithm is another modern tool used for damage detection. SVM can be defined as the method of separating data with the help of a subset called as support vectors (Bridgelall, 2017). It classifies the data with using hyperplane created with support vectors. SVM model obtained by training of pre-known data for which class it belongs to can separate unknown data by looking where training data are in the plane. In field of SHM, it can be decided that the current state of structure is damaged or healthy using the model created with data that belongs to healthy state of structure. As data, dynamic features of structure can be used. There are many publications in the literature that detect damage using SVM such as Shimada et al. (2006), HoThu and Mita (2013), Shin et al. (2005), Worden and Lane (2001), Long and Büyüköztürk (2016).

Main differences between SVM and ANN is used risk minimization method. Empirical risk minimization principle that is employed by ANN is based on minimizing the error on the training data (Gunn, 1998). Structural risk minimization principle that is

employed by SVM is based on minimizing the risk functional with respect to both terms, the empirical risk, and the confidence interval (Vapnik, 1995).

1.2 Objective of Thesis

The main objective of this thesis is tested the success of SVM in detecting the situations of structure whether damaged or undamaged. For this purpose, a 2-D structure was used with one major and six minor damage cases. It was tried to determine at what level the damage started to be detected by gradually increasing the damage level. By using three local features and three global feature features of the structure, it was tried to make a comparison in terms of the damage detection ratio. In this way, the response of SVM model in damage detection was measured depending on feature selected and the type of damage.

1.3 Scope of Thesis

In this study, first step is the extraction of the structural features using accelerations from the finite element model. It is tried to explain how this process is done in section 2. This section that started with explaining the relationship between input and output data ended with explaining the method (Eigensystem Realization Algorithm) used to find the dynamic properties of the structure.

Second step is to create an one-class support vector machine(OC-SVM) model using the extracted features from undamaged state of structure. In section 3, the mathematical background of the this SVM method used as classification method tried to describe.

The third step is the classification of damaged or undamaged status data. In the first part of section 4, steps of methodology used in classification was explained. Then, there are discussed results of damage detection for different features and different damage cases. Conclusion of this study are are discussed in the section 5. The performance of damage detection of the model was compared according to used structural feature.

2. FEATURE EXTRACTION

2.1 Introduction

Damage sensitive features are quantities extracted from the structural response data which is correlated with presence of damage. As damage progresses, one expects ideally that these features will also change in a consistent manner.

Modal parameters, more specifically the natural frequencies, auto-regressive model parameters and residual errors are selected as damage identifying features. This chapter discusses the theoretical background required for selected features. First, autoregressive model with exogeneous input (ARX) parameters utilized as damage detection feature will be introduced. Fundamental concepts behind the extracting the modal parameters will be discussed in the context of Eigensystem Realization Algorithm (ERA).

2.2 ARX Model

In time domain, relationship between input and output data can be written as;

$$\begin{aligned} &y(k)+a_1 y(k-1)+\ldots+a_p y(k-p) \\ &=\beta_o u(k)+\beta_1 u(k-1)+\ldots+\beta_p u(k-p) \end{aligned} \tag{2.1}$$

p is indicating order of model, k is indicating time steps. Also, $a_1 \ldots a_p$, and $\beta_1 \ldots \beta_p$ represent coefficient matrices, y is m_{x1} output vector, u is r_{x1} input vector.

This equation is called as ARX model in time domain. And it can be rewrite as;

$$\begin{aligned} &y(k)=-a_1 y(k-1)-\mathrm{L}\ -a_p y(k-p) \\ &+\beta_o u(k)+\beta_1 u(k-1)+\mathrm{L}\ +\beta_p u(k-p) \end{aligned} \tag{2.2}$$

In the practice, coefficient matrices are unknown, but they can be calculated from experimental input and output signals.

The equation 2.2 can be extended;

$$
\begin{aligned}
&y(1)=\beta_o u(1)\\
&y(2)=-a_1 y(1)+\beta_o u(2)+\beta_1 u(1)\\
&y(2)=-a_1 y(2)-a_2 y(1)+\beta_o u(3)+\beta_1 u(2)+\beta_1 u(2)\\
&\text{M}\qquad\qquad\text{M}\qquad\qquad\text{M}\\
&y(k)=-a_1 y(k-1)-\text{L}\ -a_p y(k-p)\\
&+\beta_o u(k)+\beta_1 u(k-1)+\text{L}\ +\beta_p u(k-p)
\end{aligned}
\tag{2.3}
$$

This equation can be described as matrix;

$$
[y_o y]=\theta[V_o V] \tag{2.4}
$$

More explicit expression of matrix elements:

$$
y_o=\begin{bmatrix} y(1) & y(2) & \text{L} & y(p)\end{bmatrix}
$$

$$
y=\begin{bmatrix} y(p+1) & y(p+2) & \text{L} & y(l)\end{bmatrix}
$$

$$
\theta=\begin{bmatrix} \beta_o(-a_1\beta_1) & \text{L} & (-a_{p-1}\beta_{p-1}) & (-a_p\beta_p)\end{bmatrix}
$$

$$
v(k)=\begin{bmatrix} y(k)\\ u(k)\end{bmatrix}\quad k=1,2,3,\text{K},l
$$

$$
V_o=\begin{bmatrix}
u(1) & u(2) & \text{L} & u(\rho)\\
v(0) & v(1) & \text{L} & v(p-1)\\
\text{M} & \text{M} & \text{O} & \text{M}\\
v(2-p) & v(3-p) & \text{L} & v(1)\\
v(1-p) & v(2-p) & \text{L} & v(0)
\end{bmatrix}
\tag{2.5}
$$

$$
V=\begin{bmatrix}
u(p+1) & u(p+2) & \text{L} & u(l)\\
v(p) & v(p+1) & \text{L} & v(l-1)\\
\text{M} & \text{M} & \text{O} & \text{M}\\
v(2) & v(3) & \text{L} & v(l-p+1)\\
v(1) & v(2) & \text{L} & v(l-p)
\end{bmatrix}
$$

Coefficient matrices, θ, in Equation (2.5) can be calculated using of pseudoinverse;

$$\theta = \begin{bmatrix} y_o y \end{bmatrix} \begin{bmatrix} V_o V \end{bmatrix}^n \tag{2.6}$$

V_o and y_o represent initial input and output data, but when initial condition is unknown, V_o and y_o cannot be calculated, so equation is transformed;

$$\theta = yV^n \tag{2.7}$$

2.3 Markov Parameters

When coefficient matrix is calculated for *y(k), y(k+1)* is can be estimated with same coefficients;

$$\begin{aligned} y(k+1) &= -a_1 y(k) - a_2 y(k-1) - \mathrm{L} \; - a_p y(k-p+1) \\ &+ \beta_o u(k+1) + \beta_1 u(k) + \mathrm{L} \; + \beta_p u(k-p+1) \end{aligned} \tag{2.8}$$

y(k) in equations 2.4 can be written in equations 2.8 and so *y(k+1)* can be calculated without knowing *y(k).* If this procedure can be repeated several times, *y(k+j)* can be expressed with past output *y(k-1) …y(k-p).*

$$\begin{aligned} y(k+j) &= -a_1{}^j y(k-1) - a_2{}^j y(k-1) - \mathrm{L} \; - a_p{}^j y(k-p) \\ &+ \beta_o u(k+j) + \beta_o{}^j u(k+j-1) + \mathrm{L} \; + \beta_o{}^j u(k) \\ &+ \beta_1{}^j u(k-1) + \beta_2{}^j u(k-2) + \mathrm{L} \; + \beta_p{}^j u(k-p) \end{aligned} \tag{2.9}$$

Where,

$$\begin{aligned} a_1^{(j)} &= a_2^{(j-1)} - a_1^{(j-1)} a_1 \\ a_1^{(j)} &= a_3^{(j-1)} - a_1^{(j-1)} a_2 \\ \mathrm{M} \quad & \quad \mathrm{M} \quad \mathrm{M} \\ a_{p-1}^{(j)} &= a_p^{(j-1)} - a_1^{(j-1)} a_{p-1} \\ a_p^{(j)} &= -a_1^{(j-1)} a_p \end{aligned} \tag{2.10}$$

and;

$$
\begin{aligned}
\beta_0^{(j)} &= \beta_1^{(j-1)} - a_1^{(j-1)}\beta_o \\
\beta_1^{(j)} &= \beta_2^{(j-1)} - a_1^{(j-1)}\beta_1 \\
&\text{M} \quad\quad \text{M} \quad\quad \text{M} \\
\beta_{p-1}^{(j)} &= \beta_p^{(j-1)} - a_1^{(j-1)}\beta_{p-1} \\
\beta_p^{(j)} &= -a_1^{(j-1)}\beta_p
\end{aligned}
\tag{2.11}
$$

Equation 2.12 can be described as matrix;

$$
y_s(k) = Tu_s(k) - Bu_p(k-p) - Ay_p(k-p) \tag{2.12}
$$

More explicit expression of matrix elements:

$$
T = \begin{bmatrix}
\beta_o & & & \\
\beta_o^1 & \beta_o & & \\
\text{M} & \text{M} & \text{O} & \\
\beta_o^{(s-1)} & \beta_o^{(s-2)} & \text{L} & \beta_o
\end{bmatrix}
$$

$$
B = \begin{bmatrix}
\beta_p & \beta_{p-1} & \text{L} & \beta_1 \\
\beta_p^1 & \beta_{p-1}^{(1)} & \text{L} & \beta_1^{(1)} \\
\text{M} & \text{M} & \text{O} & \text{M} \\
\beta_p^{(s-1)} & \beta_{p-1}^{(s-1)} & \text{L} & \beta_1^{(s-1)}
\end{bmatrix}
$$

$$
A = \begin{bmatrix}
a_p & a_{p-1} & \text{L} & a_1 \\
a_p^1 & a_{p-1}^{(1)} & \text{L} & a_1^{(1)} \\
\text{M} & \text{M} & \text{O} & \text{M} \\
a_p^{(s-1)} & a_{p-1}^{(s-1)} & \text{L} & a_1^{(s-1)}
\end{bmatrix}
\tag{2.13}
$$

$$
y_s(k) = \begin{bmatrix}
y(k) \\
y(k+1) \\
\text{M} \\
y(k+s-1)
\end{bmatrix}
$$

$$y_p(k-p)=\begin{bmatrix} y(k-p) \\ y(k-p+1) \\ \text{M} \\ y(k-1) \end{bmatrix}$$

$$u_p(k-p)=\begin{bmatrix} u(k-p) \\ u(k-p+1) \\ \text{M} \\ u(k-1) \end{bmatrix}$$

In Equation 2.13, output vector is described as $y_s(k)$ and $y_p(k\text{-}p)$, input vector is described as $u_s(k)$ and $u_p(k\text{-}p)$.

Markov parameters, also known as impulse response functions can be identified with using input-output experimental data. They can be used to compute a state-space model of structural systems.

If system is initially at rest, elements of output matrix, $y_p(k\text{-}p)$, in equation 2.13 is become zero. To simplify elements of input matrices, $u_s(k)$ and $u_p(k\text{-}p)$, can be assumed as zero except at time k, $u(k) = 1$. With this assumption, equation 2.14 can be arranged as follows:

$$\begin{bmatrix} y(k) \\ y(k+1) \\ \text{M} \\ y(k+s-1) \end{bmatrix}=\begin{bmatrix} \beta_o \\ \beta_o^{(1)} \\ \text{M} \\ \beta_o^{(s-1)} \end{bmatrix} \tag{2.14}$$

Equation 2.15 can be called Toeplitz pulse-response matrix. It can be used in discrete-time state-space model:

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k) \\ y(k) &= Cx(k)+Du(k) \end{aligned} \tag{2.15}$$

where x is the state vector. With zero initial condition, $x(k) = 0$ and with unit pulse $u(k) = 1$:

$$\begin{array}{cc} x(k)=0 \Rightarrow & y(k)=D \\ x(k+1)=B \Rightarrow & y(k+1)=CB \\ x(k+2)=AB \Rightarrow & y(k+2)=CBA \\ \text{M} & \text{M} \\ \text{M} & \text{M} \\ x(k+s-1)=A^{s-2}B \Rightarrow & y(k+s-1)=CA^{s-2}B=\beta^{s-1} \end{array} \tag{2.16}$$

D, CB, CAB…CA^{s-2}B are called as Markov parameters.

2.4 Eigensystem Realization Algorithm (ERA)

Feature extraction is carried out using a time-domain identification technique, namely Eigensystem Realization Algorithm (ERA) (Juang, 1994). ERA was proposed for modal parameter identification and model reduction of linear dynamical systems formulated. Operating on pulse response functions, also known as Markov Parameters, this technique produces an input/output mapping having the smallest state vector dimension that is compatible with a given accuracy.

The system realization starts with created the Hankel matrix of size $m_s \, x \, r_s$ from terms in Equation 2.14,

$$H(k-1)=\begin{pmatrix} y(k) & \text{K} & y(k+s-1) \\ \text{M} & \text{O} & \text{M} \\ y(k+s-1) & \text{L} & y(k+2(s-1)) \end{pmatrix} \tag{2.20}$$

where s is an integer that determines the size of the matrix. To remind, size of output vector is m_{x1}, and size of input vector is r_{x1}.

The basic formulation of ERA begins with the factorization of the Hankel matrix using the singular value decomposition,

$$H(0)=USV^T \tag{2.21}$$

$$H(1)=US^{1/2}AS^{1/2}V^T \tag{2.22}$$

Thus, the following triplet is a minimum realization:

$$\hat{A} = S^{-1/2} U_1^T H(1) V_1 S^{-1/2} \tag{2.21}$$

$$\hat{B} = S^{1/2} V_1^T E_r \tag{2.22}$$

$$\hat{C} = E_m^T U_1 S^{1/2} \tag{2.21}$$

where $E_m^T$is $[I_m\ O_m\ \cdots\ O_m]$, E_r^T is $[I_r\ O_r\ \cdots\ O_r]$, O_i is a null matrix and I_i is an identity matrix of order *i*.

Once the system realization is obtained, the eigenvalues (Λ) and eigenvectors (Ψ) of the state matrix *A* and the state-to-output influence matrix *C* can be used to obtain the modal parameters.

$$\hat{A} = \Psi \Lambda \Psi^{-1} \tag{2.22}$$

The eigenvalues (v) can be converted from discrete time to continuous time representation:

$$w_i = \frac{\ln(\Lambda)}{dt} \tag{2.23}$$

w is natural frequencies of structure.

3. SUPPORT VECTOR MACHINES FOR DAMAGE DETECTION

3.1 Introduction

There are two main approaches to structural health monitoring: model based and data driven approaches. In the model bases approaches the inverse problem technique is applied to finite element model of the structure for calibrating the model to relate with data. Damage is identified by fitting the measured data from the structure to finite element model of structure. The data-driven technique in the field of machine learning operates with pattern recognitions algorithm. The behavior of the structure is learned from past data in a supervised or unsupervised manner and using the same principles as the human brain the algorithms are applied with trained machine to identify damage.

There are two types classification problem: (1) single class classification and (2) multi class classification. Single class classification tells us whether data belongs a known class. Multi class classification aims to classify data object into one of several known classes.

Support Vector Machine (SVM) is the classification method built on Vapnik's (1995) theory. This method uses hyperplanes which is constructed by support vector to separate data. Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane.

3.2 Multi-class Classification

In the two or multi-class classification, maximizing the distance between the separating hyperplane and data to find optimal hyperplane is the goal of the SVMs (Shin et al.). In other words, algorithm constructs optimal hyperplane using training data to categorize new data. In the binary classification, the line as hyperplane is used to separate data (Figure 3.1).

Selecting optimal hyperplane is become a problem when wanting to separate datas into two classes and support vector machine can be solution of this problem (Figure 3.2).

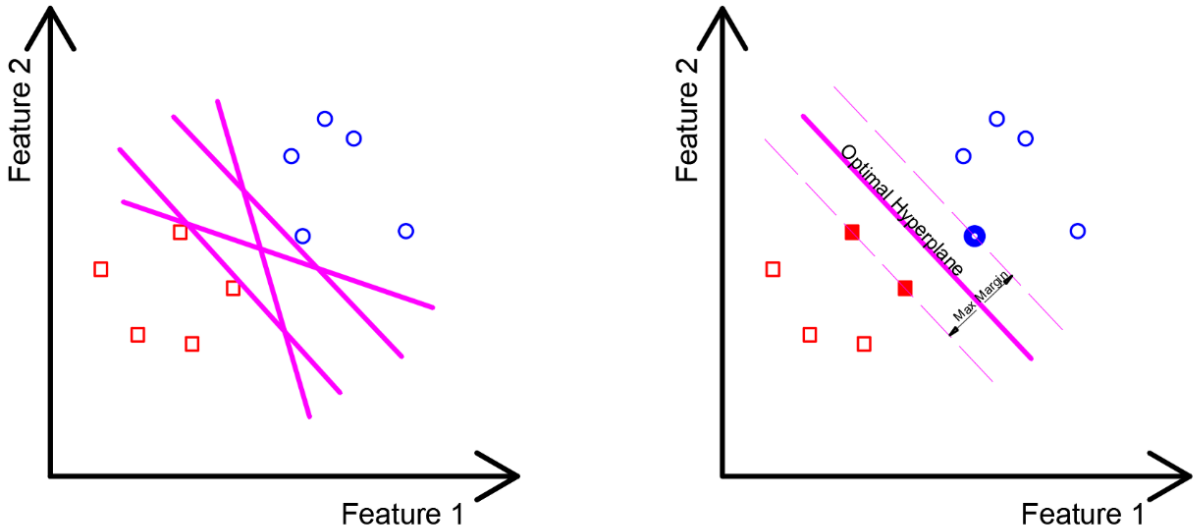


Figure 3.1 : Binary classification example.

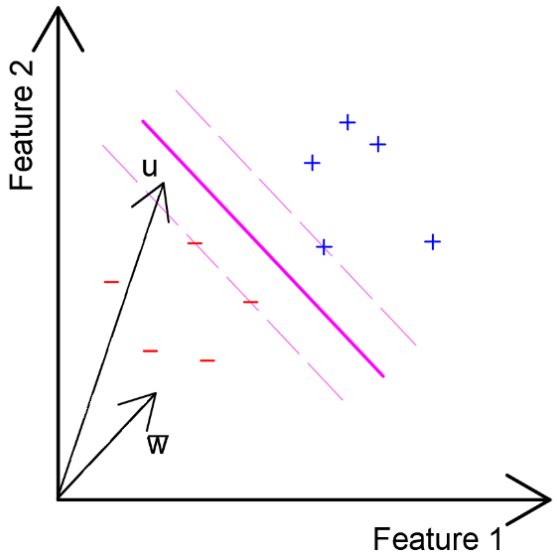


Figure 3.2 : Positive and negative classes.

w is vector, constrained to be perpendicular to the optimal hyperplane and u is vector which is unknown length.

$$w.u \geq c \quad \text{and} \quad c = -b \tag{3.1}$$

and

$$w.u + b \geq 0 \tag{3.2}$$

Equation 3.3 is the decision rule for classification. This equation can be written as;

$$w.x_{+} + b \geq 1 \quad \text{and} \quad w.x_{-} + b \geq 1 \tag{3.3}$$

x_+ is represent positive sample, x_- is represent negative sample.

y_i is the variable that takes positive values for positive samples and negative values for negative samples. Then, equation 3.3 is multiplied by 'y_i' ;

$$y_i\left(w.x_{+} + b\right) \geq 1 \quad \text{and} \quad y_i\left(w.x_{-} + b\right) \geq 1 \tag{3.4}$$

When data point (x_i) is support vector, equation turns into;

$$y_i\left(w.x_i + b\right) - 1 = 0 \tag{3.5}$$

The length of margin (Figure 3.3) can be obtained based on following formula;

$$\left(x_{+} - x_{-}\right).\frac{w}{\|w\|} = x_{+}.\frac{w}{\|w\|} - x_{-}.\frac{w}{\|w\|} = \frac{1-b}{\|w\|} + \frac{1+b}{\|w\|} = \frac{2}{\|w\|} \tag{3.6}$$

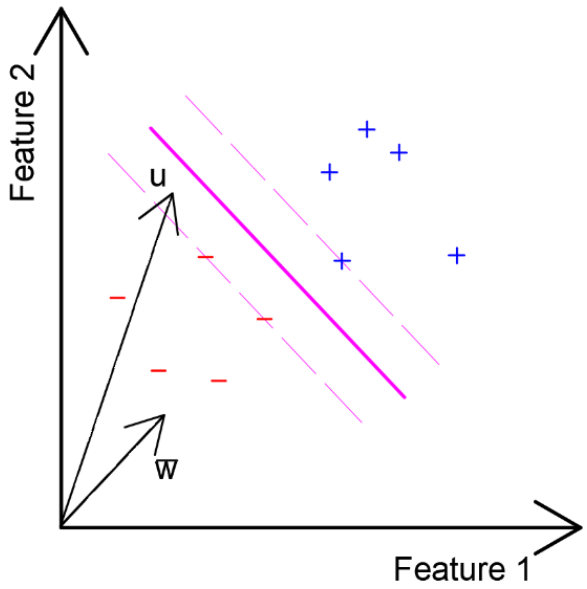


Figure 3.3 : Calculation of margin.

To maximize margin, one of values of max ($\frac{2}{\|w\|}$), max ($\frac{1}{\|w\|}$), min ($\|w\|$) or min ($\frac{1}{2}\|w\|^2$) should be calculated.

Let's applied Lagrange to equation 3.5 and equation 3.6;

$$L(w,b,a) = \frac{1}{2}\|w\|^2 - \sum a_i \left[y_i \left(w.x_i + b \right) - 1 \right] \tag{3.7}$$

$\boldsymbol{\alpha} = (\boldsymbol{\alpha_i} \ldots . \boldsymbol{\alpha_n})$ is the vector of Lagrange multipliers.

If the partial derivative respect to '*w*' of equation 3.7 is taken;

$$\frac{\partial L(w,b,a)}{\partial w} = w - \sum a_i y_i x_i = 0 \quad \rightarrow w = \sum a_i y_i x_i \tag{3.8}$$

If the partial derivative respect to '*b*' of equation 3.8 is taken;

$$\frac{\partial L(w,b,a)}{\partial w} = \sum a_i y_i \tag{3.9}$$

Let's rewrite equation 3.10 using equation 3.8 and equation 3.9;

$$\begin{aligned} L(w,b,a) = \frac{1}{2}\left(\sum a_i y_i x_i\right)\left(\sum a_j y_j x_j\right) - \left(\sum a_i y_i x_i\right)\left(\sum a_j y_j x_j\right) - \\ \text{L} \ -\left(\sum a_i y_i b\right) + \sum a_i \end{aligned} \tag{3.10}$$

If we edit equation 3.10 and write the final form of the equation;

$$L(a) = \sum a_i - \frac{1}{2}\sum\sum a_i a_j y_i y_j x_i x_j \tag{3.11}$$

Equation 3.11 can be solved to find maximum values via quadratic programming.

If data points can be separated linearly like in Figure 3.1, this formulation can be successful in classification. When data points are as in the Figure 3.4, kernel methods can be used. Kernel methods transforms data points to another dimension, generally higher:

$$K\left(x_i, x_j\right) = \varphi\left(x_i\right).\varphi\left(x_j\right) \tag{3.12}$$

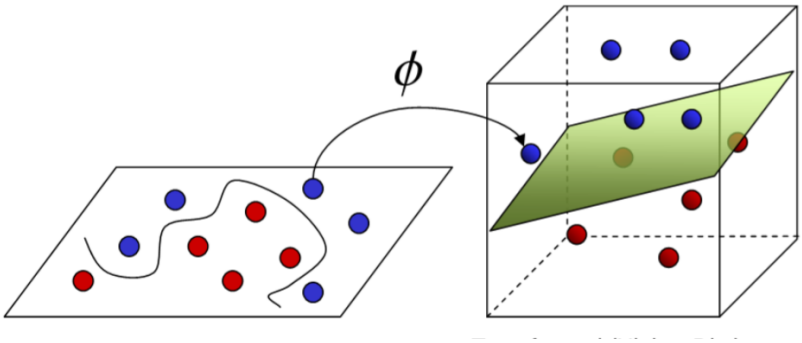


Figure 3.4 : Linearly unseperable data points.

There are few kernel methods can be used in support vector machine algorithm in MATLAB:

$$K\left(x_i, x_j\right) = \exp\left(-\frac{\|x - x\|}{2\sigma^2}\right)^2 \rightarrow \text{Gaussian or Radial Basis Kernel}$$

$$K\left(x_i, x_j\right) = x_i^T . x_j \rightarrow \text{Linear Kernel}$$

$$K\left(x_i, x_j\right) = \left(1 + x_i^T . x_j\right)^q \rightarrow \text{Polynomial Kernel}$$

$$K\left(x_i, x_j\right) = \tanh(1 + \beta_0 x_i^T x_j) \rightarrow \text{Sigmoid Kernel}$$

When kernel methods are used in algorithm, equation 3.12 becomes;

$$L(a) = \sum a_i - \frac{1}{2}\sum\sum a_i a_j y_i y_j \varphi\left(x_i\right)\varphi\left(x_j\right) \tag{3.14}$$

3.3 Single Class Classification

One-Class Classification methods have emerged in recent years as promising classification methods for problems where the main objective is to distinguish between a set of objects belonging to one class, termed the target class and the rest of the samples not belonging to this class, referred to as outliers.

In order to classify the data, an OC-SVM estimates a distribution that encompasses most of the observations, and then those that lie far from it with respect to a suitable metric are identified as outliers.

There are two popular approaches to solve single class support vector machine (Figure 3.5): (1) Support Vector Data Description (SVDD) and (2) One-Class SVM (OC-SVM).

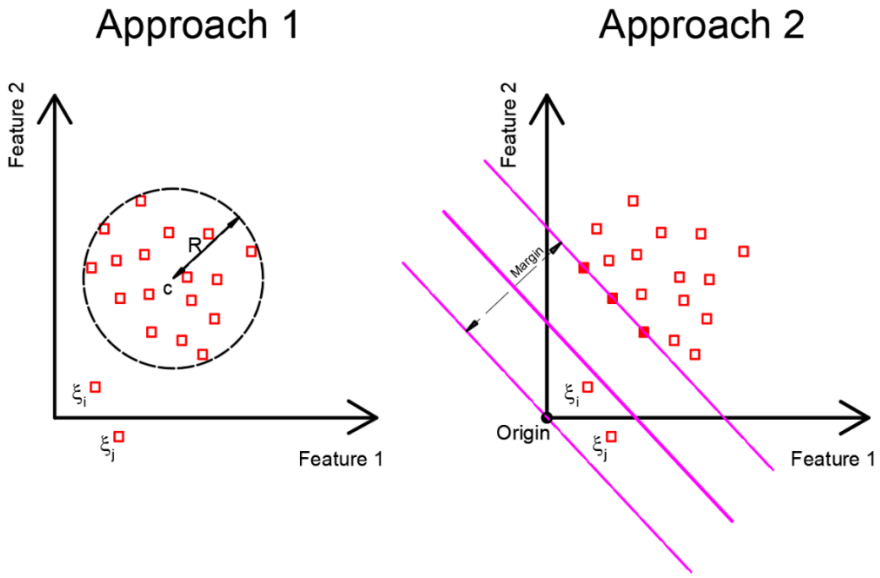


Figure 3.5 : Single class support vector machines solving options.

SVDD proposed by Tax and Duin (2014) describes a circle to encompass all training data with minimum radius. R is radius of circle, a is center of radius, C is constant and ξ_i is the non-zero slack variables. It is minimization problem;

$$f(R,a) = R^2 + C\sum \xi_i \tag{3.15}$$

with constraints that almost all objects are within the sphere;

$$\|x_i - a\|^2 \leq R^2 + \xi_i \quad \text{and} \quad \xi_i \geq 0 \tag{3.16}$$

OC-SVM proposed by Schölkopf et al. (2001) separates the training data from the origin with maximum margin. The idea behind the OC-SVM is to describe the target class by a function that maps most part of it to a region where the function is nonzero. To this end, the origin is treated as the only available member of the non-target class, an outlier, and then, the problem is solved by finding a hyperplane with maximum margin separation from the origin.

Let's define a training data set x, in an m dimensional real space:

$$x = \left(x_i, x_2, \mathrm{K}, x_m \right) \tag{3.17}$$

and a mapping function as mentioned above kernel methods, φ () that can map the data from their original feature space to a new feature space H such that the hyperplane Π in H has the largest distance to the origin, that is:

$$w^T . \varphi\left(x\right) - \rho = 0 \tag{3.18}$$

where w is the normal vector of Π and all the mapped training data lie at the opposite side of the hyperplane such that

$$\min_{w,\xi,\rho} \frac{1}{2} w^T w - \rho + \frac{1}{vm} \sum_{i=1}^{m} \xi_i = 0 \tag{3.19}$$

where $\xi_i \geq 0$ for all i and ν is the regularization coefficient controlling the trade-off between the maximizing the distance between the origin and keeping most of the data in the hyperplane, and ξ_i's are the non-zero slack variables that enable OC-SVM to have a soft margin by allowing some $\boldsymbol{x}_i$ to be located outside the decision boundary during the training phase. Since the mapping $\boldsymbol{\varphi}()$ is usually implicit, the above optimization problem is usually solved by its dual form:

$$\max_{\alpha} \frac{1}{2} \sum_{i,j=1} a_i a_j K\left(x_i, x_j\right) = 0 \tag{3.20}$$

such that

$$\sum_{i=1}^{m} a_i = 1 \quad 0 \leq a_i \leq \frac{1}{vm} \tag{3.21}$$

and

$$K\left(x_i, x_j\right) = \varphi\left(x_i\right)^T . \varphi\left(x_j\right) \tag{3.22}$$

Instead of defining the mapping function $\boldsymbol{\varphi}()$ it is practical to define directly the kernel function $K(x_i, x_j)$. As discussed by Schölkopf et al. (2001), the Gaussian kernel guarantees the existence of possible solutions and the Gaussian kernel is usually the standard choice:

$$K\left(x_i, x_j\right) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{\sigma^2}\right) \tag{3.23}$$

where σ is the Gaussian kernel width

The above optimization problem (#) will lead to α_i's and any $\boldsymbol{x}_i$ that correspond to a non-zero α_i's will be the support vectors. The decision function for any new data, $\boldsymbol{x}_t$, can be expressed as

$$f\left(x_t\right) = \operatorname{sgn}\left(\sum_{i=0}^{m} a_i K\left(x_t, x_i\right) - \rho\right) \tag{3.24}$$

and ρ can be obtained based on the following formula

$$\begin{aligned} \rho &= \sum_j a_j K\left(x_j, x_t\right) \\ \text{where} \quad a_i &\in \left(0, \frac{1}{vm}\right) \end{aligned} \tag{3.25}$$

The geometrical interpretation of the idea of one-class classification is illustrated in Figure 3.6. In Figure 3.6(a), the hyperplane separates with maximum margin all target data from the origin by mapping all targets to the upper side of the hyperplane and outliers to the lower side. Only the green points that are on the hyperplane are needed to make predictions. Figure 3.6(b) illustrates the approach in high dimensional feature space (Li and Liu, 2003). The "normal" class denoted by blue squares and marked as '+1 Class', correspond to a "healthy" structural system. The OC-SVM approach, at the training stage constructs a minimal-volume hypersphere around +1 Class. The origin and its proximity are associated with 'outlier' data representing the "damaged" class (-1 Class). The approach classifies a novel point belonging to a "healthy" or "damaged" class using the geometry of the trained OC-SVM in feature space based on its location relative to the hyperspace even though no training data from the damaged class was available. Any point outlying the hypersphere is considered to be a member of the "damaged" class.

The computational complexity of the training phase is intensive because the training of an OC-SVM involves a quadratic programming problem, but once the decision function is determined, it can be used to predict the class label of new test data effortlessly.

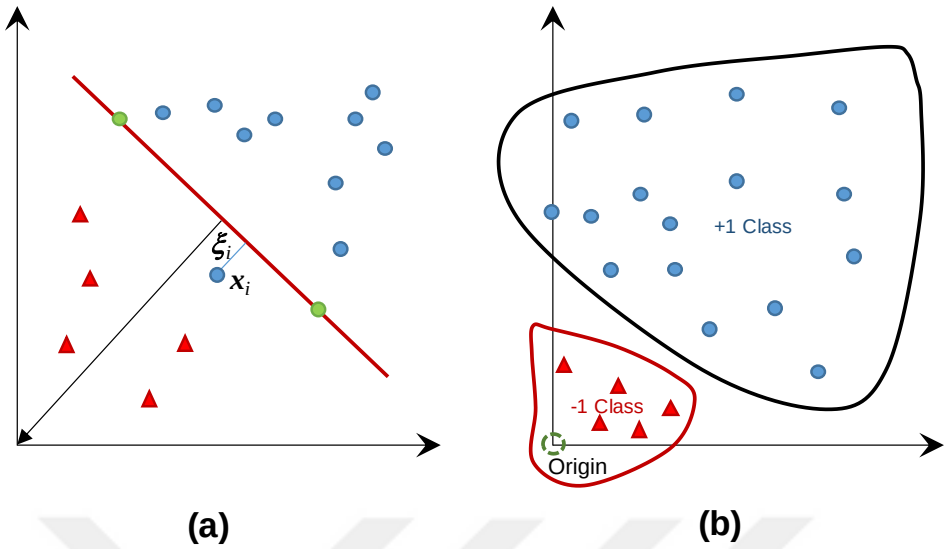


Figure 3.6 : a) OC-SVM approach b) Improved OC-SCM approach (Li and Liu, 2003)

.

4. DAMAGE NUMERICAL ILLUSTRATION OF THE DAMAGE DETECTION STRATEGY

4.1 Introduction

In this chapter, damage detection performance of one-class support vector machine (OC-SVM) algorithm was investigated. To validate the methodology, different types of damages at different levels were applied to 3-storey 2-D plane frame. The first three natural frequencies and the arx model parameters are explored as features for potential damage detection indicators for different damage classes and levels. Afterwards, damage detection performance of these features were compared with each other.

4.2 Finite Element Model

2-D structure with selected sensor locations and directions for acceleration measurement is as shown in the Figure 4.1. The span length and story height of structure are 6 m, 3 m respectively. SAP2000 V15.2.1 was employed to analyse to this 2-D structure. The modal frequencies and periods corresponding to the translational modes of the structure are given in a Table 4.1 and mode shapes are shown in graphically in Figure 4.2.

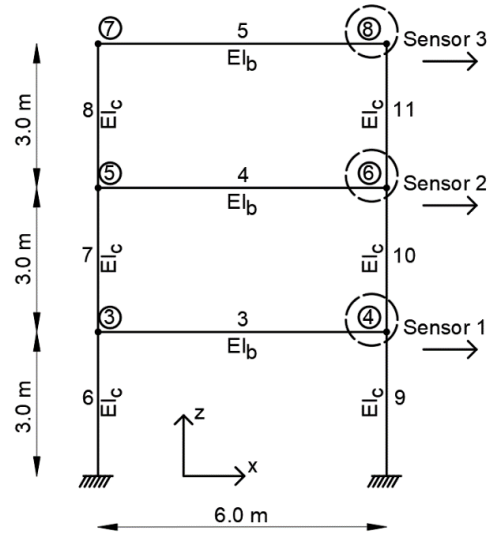
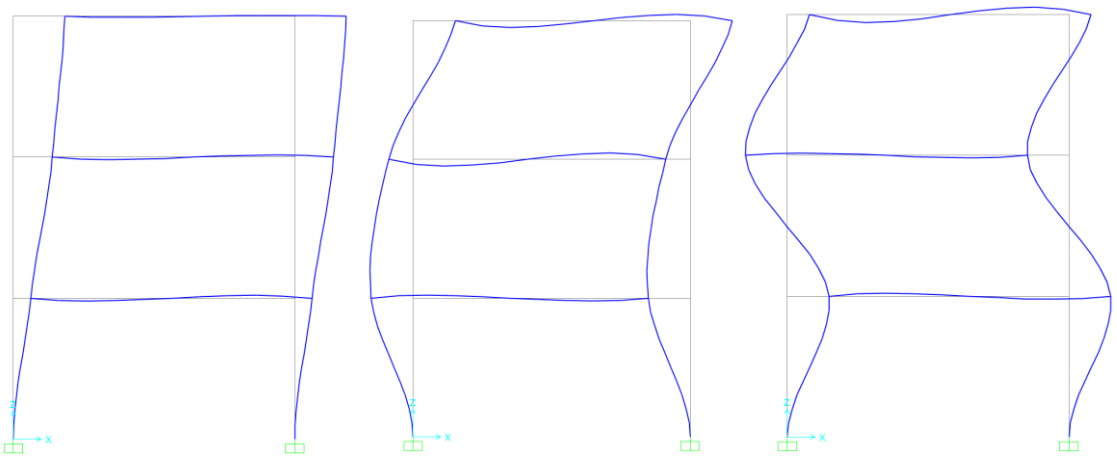


Figure 4.1 : 2D Frame structure used in the simulations and the sensor locations.

Table 4.1 : Model properties of structure.

Modes	Period (sec)	Frequency (cyc/sec)
1	0.939	1.065
2	0.299	3.345
3	0.179	5.597



(1st mode) (2nd mode) (3rd mode)

Figure 4.2 : 2D Frame structure mode shapes.

4.3 Implementation of the Methodology

To initially simulate measured acceleration response of selected nodes corresponding to undamaged state of structure, the structure was subjected to ground motion excitation: Kocaeli earthquake record from Yarımca (Figure 4.3). Undamaged state of structure essentially will be considered as the nominal or the safe state of the structure (Figure 4.4).

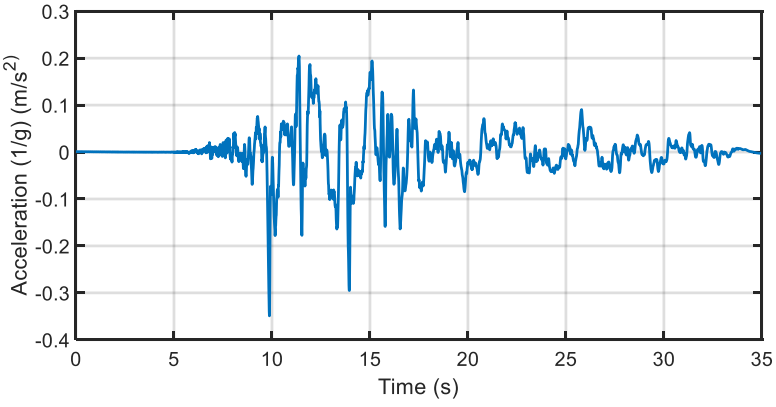


Figure 4.3 : Kocaeli earthquake record from Yarımca station.

For simulated data to resemble real life sensor measurements, the data generated with dt = 0.01 (F_s = 100 Hz) is polluted artificially with random noise of variance σ^2 such that

$$SNR = \frac{\sigma^2_{noise}}{\sigma^2_{signal}} \tag{4.1}$$

where σ^2_{signal}, σ^2_{noise} are the variance of the simulated acceleration and the noise signals and *SNR* is the desired signal to noise ratio. Data was corrupted with various SNR ratios ranging 5%-10% that are specified randomly using a random number generator (Figure 4.5). Furthermore, this measurement was polluted with 300 times for the results to have statistical meaning.

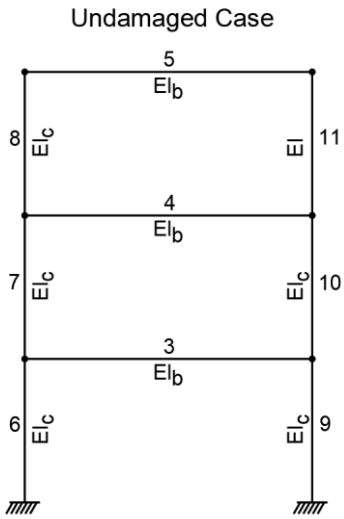


Figure 4.4 : Nominal state of structure.

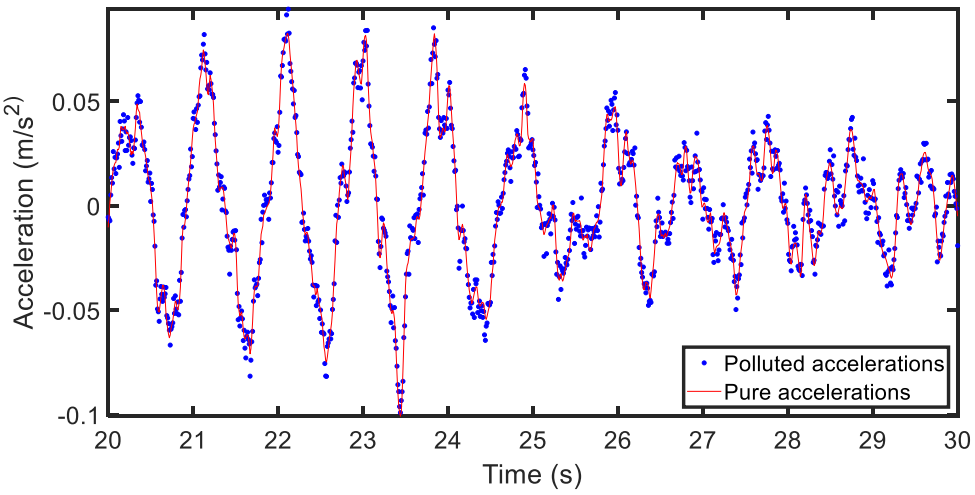


Figure 4.5 : Noisy acceleration data versus pure acceleration data from Sensor 3.

Polluted accelerations belong to nominal state of structure were used to extract damage sensitive features of structure. Six different damage sensitive features listed in Table 4.2 were employed in this study. Inıtial examination of the model suggested that a 5-term ARX model would provide a good representation of the data. Hence a 5-term ARX model is generated using each acceleration recording as a single set of available measurements. Then, created ARX model were used to predict accelerations of nodes during Kobe earthquake (Figure 4.6). Differences between predicted accelerations and accelerations obtained from finite element model in all time interval were used to calculate residual mean error. This is calculated for each sensor measurement and 3-term damage sensitive feature was attained. Standard deviations of residual error for each sensor location was also explored as another alternative for damage sensitive feature. In addition to these, ARX model coefficients and the natural frequencies obtained from acceleration measurements using Eigensystem Realization Algorithm (ERA) constitute all the detection features investigated in this study.

As for training of model, MATLAB 'fitcsvm' function was employed as the classifier. This function trains a support vector machine (SVM) model for one-class and two-class (binary) classification on a predictor data set. It supports mapping the predictor data using kernel functions and use sequential minimal optimization (SMO) to solve optimization algorithm. In this thesis, radial basis function (RBF) was selected as a kernel function. As an example, created SVM model via 'fitcsvm' function was shown in the Figure 4.7.

Table 4.2 : Damage sensitive features.

Damage Sensitive Feature	Representative of Features
ARX model coefficients created using accelerations of sensor (1)	Feature (1)
ARX model coefficients created using accelerations of sensor (2)	Feature (2)
ARX model coefficients created using accelerations of sensor (3)	Feature (3)
Residual mean errors between prediction accelerations and accelerations obtained from finite element model for each sensor location	Feature (4)
Standard deviations of residual errors for each sensor location	Feature (5)
Natural frequencies of system	Feature (6)

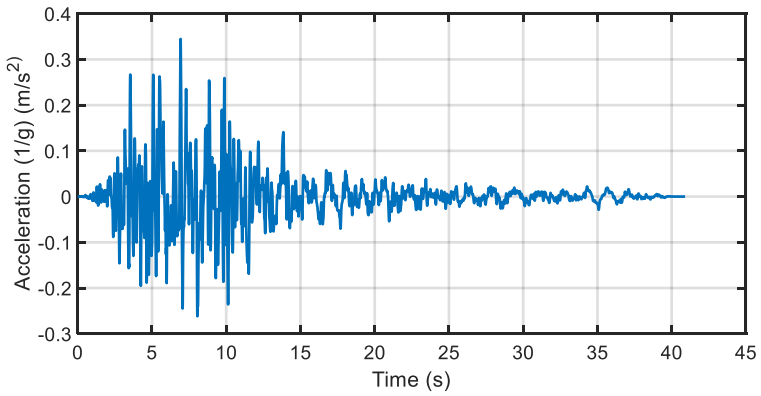


Figure 4.6 : Kobe earthquake record from Kakogawa station.

To test support vector machine model, a total of eighty damaged scenarios were performed at seven different damage cases that are increasing damage level incrementally. The damaged data set was then numerically modelled using the finite element model of the structure numerical model. There are 7 damage cases as one major damage and six local damage as listed in Table 4.3. As major damage, reduction in the elastic modulus of the material (Damage Case 1) was investigated in between 2 and 20 scenarios (Figure 4.8). The elastic modulus of the material reduced by one percent per step, and in the last step it reduced to eighty one percent of the initial value. Other damage scenarios are local damage, like reduction in rigidity of beam-column and column-beam connection. Rigidity of connection for moment reduced by ten percent per step, and in the last step rigid connection turns into pinned connection (Figure 4.8). A total of twenty-four thousand simulations were obtained by pollution to 300 times for each damage scenario.

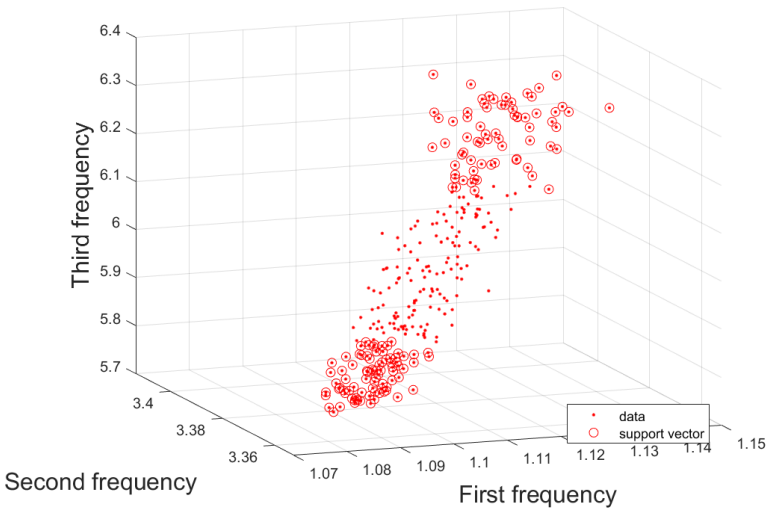


Figure 4.7 : One-class classification in graphically.

Table 4.3 : Summary of the simulation cases

Damage Case	Damage Scenarios	Number of Simulations
Undamaged	1	300
Damage 1: reduction in the elastic modulus of the material	2-20	5700
Damage 2: reduction in the rigidity of connection of beam(5) to column(11)	21-30	3000
Damage 3: reduction in the rigidity of connection of beam(4) to column(10)	31-40	3000
Damage 4: reduction in the rigidity of connection of beam(5) to column(11) + reduction in the rigidity of connection of beam(4) to column(10)	41-50	3000
Damage 5: reduction in the rigidity of connection of column(8) to beam(5)	51-60	3000
Damage 6: reduction in the rigidity of connection of column(7) to beam(4)	61-70	3000
Damage 7: reduction in the rigidity of connection of column(6) to beam(3)	71-80	3000
Total:	80	240000

To summarize, a total of twenty-four thousand simulations were performed for the eighty damage scenarios and damage detection is then carried out for and response of MATLAB ‘fitcsvm’ function indicating whether the structure is damage or not, is returned for each simulation (Figure 4.9). It means that one-class classifiers created from nominal state of structure were used to understand the status of each simulation. For one-class classification, MATLAB ‘fitcsvm’ function returns positive value if the data belongs to training class, otherwise it returns negative value. For each damage scenario, the damaged detection performance was obtained with the positive result rate of 300 simulations of the returned 'fitcsvm' function.

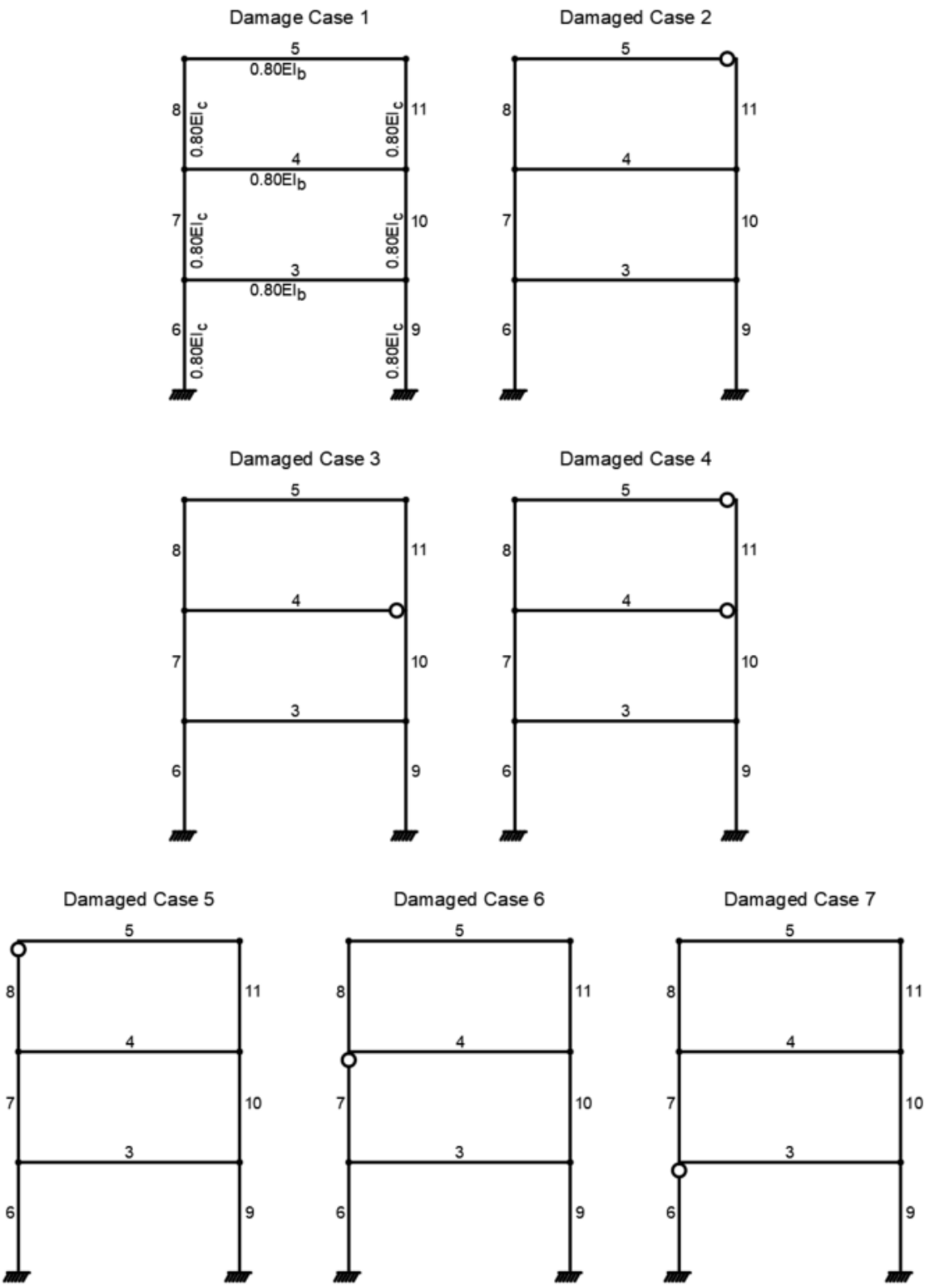


Figure 4.8 : Damage case scenarios.

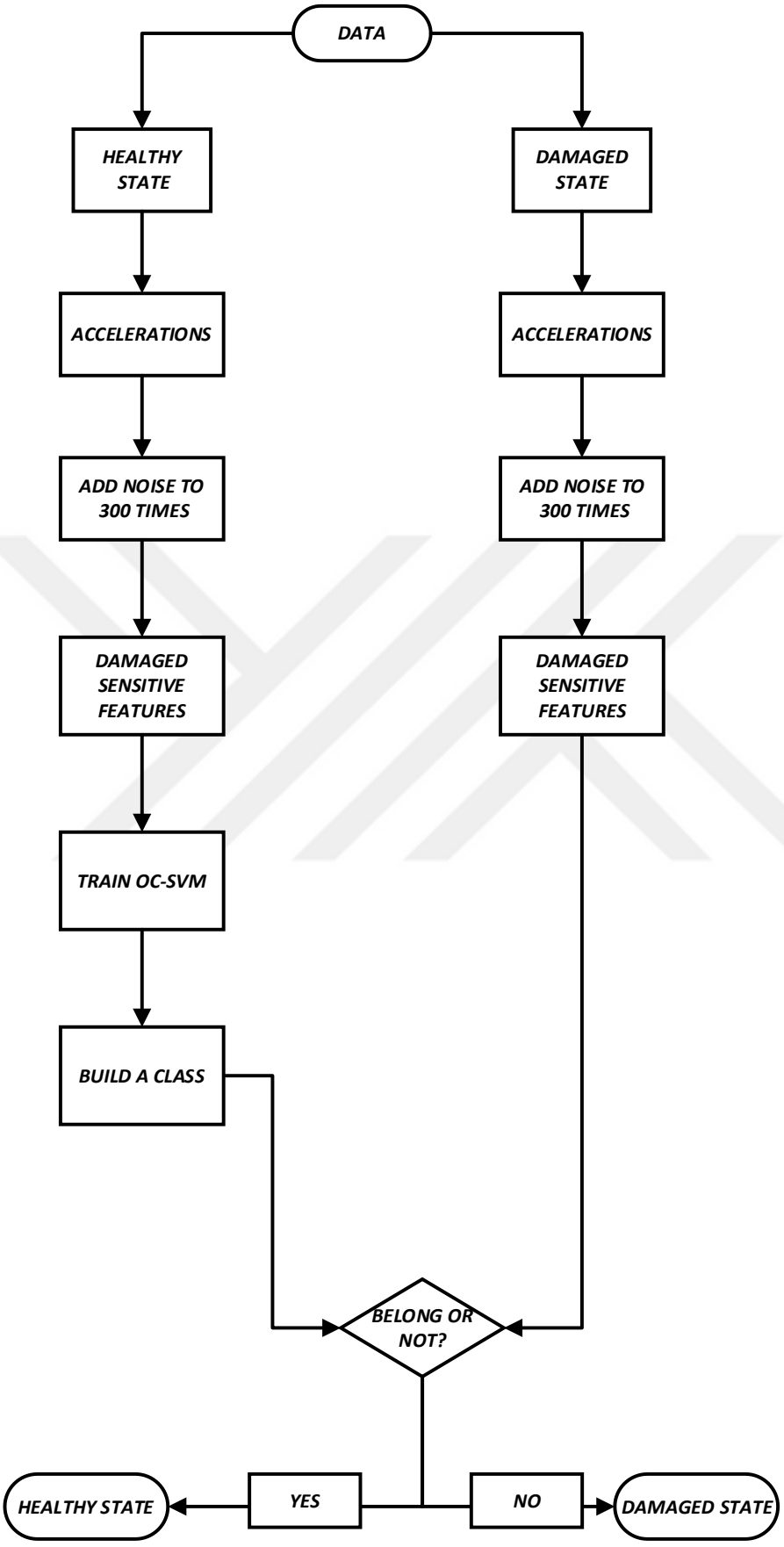


Figure 4.9 : Summary of the methodology.

4.4 Results of the Applied Methodology

Damage detection rates for 80 damage scenarios are shown in Figure 4.10 to Figure 4.15. In the figures, the x-axis respresents damage scenarios and the y-axis respresents the damage detection performance obtained from simulations. The simulation results corresponding to different damage cases are represented with diffrenet markers as shown in the legend.

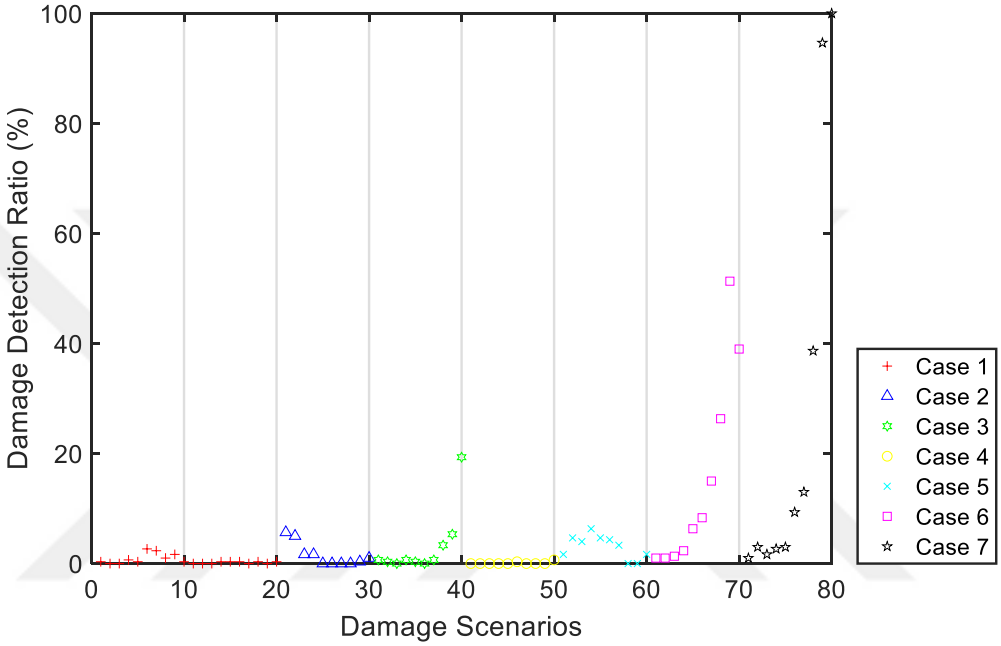


Figure 4.10 : Damage detection ratios via damage detection feature (1).

Firstly, ARX model coefficients that belong to node (4) are used to detect damage. The damage detection performance is not found satisfactory either for the local or the global damage scenarios with the exception of Damage Case 6 and Damage Case 7 as depicted in Figure 4.10. These two damage cases are reduction in the rigidity of connection of column(7) to beam(4) and reduction in the rigidity of connection of column(6) to beam(3).

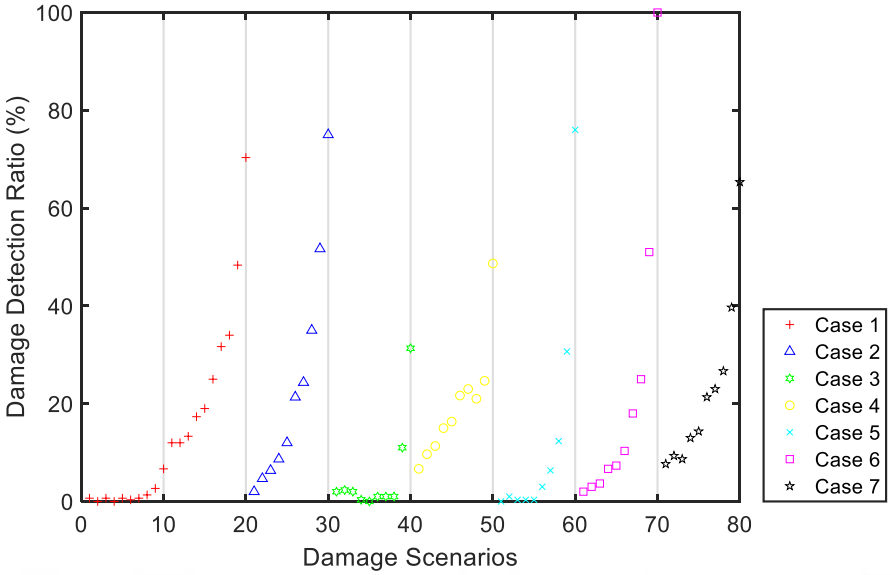


Figure 4.11 : Damage detection ratios via damage detection feature (2).

The results of damage detection that is used to ARX parameters belong to node (6) is shown in Figure 4.11. For the last step of Damage Case 1, 2, 6 and 7, damage detection ratio was obtained as higher than 70%. The best damage detection ratio, 100%, was seen at Damage Case 6 that is reduction in column-beam connection. This clearly indicates that , local damages occurring close to the node where the ARX model was created have a much better chance of detection.

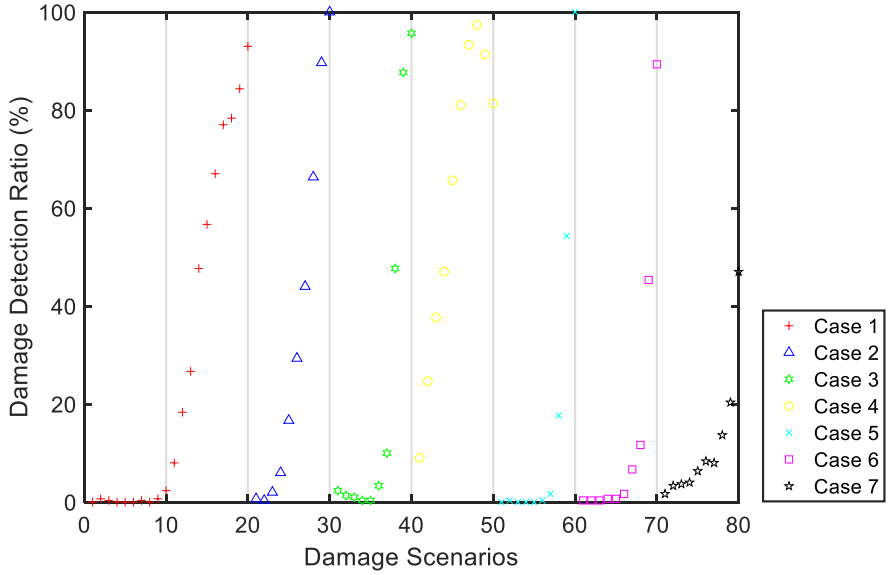


Figure 4.12 : Damage detection ratios via damage detection feature (3).

Figure 4.12 is about result of damage detection performance that is used ARX coefficients of node (8). The damage detection performance is excellent with almost 100% detection rates for both the local and global damage cases. For this specific structure, the top floor sensor data collected from joint (8) performed very well with ARX model coefficients utilized as the damage detection feature. For the global damage cases this is as expected due to effect of the modal amplitudes in enhancing the observability of the modal identification results.

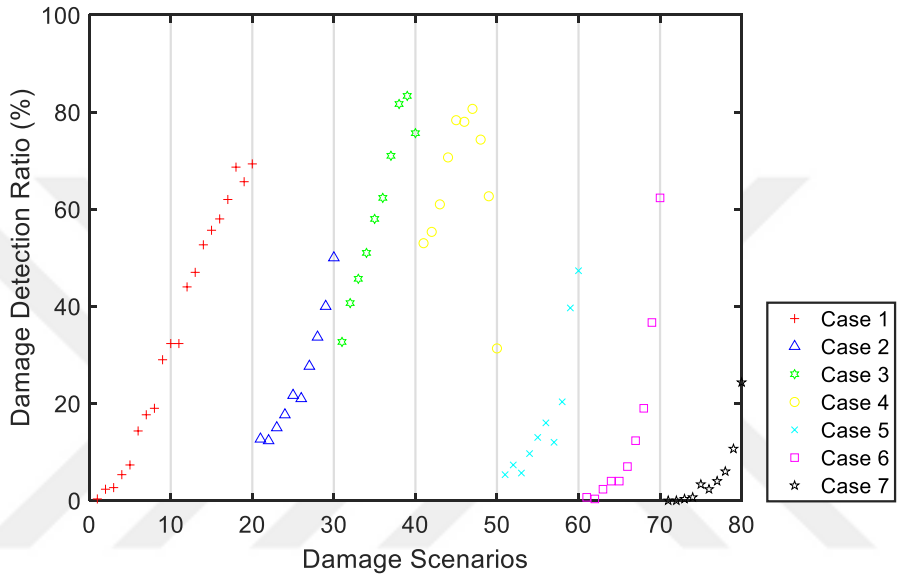


Figure 4.13 : Damage detection ratios via damage detection feature (4).

The performance of damage detection of Feature (4) is shown in Figure 4.13. For the last step of Damage Case 1 and 3, damage detection ratio was obtained as higher than 70%. As seen from figure, this SVM model created with residual mean errors are more sensitive to the detection of damage of beam-column connection than the other features. However, damage detection ratio for the last simulation of Damage Case 4 appears as an outlier.

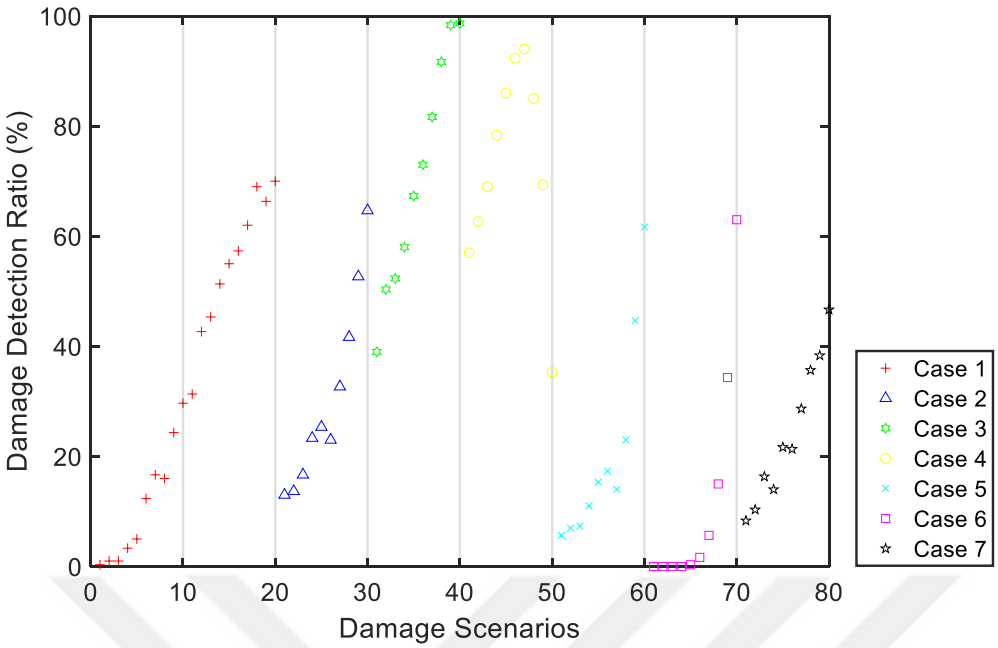


Figure 4.14 : Damage detection ratios via damage detection feature (5).

Damage detection performance obtained from SVM model using standard deviations is shown in Figure 4.14. For the last step of Damage Case 1 and 3, damage detection ratio was obtained as higher than 70%. These results are similar to previous results.

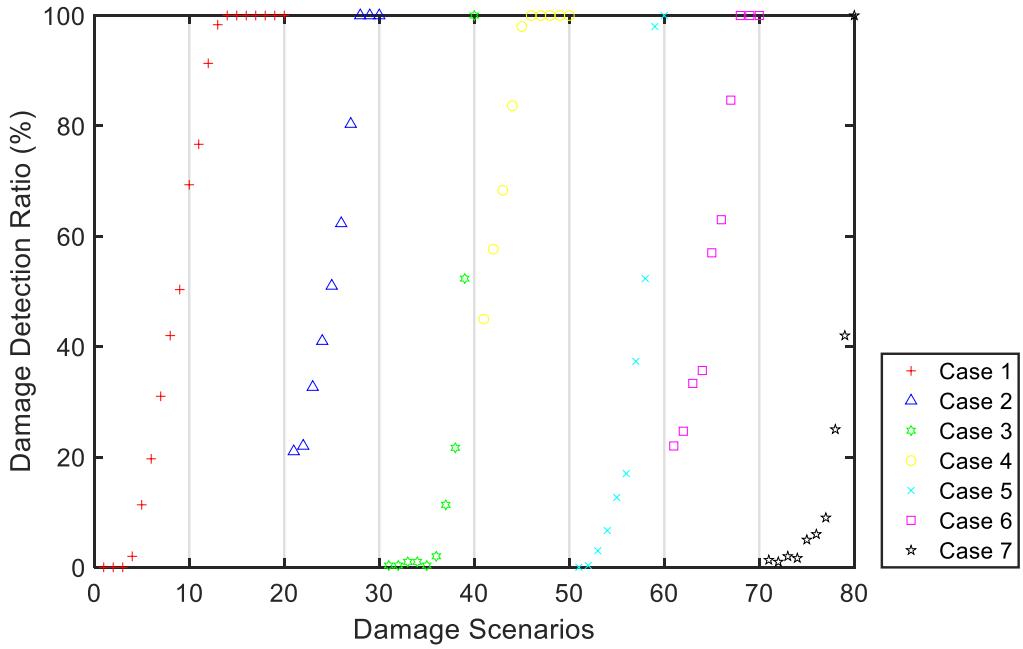


Figure 4.15 : Damage detection ratios via damage detection feature (6).

The result of damage detection ratios that using natural system frequencies as damage sensitive feature is shown in Figure 4.15. Damages were detected 100% by one-class classifier created with Feature (6) at the last step of damage scenario.

The graphs given above are presented to evaluate each damage detection feature used in the one-class classification to draw meaningful conclusions for a specific feature. In order to compare the models with each other, the results for the initial and final damage levels are given in Figure 4.16 through Figure 4.22. The initial damage level corresponds to one percent reduction in the elastic modulus of the material for Damage Case 2, ten percent reduction in the of rigidity of the connection for other damage cases. The final damage level corresponds to eighty one percent reduction in the elastic modulus of material for Damage Case 2, the loss of total rotational rigidity of the connection for other damage cases.

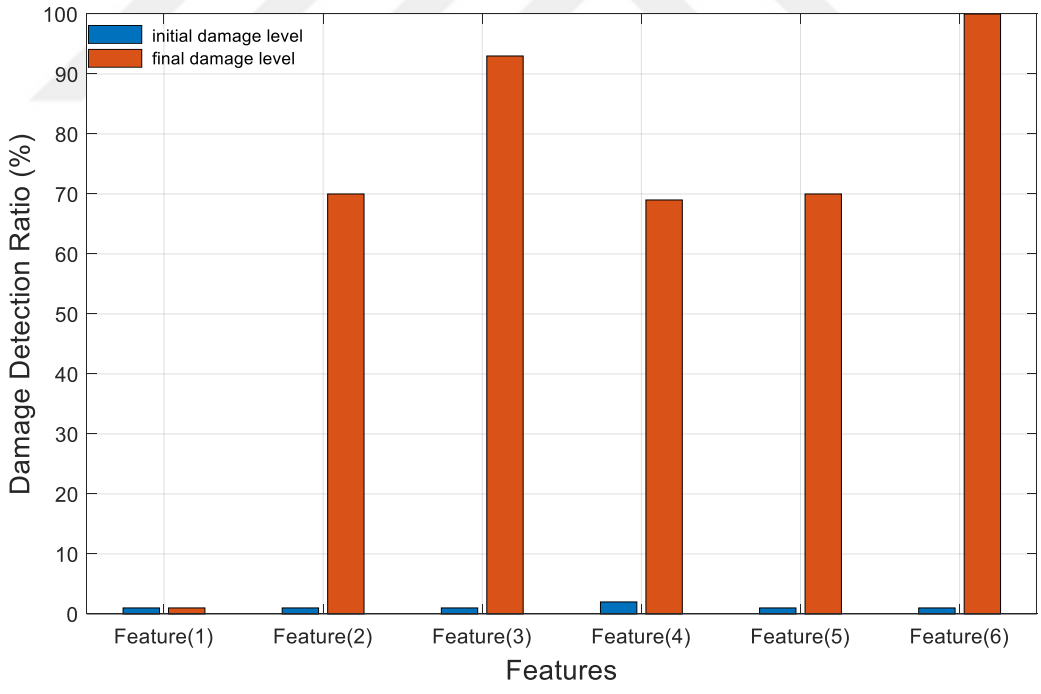


Figure 4.16 : Compression of damage detection ratios for damage case (1).

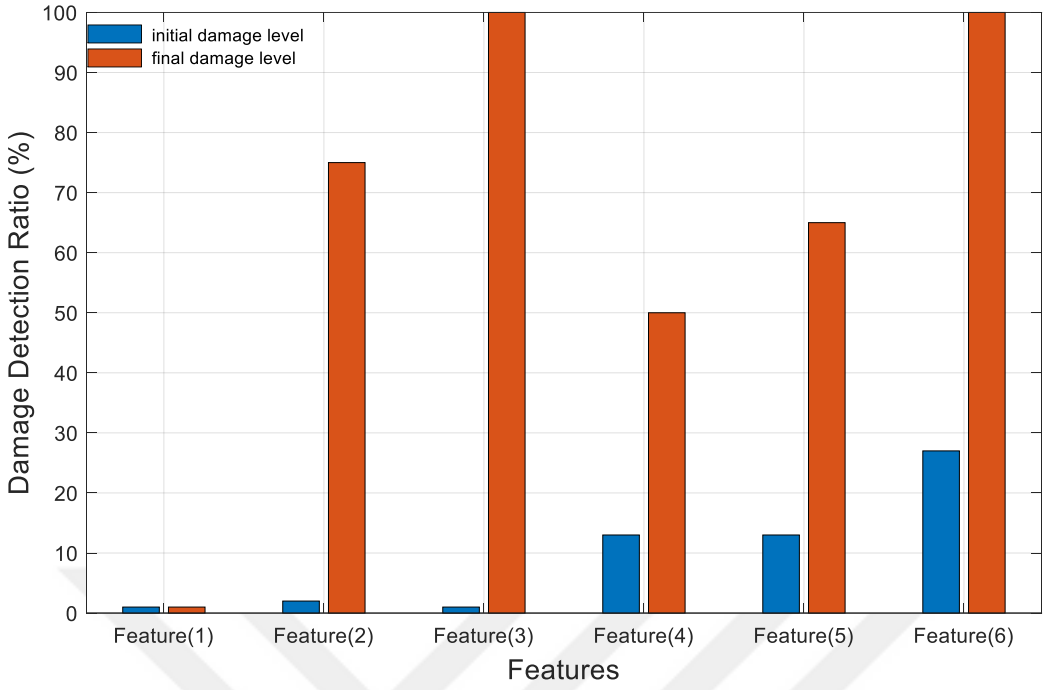


Figure 4.17 : Compression of damage detection ratios for damage case (2).

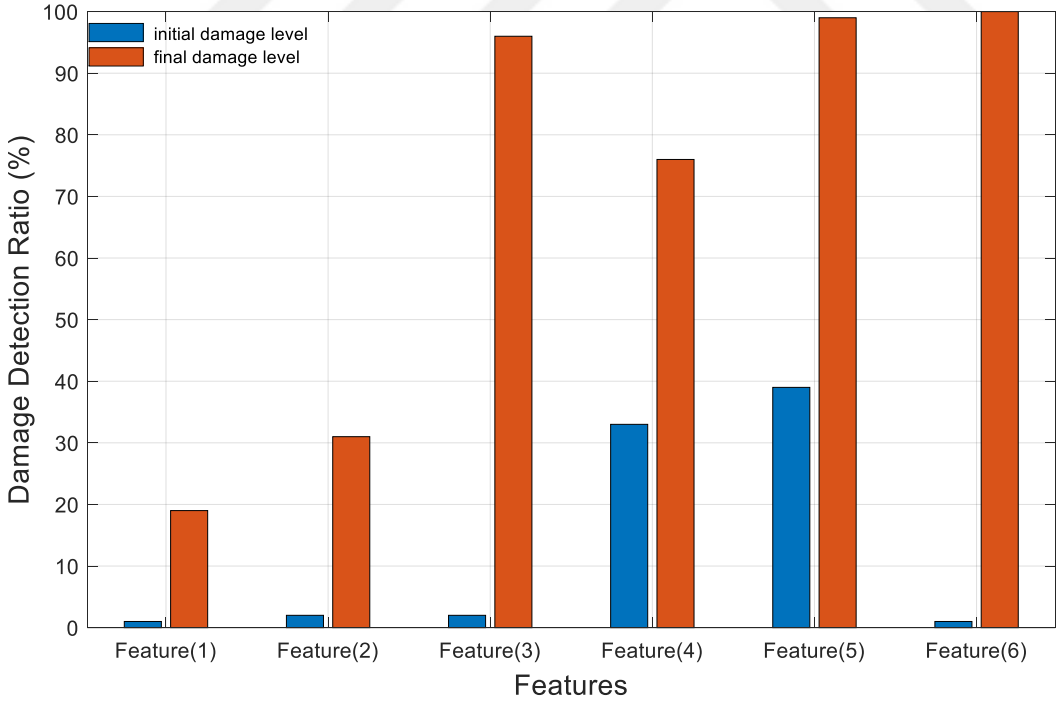


Figure 4.18 : Compression of damage detection ratios for damage case (3).

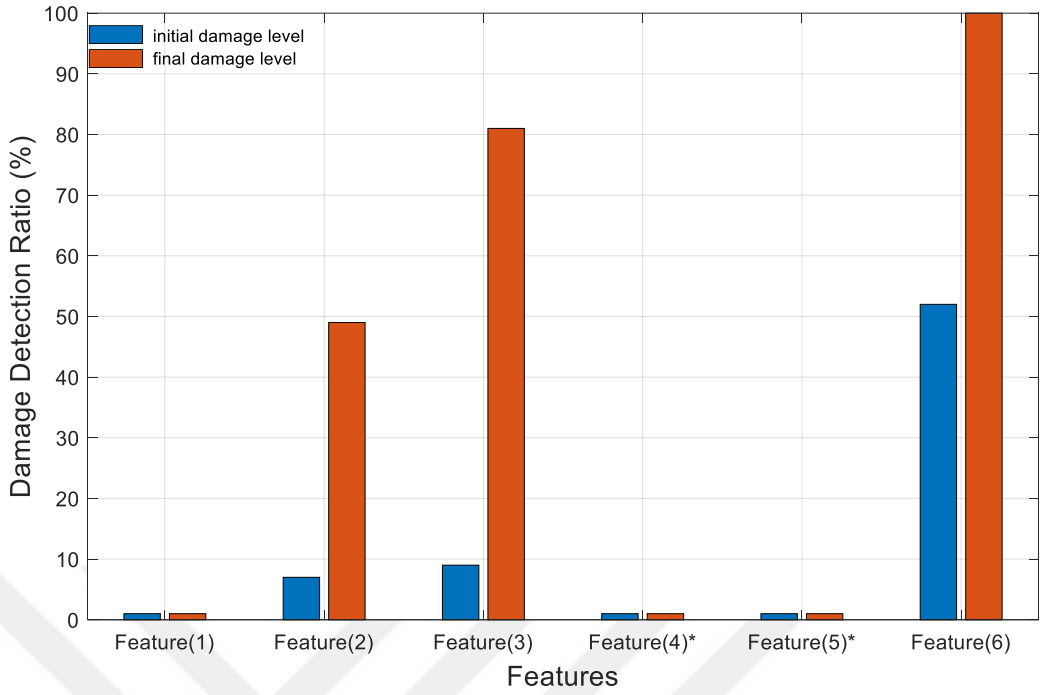


Figure 4.19 : Compression of damage detection ratios for damage case (4).

*The results of Feature (4) and Feature (5) were not included in the figure due to their unsatisfactory performance in the individual examination of the features as discussed above.

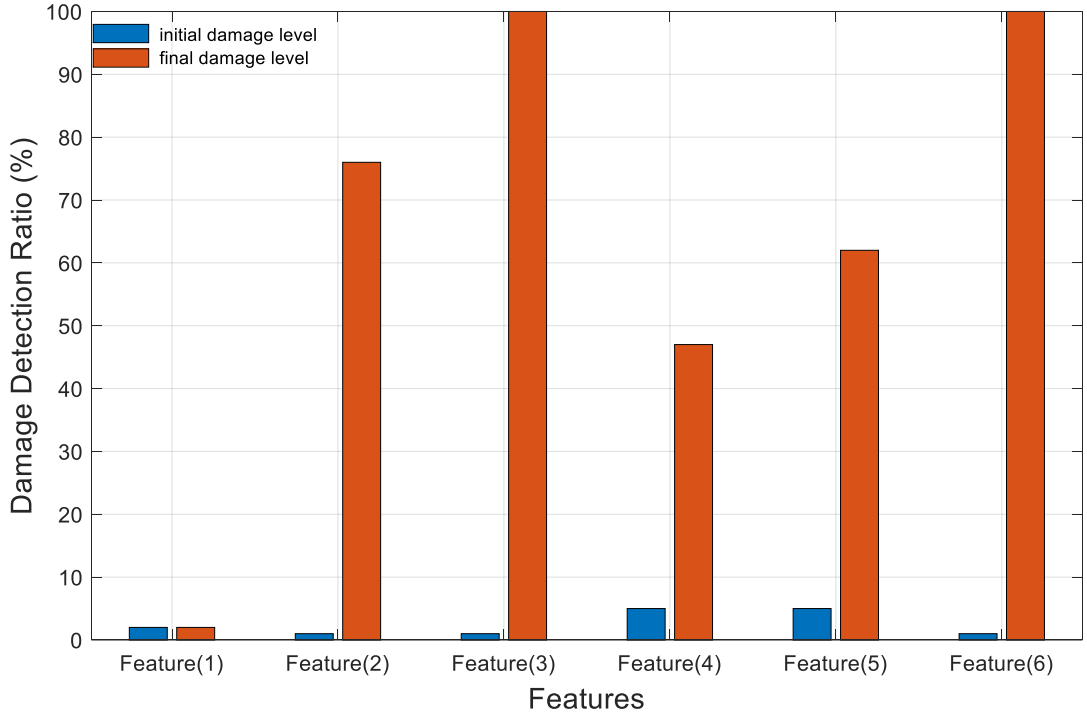


Figure 4.20 : Compression of damage detection ratios for damage case (5).

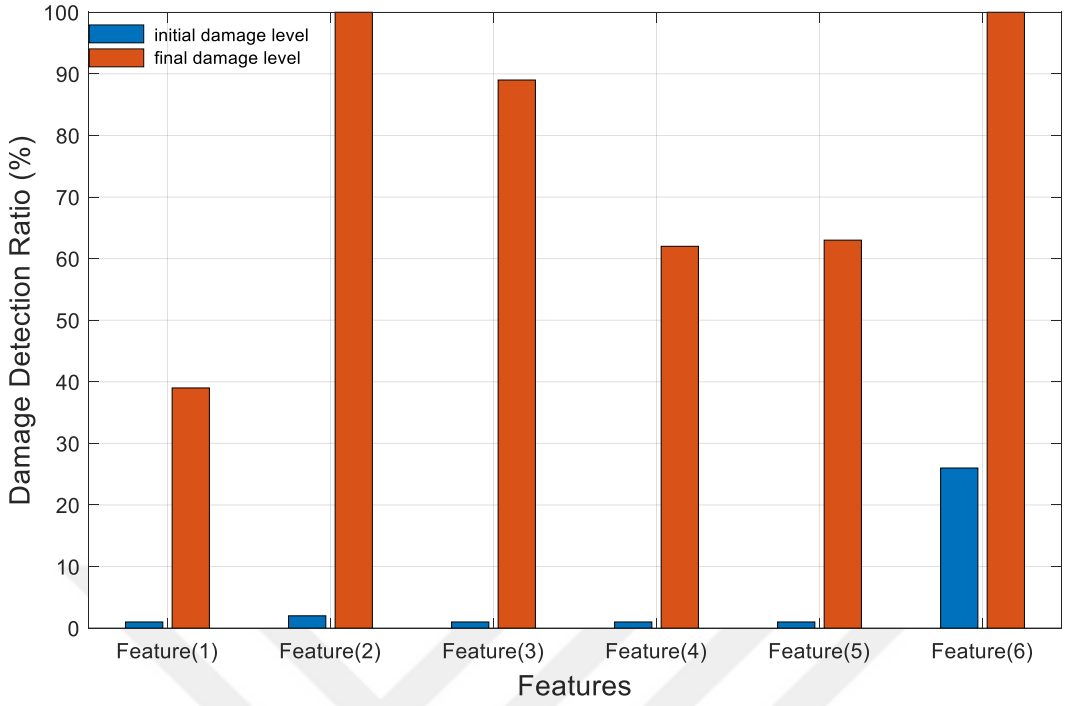


Figure 4.21 : Compression of damage detection ratios for damage case (6).

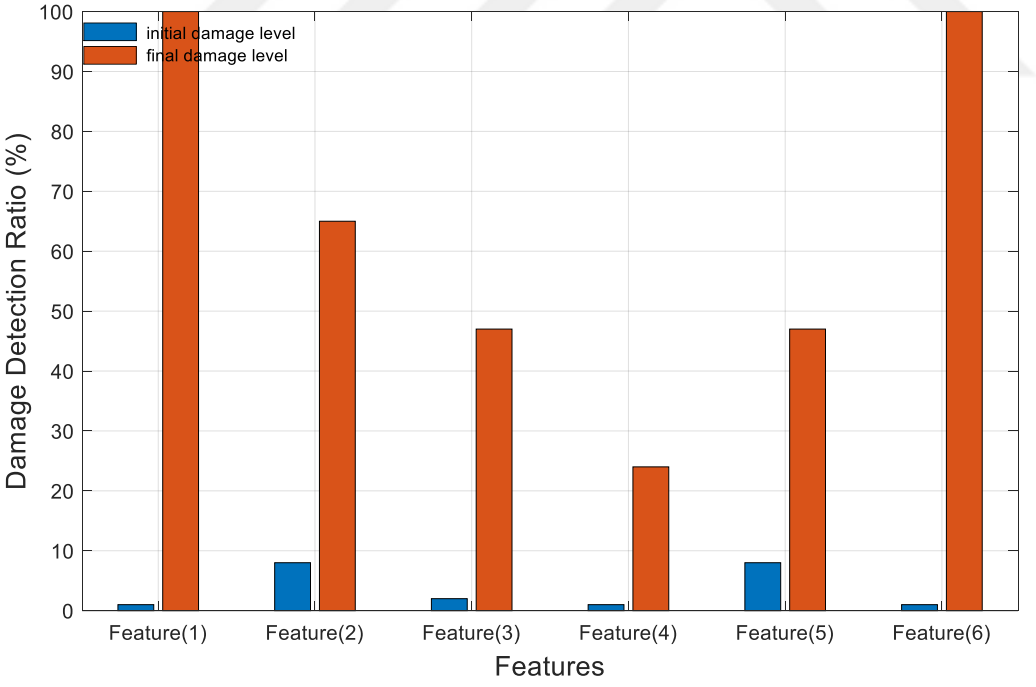


Figure 4.22 : Compression of damage detection ratios for damage case (7).

As can be seen from the figures, the detection performance of Feature (6) appears to produce the best results to detect damage. The damage detection performance of one-class classification model created with Feature (3) is also found satisfactory. For

detecting local damage as represented by reduction in the rigidity of the beam-column connection, the one class classification models created via ARX coefficients of the node close to the damaged connection also performed well. The residual mean errors or standard deviations as classifier gives satisfactory results in reducing the rigidity of the beam-column connection are not found satisfactory in detecting the local damage explored in this study.

5. CONCLUSIONS

In this thesis, different damage simulations were performed to test the efficiency of one class SVM models with several damage detection features. To consider measurement noise, accelerations obtained from each damage scenario were polluted artificially with random noise. Six different dynamic properties of structure were employed as the damage detection sensitive feature used in SVM model. Twenty-four thousand simulations obtained after pollution process was investigated to study the damage detection performance of the SVM models.

To compare the success of the damage detection for different types of damage, one global and six local damage cases was used. Damage Case (1) as global damage case, others as local damage cases can be considered. For global damage case, all created SVM models except the one using Feature (1) was able to detect all damage simulations after certain damage levels. Among the local damage cases, reduction in the rigidity of connection of the second-floor beam or column, Damage Case (3) and Damage Case (6) had highest damage detection ratio for most SVM models.

To compare the success of the damage detection based on feature used by SVM model, six different damage detection features are used. From SVM models created by using ARX parameters, SVM model trained with ARX parameters of top floor sensor appears to produce the best results. Using ARX parameters of bottom floor sensor as damage sensitive feature has failed to detect damage for almost all damage cases. SVM model created with Feature (2) has yielded satisfactory results in especially global damage and local damages at second-floor. SVM model trained with ARX parameters of top floor sensor was able to accurately classify the data of the damage simulations after a certain damage levels, except for the damage at the bottom-floor column connection. Although Feature (4) and Feature (5), mean errors and standard deviations of errors, used by SVM model achieved over 70% success in detecting global damage, they did not yield reliable results in detecting local damage. SVM model created with the first natural frequencies of structure resulted in the best performance of to the damage detection ratios compared to the all other features.

Investigating the results of the SVM model using local features, it can be seen that the location where the damage occurred is obviously important in the damage detection performance of the feature. For example, while SVM model created with Feature (2) detected 100% damage for Damage Case (6) which occurred close to node (2), SVM Model created with Feature (3) detected 100% damage for Damage Case (5) which is damage case close to this feature.

Although the binary decision returned for each simulation only provides information regarding the existence of damage, using localized information in the damage detection stage may further be utilized in the damage assessment problem for the spatial distribution of damage.

Considering the results in general, increased damage detection with damage severity applied to the structure at every step and the success of the features of the upper floors which are expected to be more affected by the damage, showed that the proposed methodology is compatible with the engineering knowledge. Nevertheless, since all the data used is obtained in a virtual environment with finite element method, the study should be tested in laboratory environment and full-scale field experiments.

REFERENCES

Bridgelall, R. (2017). Introduction to Support Vector Machines. Retrieved February 2, 2017, from https://www.ugpti.org/smartse/resources/downloads/support-vector-machines.pdf

Cao, M. S., Sha, G. G., Gao, Y. F. and Ostachowicz, W. (2017). Structural damage identification using damping: a compendium of uses and features. *Smart Materials and Structures*, 26(4).

Doebling, S. W., Farrar, C. R., Prime, M. R. and Shevits, D. (1996). Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review. *Los Alamos National Laboratory*, 1-136.

Doebling, S. W., Farrar, C. R. and Prime, M. R. (1998). A summary review of vibration-based damage identification methods. *The Shock and Vibration Digest*, 30(2), 91-105.

Fan, F. and Qiao, P. (2011). Vibration-based damage identification methods: a review and comparative study. *Structural Health Monitoring*, 10(1), 83-111.

Frigui, F., Faye, J. P., Martin, C., Dalverny, O., Peres, F., Judenherc, S. (2018). Global methodology for damage detection and localization in civil engineering structures. Retrieved from http://oatao.univ-toulouse.fr/21553.

Gunn, S. R. (1998). Support Vector Machines for Classification and Regression. Technical Report. Image Speech and Intelligent Systems Research Group, University of Southampton. Retrieved May 14, 1998, from https://svms.org/tutorials/Gunn1998.pdf

Ghaboussi, J., Garrett Jr, J. H. and Wu, X. (1991). Knowledge-based modeling of material behaviour with neural networks. *Journal of Engineering Mechanics*, 117(1), 132-153.

HoThu, H., Mita A. (2013). Damage detection method using support vector machine and first three natural frequencies for shear structures. *Open Journal of Civil Engineering*, 03(02), 104-112.

James, L., Büyüköztürk, O. (2016). Decentralised one-class kernel classification-based damage detection and localisation. *Structural Control and Health Monitoring*, .

Juand, J.N., Phan, M. Q. (2004). Identification and Control of Mechanical Systems. Cambridge, Cambridge University Press.

Juang, J. N. (1994). *Applied System Identification*. Prentice-Hall, Englewood Cliffs.

Li, X., Liu, B. (2003). Learning to classify texts using positive and unlabeled data. *In Proceedings of 18th International Joint Conference on Artificial Intelligence*, Mexico, 587–594.

Limongelli, M. P. (2019). Vibration-based structural health monitoring: challenges and opportunities. *Proceedings of the 7th International Conference on Structural Engineering, Mechanics and Computation*, South Africa, 2-4 September.

McCormick, A. C. and Nandi, A. K. (1997). Real-time classification of rotating shaft loading conditions using artificial neural networks. *IEEE Transactions on Neural Networks*, 8(3), 748-757.

Mehrjoo, M., Khaji, N., Moharrami, H. and Bahreinninejad, A. (2008). Damage

detection of truss bridge joints using Artificial Neural Networks. *Expert Systems with Applications*, 35(3), 1122-1131.

Rytter, A. (1993). *Vibrational based inspection of civil engineering structures* (Ph.D. thesis). Aalborg University, Department of Building Technology and Structural Engineering, AALBORG.

Samanta, B., Al-Balushi, K. R., Al-Araimi, S. A. (2003). Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection. *Engineering Applications of Artificial Intelligence*, 16(7-8), 657-665.

SAP2000 (Version 15.2.1) [Computer Software]. *Integrated software for structural analysis and design program*, Computers and Structures Inc., Berkeley, California.

Schölkopf, B., Plat, J. C., Shawe-Taylor, J., Smola, A. J., Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. , *Neural Comput*., *13*(7), 1443-1471.

Shimada, M., Mita, A. and Feng, M. Q. (2006). Damage detection of structures using support vector machines under various boundary conditions. *Proceedings of SPIE - The International Society for Optical Engineering*, United States, 27 February-2 March, 6174(2).

Shin, H. J., Eom, D. H., Kim, S. S. (2005). One-class support vector machines-an application in machine fault detection and classification. *Computer and Industrial Engineering*, 48, 395-408.

Tax, D. M. J., Duin, R. P. W. (2004). Support vector description. *Machine Learning, 54, 45-66.*

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. New York, Springer-Verlag.

William, C., and Salawu, O. S. (1997). Damping as a damage indication paramater. *In Proceedings of the 15th International Modal Analysis Conference*, Japan, 1-4 September, pp. 1531-1536.

Worden, K., Lane, A. J. (2001). Damage identification using support vector machines. *Smart Materials and Structures*, 10(3), 540-547.

Yan, Y. J., Cheng, L., Wu, Z. Y. and Yam, L. H. (2007). Development in vibration-based structural damage detection technique. *Mechanical Systems and Signal Processing*, 21, 2198-2211.

Yan, Y. J., Cheng, L., Wu, Z. Y. and Yam, L. H. (2007). Development in vibration-based structural damage detection technique. *Mechanical Systems and Signal Processing*, 21, 2198-2211.

CURRICULUM VITAE

Name Surname	**: Efkan OK**
Place and Date of Birth	**: Istanbul/27.01.1992**
E-Mail	**: efkannok@gmail.com**
B.Sc.	**: Istanbul Technical University**