Improved Traffic Crash Modeling through Accuracy and Response Time Using Classification Algorithms: A Model Comparison Approach

Iman Aghayan

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Doctor of Philosophy
in
Civil Engineering

Eastern Mediterranean University
January 2013
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Civil Engineering.

Asst. Prof. Dr. Mürüde Çelikağ
Chair, Department of Civil Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Civil Engineering.

Asst. Prof. Dr. Mehmet Metin Kunt
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Adham Mackieh
2. Assoc. Prof. Dr. Mustafa Gürsoy
3. Assoc. Prof. Dr. Umut Türker
4. Asst. Prof. Dr. Giray Özay
5. Asst. Prof. Dr. Mehmet Metin Kunt

ABSTRACT

This research focuses on predicting the severity of freeway traffic crashes by employing two different dataset including Iranian and Cyprus data. In Iranian data, twelve variables related to crash parameters were used by considering genetic algorithm, combined genetic algorithm and pattern search, and artificial neural network methods. The genetic algorithm evaluated eleven equations to obtain the best equation, and then the genetic algorithm and pattern search methods were combined using the best genetic algorithm equation. The neural network used a multi-layer perceptron architecture that consisted of a multi-layer feed-forward network with hidden sigmoid and linear output neurons that can also fit multi-dimensional mapping problems arbitrarily well. In Cyprus data, seven variables were selected to compare two fuzzy clustering algorithms—fuzzy subtractive clustering and fuzzy C-means clustering— with a multi-layer perceptron neural network. Four clustering algorithms—hierarchical, K-means, subtractive clustering, and fuzzy C-means clustering—were used to obtain the optimum number of clusters based on the mean silhouette coefficient and R-value before applying the fuzzy clustering algorithms.

The selected models used in Iranian and Cyprus dataset were able to predict the severity of crash injuries and to estimate the response time on the traffic crash data in which the prediction accuracy was determined according to R-value, root mean square errors, mean absolute errors, and sum of square error.

Based on the results obtained from Iranian data, the highest R-value and the highest amount of time were obtained for the artificial neural network around 0.87 and 7.627 seconds, respectively. The results demonstrated that the artificial neural

network provided the best prediction accuracy with highest response time, while genetic algorithm had the lowest value for prediction accuracy (0.79) and response time (0.687) among the applied models. The combination of the GA and PS methods allowed for various prediction rankings ranging from linear relationships to complex equations.

Based on the results obtained from Cyprus data, the highest R-value and the highest amount of time were obtained for the multi-layer perceptron around 0.89 and 2.635, respectively demonstrating that the multi-layer perceptron had a high accuracy in traffic crash prediction among the prediction models, and that it was stable even in the presence of outliers and overlapping data. Meanwhile, in comparison with other prediction models, fuzzy subtractive clustering provided the lowest value for response time (0.284), 9.28 times faster than the time of multi-layer perceptron.

Overall, the results showed that the MLP can be the best model to predict the traffic crash severity regardless of the variables involved with crash data in which the accuracy was the important criterion. Meanwhile, more than one model can be appropriate according to the determined criteria. Considering prediction accuracy and response time could lead to developing an on-line system for processing data from detectors and/or a real-time traffic database as well as the system may be implemented in an incident management to prevent the traffic crash or secondary traffic crash in which the model can be extended through improvements based on additional data through induction procedure.

Keywords: Accuracy, Classification algorithms, Prediction, Response time, Traffic crash severity.

ÖZ

Bu araştırma, İran ve Kıbrıs verileri olmak üzere iki farklı veri seti kullanılarak otoyol trafik kazaları ciddiyetinin tahmininde odaklanmıştır. İran verileri, çarpışma parametreleri ile ilgili oniki değişkene, genetik algoritma, kombine genetik algoritma, ve yapay sinir ağları yöntemleri dikkate alınarak kullanılmıştır. Genetik algoritma uygulamasında en iyi denklemi elde etmek için onbir denklem değerlendi. Sonra genetik algoritma ve desen arama yöntemleri en iyi genetik algoritma denklemi kullanılarak birleştirildi. Sinir ağı da çok boyutlu haritalama sorunlarını da rastgele modelleme yapabileceği gizli sigmoid ve lineer çıkış nöronlar ile çok katmanlı ileri beslemeli ağ oluşur ve çok katmanlı algılayıcı mimarisi ile kullanıldı. Kıbrıs verileri, yedi değişkenin iki bulanık kümeleme algoritmaları-bulanık eksiltici kümeleme ve bulanık C-aracı birçok katmanlı algılayıcı sinir ağı kümeleme ile karşılaştırmak için seçilmiştir. Dört kümeleme algoritmaları-hiyerarşik, K-means, eksiltici kümeleme ve bulanık C-means kümeleme ile elde edildi ve bulanık kümeleme algoritmaları uygulamadan önce ortalama siluet katsayısı ve R-değeri esas alınarak kümelerinin optimum sayıda elde etmek için kullanılırdı.

İran ve Kıbrıs verileri için kullanılan seçili modellerin, kazalarda yaralanma şiddetini tahmin etme doğruluğu ve tahmin sürelerinin tasbiti yapıldı. Tahmin doğruluğu R-değerine göre kararlı olan, kök, hata karelerinin ortalamalarının mutlak hataları anlama ve hata kareler toplamıdır.

İran verileri, yüksek R-değeri ve zaman en yüksek miktarda elde edilen sonuçlara dayanarak yapay sinir ağı için elde edildi ve bunlar sırasıyla 0.87 ve 7,627. Sonuçlar yapay sinir ağının yüksek tepki süresi ile en iyi tahmin doğruluğu sağladığını göstermiştir, genetik algoritma uygulanan modeller arasında tahmin doğruluğu (0.79)

ve tepki süresi (0,687) için en düşük değerdir. Doğrusal ilişki karmaşık denklemleri kadar çeşitli tahmini sıralaması için GA ve PS yöntemlerinin kombinasyonu kullanılmıştır.

Kıbrıs verilerinde, R-değeri yüksek zaman ve en yüksek miktarda elde edilen sonuçlara göre, çok-katmanlı algılayıcı ile elde edilmiştir. Çok katmanlı algılayıcı tahmin modellerinin yanında trafik kazasında tahmini yüksek doğruluğu taşıdığını gösteren sırasıyla 0.89 ve 2.635, çevresindeki hatta sapan ve üst üste gelen verilerin mevcudiyetinde bile kararlı oldu gözlemlenmiştir. Bu arada, diğer tahmin modelleri ile karşılaştırıldığında, bulanık kümeleme eksiltici ve düşük değer sağlanan çok katmanlı algılayıcı süresinden daha hızlı tepki süresi gerektirmiş (0.284), 9.28 kat kadardır.

Genel olarak, tahmini doğruluk ve tepki süresi dikkate alındığında verilerin işlenmesi için bir gerçek zamanlı sistemi geliştirmek için olabilir ayrıca detektörlerden gelen ve gerçek zamanlı trafik veri tabanı oluşturulduğunda hem kaza yönetim sisteminin çökmesi ya da ikinci kaza oluşumunu önleyebilir. Geliştirilen modelde indüksiyon prosedürü aracılığıyla ek verilere dayanarak iyileştirmeler yapılabilir ve bu yol ile uygulama aralığı geliştirebilir.

Anahtar Kelimeler: Doğruluk, Sınıflandırma algoritmaları, Tahmin, Tepki süresi, Trafik kaza şiddeti.

My parents deserve a heartfelt thank you. Without your hard work, I couldn't have the opportunity to pursue higher education and study abroad. Your love, support, encouragement helped me reach my dream. I love you!

ACKNOWLEDGMENT

I am so appreciative of my supervisor Asst. Prof. Dr. Mehmet Metin Kunt for his patient guidance and encouragement throughout this study. His experience and knowledge have played an important role in my research.

I could not forget to state the invaluable supports of my friend, Nima Noii who his friendship and exchange of opinions helped in this investigation. And I would like to thank my dear friend, Kani Akhavan Kazemi, for her support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABS……...Anti-lock Braking System

ANNs……...……....Artificial Neural Networks

EAs……..Evolutionary Algorithms

FCM……..……Fuzzy C-means

FIS……Fuzzy Inference System

FS……...…….Fuzzy Subtractive

GA…….…....Genetic Algorithm

GIS……...…Geographic Information System

GOF…….…...Goodness of Fitness

GP……..……..Genetic Programming

GPS……..Generalized Pattern Search

KMCAW……..…K-means Clustering According to Attribute Weighting

LMS……....Least Mean Square Error

MADS…….…Mesh Adaptive Search

MAE…….….Mean Absolute Error

MF……..……Membership Function

MLP…….……Multi-Layer Perceptron

MSE……....Mean Squared Error

PNN…….....Probabilistic Neural Network

PS…….Pattern Search

RMSE……..Root Mean Square Error

SSE……..…Sum of Squared Error

SSR……..….Sum of Squared Residuals

TSK……..….Takagi-Sugeno-Kang

ZINB…….…..Zero Inflated Negative Binomial

ZIP ……..….Zero Inflated Poisson

LIST OF ALGORITHMS

Chapter 1

INTRODUCTION

1.1 Background

As the world population grows and as cars become increasingly common, the number of traffic crashes worldwide is increasing. In recent years, a dramatic increase in traffic crashes worldwide has brought the problem of improving traffic safety to the attention of health officials who now approach the problem as they would a biological disease. More than 28,000 and 40 people are killed per year on Iranian and Cyprus roads with economic and social consequences, respectively. According to the World Health Organization (WHO, 2004), worldwide motor vehicle crashes are the second most frequent cause of death for people 5-29 years old. As summarized by WHO, "an estimated 1.2 million people are killed each year in road crashes and as many as 50 millions are injured. Projections indicate that these figures will increase by about 65% over the next 20 years unless there is new commitment to prevention". Traditional measures to reduce crashes include improved geometric design, congestion management strategies and better driver education and enforcement. While these measures are generally effective, they are often not feasible or prohibitively expensive to implement. Many factors are involved in traffic crashes, and some of these factors have profound impacts on one other, preventing transportation safety designers from using only one parameter to fully explain traffic crash severity. Studying the parameters involved in traffic crashes together using modern models that include the interactions of input and

output variables can lead to a decrease in the number of traffic crashes. The relationship between a crash and the influencing factors is nonlinear and complicated; thus, it cannot be described with an explicit mathematical model. The crash prediction model (also called the safety performance function) is one of the most important techniques to investigate the relationship between crash occurrence and risk factors associated with various traffic entities. Factors with a profound impact on traffic crash severity include the demographic or behavioral characteristics of the driver (vehicle speed, driver's age, driver's gender, seat belt use, alcohol involvement), environmental factors and roadway conditions at the time of the crash (crash time, weather conditions, road surface, crash type, collision type, traffic flow, trafficway character) and the technical characteristics of the vehicle itself (vehicle type, safety of the vehicle).

1.2 Objectives of the study

The primary goal of this study is to compare various models and select the most suitable model in order to predict traffic crash severity and estimate response time on two different traffic crash datasets (Iranian and Cyprus data). It means that the most suitable prediction model from among the tested models is determined based on two criteria: accuracy (R-value, root mean square (RMSE), mean absolute errors (MAE), and sum of square error (SSE)) and response time (t). The accuracy factor establishes if a model is able to accurately predict traffic crash severity, while the response time establishes if the model can produce results in a reasonable period of time. Thus, prediction models are chosen based on having the highest accuracy and the lowest response time. In addition, another purpose of this study is to create a model that can be updated with additional data beyond whatever is previously used so that the prediction models are improved based on new information through induction

procedure. Finally, the system can prevent the traffic crash or secondary traffic crash by using real-time traffic dataset and detectors.

1.3 Works Undertaken

In connection with Iranian data, driver's age, driver's gender, use of seat belt, type of vehicle, safety of vehicle, weather condition, road surface, speed ratio, crash time, crash type, collision type and traffic flow were selected as input variables and three output variables consisted of no injury, evident injury, and disabling injury/fatality. Three modeling techniques were applied to Iranian and Cyprus data to achieve a high predictive accuracy and low response time. The first model was an artificial neural network (ANN) which was able to capture highly nonlinear relationships between the predictor variables (crash factors) and the target variables (severity level of the injuries). Neural networks can be useful particularly when the relationship between the variables is unknown or complex; therefore, it is difficult to be handled, statistically. A neural network is composed of simple elements operating in parallel, as found in biological nervous systems. As in nature, the connections between the elements largely determine the network function. In this study, a multi-layer perceptron (MLP) neural network architecture that is consisted of a multi-layer feed-forward network with sigmoid hidden neurons and linear output neurons was used. The second model was a genetic algorithm (GA), which is used to solve both constrained and unconstrained optimization problems based on natural selection, the process that drives biological evolution. The third investigated model was a model combining the GA and pattern search (PS) models. The use of GA-PS models in transportation safety studies is relatively new, so we test combining these methods in order to improve prediction accuracy.

In connection with Cyprus data, seven variables were selected as input for such a model: driver's gender, driver's age, crash time, type of vehicle, weather conditions, trafficway characteristic, and collision type. The selected output variable was the injury severity, which consisted of three levels: no injury, evident injury, and disabling injury/fatality. Two fuzzy clustering algorithms- fuzzy C-means (FCM) and fuzzy subtractive (FS) clustering- and a multi-layer perceptron were used to determine the suitability of those input variables and injury severity levels for model predictions. Clustering techniques focus on obtaining useful information by the grouping of multi-dimensional data into clusters. In this study, four clustering algorithms—hierarchical, K-means, subtractive clustering, and FCM clustering—were used to obtain the optimum number of clusters before conducting an analysis with the fuzzy clustering algorithms. The suitability of the input and output variables was determined with fuzzy inference system (FIS) using FCM clustering based on Takagi-Sugeno-Kang (TSK) and Mamdani. The FCM clustering model optimizes the objective function to obtain the membership degree for each sample point relative to all the cluster centers. Then it determines the generic of the sample points, and finally it achieves automatic classification for data samples. A FIS using subtractive clustering based on the TSK-FIS structure was considered as the prediction model. The aim of subtractive clustering was to estimate both the number and initial locations of cluster centers and to extract the TSK fuzzy rules from the input/output variables.

Chapter 2

LITERATURE REVIEW AND BACKGROUND

Previous studies have focused on identifying a defensible statistical relationship between crash counts and exposure. The common models studied in traffic safety are the traditional Poisson and Poisson-gamma models. Those models are applied for modeling discrete, independent, and non-negative events. The negative binominal (NB) model arises mathematically (and conveniently) by considering presumably that unobserved crash heterogeneity (variation) across sites (intersections, road segments, etc.) is Gamma distributed while crashes within sites are Poisson distributed (Washington et al., 2003). The Poisson and Poisson-gamma models have been used for predicting motor vehicle crashes (e.g., Donnell and Mason, 2006; Lord, 2008). Bayesian empirical methods have also been developed (Mahal et al., 1982; Ng and Sayed, 2004; Wright et al., 1988). Other statistical models applied to accident data include the following: binomial, zero-inflated Poisson (ZIP), zero-inflated negative binomial (ZINB), and multinomial probability models. Lord et al. (2005) used multinomial probability models to conduct an analysis of the relationships among crash, density (vehicles per kilometer per lane), and the volume/capacity (v/c) ratio. The authors found that, with an increasing v/c ratio, fatal and single-vehicle crashes decreased after a certain point, and crash rates followed a U-shaped relationship. Bedard et al. (2002) applied a multivariate logistic regression analysis to investigate the effects of driver, crash, and vehicle characteristics on fatal crashes. Valent et al. (2002) used a logistic regression technique to evaluate the

relationship between driver characteristics and injury severity. The effects of road geometry and traffic characteristics on crash rates for rural two-lane and multilane roads were investigated by Karlaftis and Golias (2002) according to hierarchical tree-based regression (HTBR). Huang and Abdel-Aty (2010) used Bayesian analysis in traffic safety in which they have conducted some improvement on model fitting and the accuracy of prediction for the multi-level data structure.

ANNs have been verified to be efficient in many fields. Neural networks are commonly used in non-linear modeling and forecasting. In traffic safety, some studies have applied ANN to the prediction of crash rates and to the analysis of crashes, but none have used from twelve parameters related to Iranian data, including important factors with detail. Thus, this study attempted to incorporate all relevant parameters into the models to achieve a high percentage of crash forecasting. Mussone et al. (1999) applied ANNs to analyze vehicular crashes that occurred at an intersection in Milan, Italy. A number of studies have been conducted to investigate the groups of drivers with high risk of being injured or killed in traffic crashes (Zhang et al., 2000; Valent et al., 2002). Bedard et al. (2002) used multivariate logistic regression model to study the effects of driver, crash and vehicle characteristics on fatal crashes. Ivan et al. (2000) investigated single and multi-vehicle highway crash rates and their relationships with traffic density while controlling for land use, time of day and light conditions. Lord et al. (2005) conducted an analysis on the relationship among crash, density (vehicles per km per lane) and v/c ratio. They found that with increasing v/c ratio, fatal and single-vehicle crashes decreased after some point, and crash rates followed a U-shaped relationship. In the transportation field, ANNs have been applied to traffic flow prediction (Yin et al., 2002; Zhong et al., 2004), estimation of discharge headway (Tong and Hung,

2002), ramp control strategy (Zhang et al., 2001), incident detection (Jin et al., 2002; Yuan and Cheu, 2003), travel behavior analysis (Subba Rao et al., 1998; Hensher and Ton, 2000; Vythoulkas and Koutsopoulos, 2003) and traffic accident analysis (Mussone et al., 1996; Mussone et al., 1999; Sohn and Lee, 2003; Abdel-Aty and Pande, 2005). ANNs have scarcely been used as a modeling approach in crash injury severity analysis. For instance, Abdelwahab and Abdel-Aty (2001) investigated ANNs to analyze the relationship between driver injury severity and crash factors related to the driver, vehicle, roadway, and environmental characteristics. They have attempted to classify a traffic crash into one of three injury severity levels using the readily available crash parameters. These authors limit their domain of study to two vehicle accidents that happened at intersections with signals. The predicting performance of MLP was compared with the performance of the ordered logit model. Their results showed that the MLP achieved better classification (correctly classifying 65.6 and 60.4% of cases for the training and testing phases, respectively) than the ordered logit model (correctly classifying 58.9 and 57.1% of cases for the training and testing phases, respectively). Abdel-Aty and Pande (2005) applied a probabilistic neural network (PNN) model to predict crash occurrence on the Interstate-4 corridor in Orlando, Florida. The average and standard deviation of speed around crash sites were extracted from loop data as input variables. The results of this analysis showed that at least 70% of the crashes can be correctly identified by the proposed PNN model. Delen et al. (2006) defended the use of ANN by pointing out the non-linear relationships between injury severity and the traffic accident factors. Kunt et al. (2011) used ANN, GA and GA combined with PS for predicting the severity of freeway traffic crashes.

GAs are powerful stochastic search techniques based on the principle of natural evolution. These algorithms were first introduced and investigated by John Holland (1975). According to Chang and Chen (2000), the regression models generated by genetic programming (GP) are also independent of any model structure. According to Deschaine and Francone (2004), GP is observed to perform better than classification trees with lower error rates, and GP also outperforms neural networks in regression analysis. Several studies (Park et al., 2000; Ceylan and Bell, 2004; Teklu et al., 2007) have used GP methods in traffic signal system optimization and network optimization.

Zadeh introduced fuzzy logic in the 1960s. There are a series of justifications for using fuzzy logic in the modeling of complex processes. Fuzzy set theory techniques have been used in crash prevention. Akiyama and Sho (1993) studied the traffic safety problem on urban expressways. Hadji Hosseinlou and Aghayan (2009) used fuzzy logic to predict the traffic crash severity on the Tehran-Ghom freeway in Iran. Fuzzy logic utilized for the control of traffic systems (Kamijo et al., 2000; Mussa and Upchurch, 2002; Lanser and Hoogendoorn, 2000; Niitymaki, 2001). The combination of fuzzy logic and neural network has been applied for incident detection on freeways by Ishak and Al-Deek (1998).

Cameron (1997) indicates that clustering methods are an important tool when analyzing traffic accidents as these methods are able to identify groups of road users, vehicles and road clusters which would be suitable targets for countermeasures. A combination of cluster analysis, regression analysis and Geographical Information System (GIS) techniques is used to group homogeneous accident data together to estimate the number of traffic accidents and assess the risk of traffic accidents in a study area (Ng et al., 2002). The results can be helpful authorities effectively to

allocate resources for improving the safety levels in those areas with high accident risk. In addition, the results provided information for urban planners to develop a safer city.

Ruspini (1969) was the first to propose fuzzy c-partitions as a fuzzy approach for clustering, and then the FCM algorithms were modified by Dunn (1974) and generalized by Bezdek (1981). In connection with FCM algorithm, Sugeno and Yasukawa (1993) determined the optimal number of clusters in the output space. Chen et al. (1998) suggested the data space that should be classified with regard to the input data in addition to linear relationships between input and output data. A feature weighted FCM based on feature selection methods and on competitive agglomeration were proposed by Wang et al. (2004) and Frigui and Nasraoui (2004), respectively. Aghayan et al., (2012) investigated FCM clustering based on clustering algorithms for traffic crash in Cyprus.

In connection with the subtractive clustering, Chiu (1994) introduced the subtractive clustering in which data points were selected for cluster centers in order to solve computational difficulties that can arise in mountain clustering when problem dimensions are suitably increased for handling large datasets. Yager and Filev developed the mountain method for estimating cluster centroids (Yager and Filev, 1994). Hayajneh and Hassan (2008) applied a FS clustering and a FIS based on Sugeno type in the drilling processes. Aghayan et al. (2012) applied FCM and FS clustering compared with ANN by considering accuracy and response time criteria, the results represented that ANN can be the appropriate model for prediction accuracy and the lowest response time was achieved by FS algorithm in comparison with the applied models.

One of the clustering methods is the K-means algorithm (MacQueen, 1967). K-means clustering is an unsupervised pattern classification method. Fukunaga (1990) used K-means clustering on continuous data. Pena et al. (1999) applied various methods for process of initializing in the K-means algorithm. The K-means algorithm performance is related to initial cluster centers; thus, Khan and Ahmad, (2004); Redmond and Heneghan, (2007) suggested an algorithm for K-means clustering to determine initial cluster centers. Polat et al. (2012) applied K-means clustering according to attribute weighting (KMCAW) and classifier models with the help of GIS.

Chapter 3

METHODOLOGY

3.1 Typical Steps in Designing a Model

3.1.1 Iranian Data

The principles of the models employed in this study are shown in Figure 1 in which the main part of that Figure is represented with the dashed line. According to Iranian data, MLP, GA, and combined GA and PS were compared to investigate the behavior of these models through accuracy and response time. Initially, the 1000 records collected from police records were used to construct objective functions for these models. Then, the models were able to modify the objective function with regard to each of the 1000 records, which were added to preliminary data. In addition, the optimum coefficients of the objective function (for the new records) were the initial optimum vector in the combined GA and PS models (for the last records). GA and PS methods were combined using the best GA equation. To achieve optimal results from the MLP model, new weights and biases were calculated from the preliminary weight matrix and bias vector. Finally, the errors of objective functions were calculated by these models, and the most appropriate error with respect to its type in each model was selected to determine the final objective function. The best-fit prediction algorithms between the MLP, GA, and GA-PS were selected based on two criteria: accuracy (R, RMSE, MAE, and SSE) and response time (t). Thus, it caused the suitable models by considering both mentioned criteria to be identified.

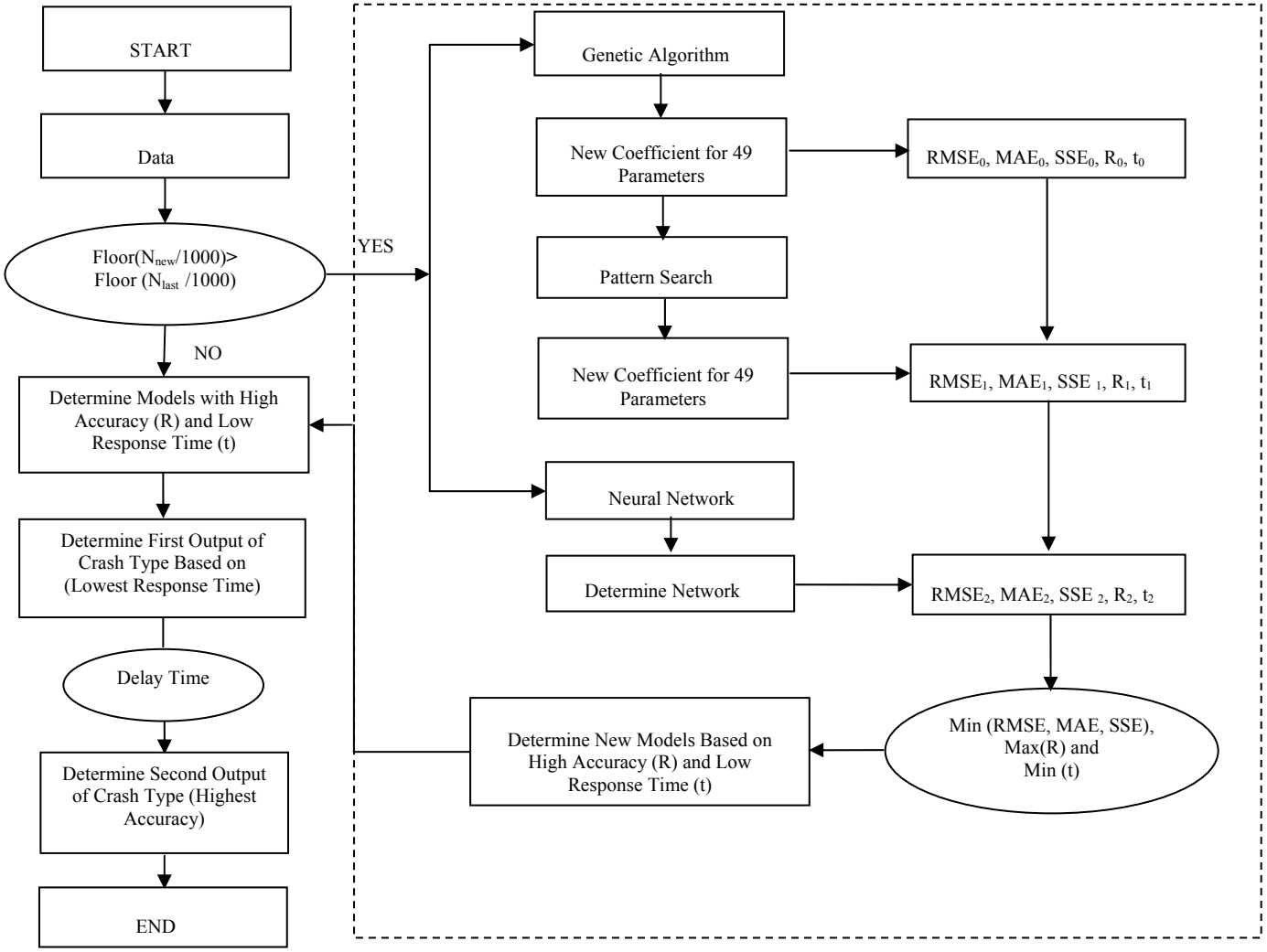

Figure 1: Flowchart for processes carried out in a typical run with Iranian data (Kunt et al., 2011)

3.1.2 Cyprus Data

In this study, a comparison of MLP with FCM clustering and FS clustering was performed by considering the optimum number of data cluster algorithms employed for improving the traffic crash prediction procedure by using of Cyprus data. The first modeling step was the training phase that used 70 percent of the data, and the other 30 percent of the data were used for model validation and testing to improve the model. The 1049 records collected from police records were used to construct the initial prediction model. However, before initiating the main part of flowchart shown in Figure 2 by the dashed line, the number of available data records was checked because the model can be updated with every batch of 1000 records. In other words, the model can be updated every additional 1000 records beyond the preliminary data. This means the model has the ability to improve itself with new data. Hierarchical, K-means, subtractive clustering and FCM clustering were employed for obtaining the optimum number of clusters based on mean silhouette coefficient and R-value.

Consequentially, the optimum number of clusters achieved before was used in FS and FCM. In addition, the best-fit prediction algorithms between the fuzzy clustering algorithms and MLP were selected based on two criteria: accuracy (R, RMSE, MAE, and SSE) and response time (t). This procedure led to identification of suitable models with respect to both accuracy (R) and response time (t). Finally, the suitable models were selected based on each criterion mentioned above.

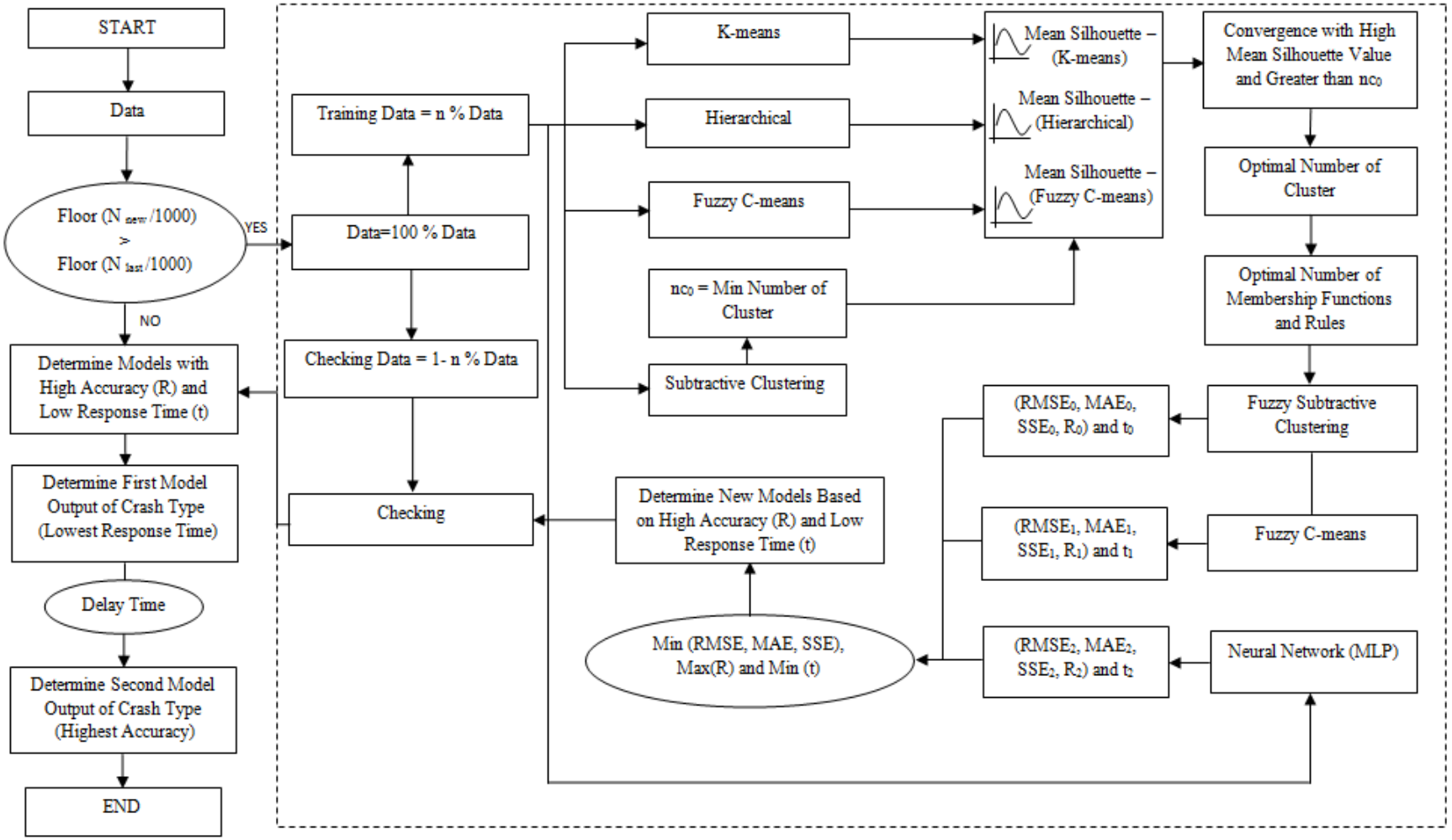

Figure 2: Flowchart for the processes in a typical run with Cyprus data (Aghayan et al., 2012)

Overall, by considering of Iranian and Cyprus data, if a fast prediction model was the goal, then this procedure identified the prediction model with the lowest response time, but if accuracy was the concern, then the procedure found the prediction model with the highest accuracy based on checking the data before the model started performing predictions. The first model output had the lowest response time, while the second model output, delayed by a few seconds, had the highest accuracy.

3.2 Data Description

3.2.1 Iranian Data

The dataset used in this study includes 1063 traffic crashes and was derived from reported traffic crashes in Tehran, the capital of Iran. These crashes were selected from the total number of crashes that occurred on the Tehran-Ghom freeway in 2007 because these were the only complete crash records. These data were used as training and testing data for the ANN, GA, and combined GA and PS methods, and the predictions of the three models were compared. The majority of crashes (74.8%) involved two vehicles. The distribution of driver injuries was around 14% fatal injuries, 38.4% evident injuries, and 47.6% no injuries.

Three injury levels were considered for this study (i.e., no injury, evident injury, and disabling injury/fatality), and twelve variables were selected from the data. The vehicle speed in police reports was calculated by a camera or the breaking distance. Speed ratio was used as one of the input variables defined as the ratio of the estimated speed at the time of a crash to the posted speed limit at the crash location. The safety of vehicle was categorized as high and low standard. High standard was used for vehicles having Anti-lock braking system (ABS) and airbag. Road geometry parameters were not taken into consideration because the selected road had a desirable geometry that is common to all crashes in the dataset. The input variables

have either numerical or dummy values for use in the program. Table 1 shows the input and output variables for Iranian data. Comparing the performance of the three modeling approaches discussed later (ANN, GA, and combined GA and PS) was obtained by using MATLAB software.

Table 1: Description of the study variables for Iranian data (Kunt et al., 2011)

Input Variables		**Variable**	**Coding/Values**	**Data**
Variables	Subdivided Variables			
1	2	Driver's Gender	Man Woman	97.56% 2.44%
2	1	Driver's Age	Year	20-34=39% 35-49=44% 50-64=10% 65-79=7%
3	2	Use of Seat Belt	In use Not in use	78.66% 21.34%
4	3	Type of Vehicle	Passenger car Bus Pick-up	83.54% 2.44% 14.02%
5	2	Safety of Vehicle	High standard Low standard	31.71% 68.29%
6	4	Weather Condition	Clear Snowy Rainy Cloudy	56.71% 7.93% 10.37% 25%
7	3	Road Surface	Dry Wet Snowy/Icy	75% 17.68% 7.32%
8	1	Speed Ratio	km/h/km/h	
9	2	Crash Time	Day Night	65.85% 34.15%
10	2	Crash Type	With vehicles With multiple vehicles	74.81% 25.19%
11	3	Collision Type	Rear-end Right-angle Sideswipe	51.95% 30.24% 17.80%
12	1	Traffic Flow	veh/h	
Output variables				
1	3	Driver Injury Severity	No injury=(1,0,0) Evident injury=(0,1,0) Fatality=(0,0,1)	47.56% 38.41% 14.02%

3.2.2 Cyprus Data

The dataset used in this study consists of 1049 traffic crashes and was derived from traffic crashes reported between 2005 and 2010 on the North Cyprus primary road network. The dataset includes only crash data that are complete with regard to all input variables that were used in this study. These data were used as training and testing data for the MLP, FCM clustering, and FS clustering as well as a comparison for the predictions from all three models. Three injury levels were taken into the consideration for this study: no injury, evident injury, disabling injury/fatality, and seven input variables were selected from the data. Table 2 shows the input and output variables for Cyprus data. The performances of the three modeling approaches (MLP, FCM clustering, and FS clustering) were obtained using MATLAB software.

Table 2: Description of the study variables for Cyprus data (Aghayan et al., 2012)

Input Variable		Coding/Values	Data
1	Driver's Gender	Man Woman	82.28% 17.72%
2	Driver's Age	Year	-
3	Crash Time	Day Night	67.17% 32.83%
4	Type of Vehicle	Passenger car Pick-up	59.76% 40.24%
5	Weather Condition	Clear Cloudy Rainy	95.19% 1.81% 3.00%
6	Trafficway Character	Curve Straight road segment	30.73% 69.27%
7	Collision Type	Rear-end Right-angle Side-wipe	12.81% 25.42% 61.77%
Output variable			
1	Driver Injury Severity	No injury=(1,0,0) Evident injury=(0,1,0) Fatality=(0,0,1)	37.84% 59.75% 2.41%

3.3 Artificial Neural Networks

Neural networks are massively parallel systems that rely on dense arrangement of interconnections and surprisingly simple processors. ANNs take their name from the networks of nerve cells in the brain. Although a great deal of biological detail is

eliminated in these computing models, the ANNs retain enough of the structure observed in the brain to provide insight into how biological neural processing may work. Thus, these models contribute to a paramount scientific challenge.

Neural networks utilize a parallel processing structure that has large numbers of processors and many interconnections between them. In a neural network each processor is linked to many of its neighbors so that there are many more interconnects than processors. The power of neural network lies in the tremendous number of interconnections. In addition, the models can be made in neutral networks to conduct useful computations as well as the capabilities of the resulting systems that provide an effective approach to previously unsolved problems. The processing power of a neural network is measured mainly with regard to the number of interconnections that update per second.

The neural network usually has three layers of processing units, a typical organization for the neural net paradigm known as back propagation. First is a layer of input units. These units assume the values of a pattern represented as a vector, which is input to the network. The middle called hidden layer is consisted of "feature detectors". The last layer is the output layer. The activities of these units are read as the output of the network.

Back-propagation is one of the networks to understand. Its learning and update procedure is intuitively appealing because it is based on the following concept. If the network gives the wrong answer, then the weights are corrected so that the error is lessened and as a result future responses of the network are more likely to be correct.

Back-propagation is a tremendous step forward compared to its processors, the perceptron. The perceptron was limited to only two layers of processing units, with only a single layer of adaptable weights. This key limitation meant that the

perceptron could only classify patterns that were linearly separable. Back-progagation overcomes this limitation since it can adapt two or more layers of weights, and uses a more sophisticated learning rule. The power of back-propagation lies in its ability to train hidden layers and thereby escape the restricted capabilities of single-layer networks.

Different neural network architectures can be made. In this study, multi-layer perceptron (MLP) neural network architecture was used which is consisted of a multi-layer feed-forward network with sigmoid hidden neurons and linear output neurons. Multi-layers of neurons with non-linear transfer function allow the network to learn non-linear and linear relationships between input and output vectors. The linear output layer lets the network produce values outside the range -1 to +1. This network with biases, a sigmoid layer, and a linear output layer are that capable of approximating any function with a finite number of discontinuities. This network can fit multi-dimensional mapping problems arbitrarily well given consistent data and enough neurons in its hidden layer. The network was trained with Levenberg-Marquardt back-propagation algorithm. This structure essentially consists of a collection of non-linear neurons organized and connected to each other in a feed-forward multi-layer structure using directed arrows as coefficients (commonly called weight and bias in neural network terminology). The structure usually consists of input nodes, a hidden layer including some neurons, and output nodes. The hidden layer is the network layer, which is not connected to the network output (for instance, the first layer of a two-layer feed forward network). This pattern is known to be well-suited to prediction and classification problems.

3.4 Genetic Algorithm

One of the best known Evolution Algorithms (EAs) is the GA developed by Holland, his student, and his colleagues at the University of Michigan. The GA is an important predecessor of the GP, from which the latter derived its name. GA has proved useful in a wide variety of real-world problems. A GA is a method for analyzing both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution.

Until recently, most efforts have been in areas other than program induction, often as methods for optimization. EAs work by defining a goal in the form of a quality criterion and then use this goal to measure and compare solution candidates in a stepwise refinement of a set of data structures. If successful, an EA returns an optimal or near optimal individual after a number of iteration. This approach is very similar to the basic principle of all evolutionary techniques. The process of selecting the best individuals for mating is simply called selection or, more accurately, mating selection. The work of De Jong (1975) demonstrated the usefulness of GAs for function optimization and was the first concerted effort to optimize GA parameters.

The two main variation operators in EAs are mutation and exchange of genetic material between individuals (Crossover). Mutation changes a small part of an individual's genome while crossover exchanges genetic material usually between two individuals, to create an offspring that is a combination of its parents. GA focuses on the crossover operator. In most applications of GA, operations are mainly either reproduction or crossover. Usually, only a small probability is used for mutations.

Figure 3 shows a genetic cycle of GA where the best individuals are continuously selected and operated on by crossover and mutation.

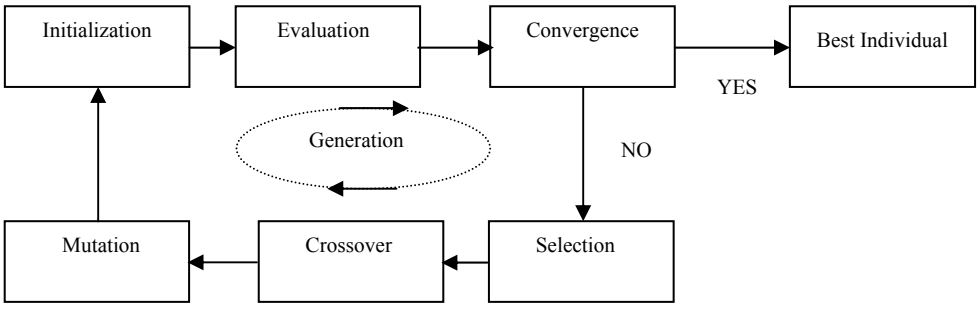

Figure 3: The general structure of GAs (Kunt et al., 2011)

3.5 Pattern Search

Direct search is a method of solving optimization problems that does not require any information about the gradient of the objective function. Unlike more traditional optimization methods that use information about the gradient or higher derivatives to search for an optimal point, a direct search algorithm searches a set of points around the current point, looking for one point where the value of the objective function is lower than the value at the current point. Direct search can be used to solve problems for which the objective function is not differentiable or is not even continuous. Pattern search (PS) algorithms are direct search methods that are capable of solving global optimization problems of highly nonlinear, multi-parameter, multimodal objective functions without the need to calculate any gradient or curvature information, especially to address problems for which the objective functions are not differentiable, stochastic, or even discontinuous (Torczon, 1997).

PS functions include two main algorithms called the generalized pattern search (GPS) algorithm and the mesh adaptive search (MADS) algorithm. Both are PS algorithms that compute a sequence of points that approach an optimal point. The PS algorithm was investigated based on GPS positive basis *2N* (Lewis and Torczon, 1999; Audet and Dennis, 2003).

At each step, the algorithm searches a set of points called a mesh around the current point that was computed in the previous step of the algorithm. The mesh is formed by adding the current point to a scalar multiple of a set of vectors called a pattern. If the PS algorithm finds a point in the mesh that improves the objective function at the current point, the new point becomes the current point in the next step of the algorithm. The MADS and GPS algorithms differ in how the mesh is computed. The GPS algorithm uses fixed direction vectors, whereas the MADS algorithm uses a random selection of vectors to define the mesh. The MADS algorithm uses the relationship between the mesh size, Δ^m and an additional parameter called the poll parameter, Δ^p to determine the stopping criteria.

For the positive bases that include *N+1* and *2N*, the poll parameter is $N\sqrt{\Delta^m}$ and $\sqrt{\Delta^m}$, respectively. The relationship for the MADS stopping criterion is $\Delta^m \leq$ mesh tolerance, where Δ^m is the mesh size.

At each iteration, the PS polls the points in the current mesh by computing the objective function at the mesh points to see if any points have function values less than the current value. The pattern that defines the mesh is specified by the poll method option. The GPS positive basis *2N* consists of the following *2N* directions, where *N* is the number of independent variables for the objective function. PSs sometimes run faster using the GPS positive basis *Np1* as the poll method rather than the GPS positive basis *2N* because the algorithm searches fewer points at each iteration (Lewis and Torczon, 2002). However, if the objective function has many local minima, using GPS Positive basis *2N* as the poll method might avoid finding a local minimum that is not the global minimum, because the search explores more points around the current point at each iteration.

3.6 Types of Fuzzy Inference Systems

3.6.1 Takagi-Sugeno-type fuzzy model

The fuzzy model methodology suggested by Takagi-Sugeno (TSK) in 1985 has been applied in theoretical analysis, control applications, and fuzzy modeling. A typical fuzzy rule for an *n*-input, single-output TSK fuzzy model has the form:

$$R_i : if\ x_1 is\ A_1^i and\ x_2 is\ A_2^i\ and\ x_n is\ A_n^i\ then\ z_i = f_i(x_1, x_2, ..., x_n)\ for\ i = 1, 2, ..., k \qquad \text{(Eq. 1)}$$

Where k is the number of fuzzy if-then rules, A_n^i is the membership function (MF), and $\mu_{A_n^i}(x_n)$ is the membership degree of n^{th} input x_n for i^{th} rule. The consequent part of the rule base represents the output of the rule. In the TSK fuzzy inference system, the output is a crisp function instead of a fuzzy membership function. The output function can be stated as

$$z_i = f_i(x_1, x_2, ..., x_n) = (a_1^i x_1 + a_2^i x_2 + ... + a_n^i x_n + c^i) \qquad \text{(Eq. 2)}$$

Where $a_1, a_2, ..., a_n, c$ are constants.

The degree of matching between the inputs and rule R_i is defined as the rule firing strength, β, and can be calculated by the minimum operator as follows:

$$\beta_i = \mu_{A_1^i}(x_1) \wedge \mu_{A_2^i}(x_2) ... \wedge \mu_{A_n^i}(x_n) \qquad \text{(Eq. 3)}$$

The overall fuzzy system output is the weighted average of all rule outputs determined as:

$$\text{Final output} = \frac{\sum_{i=1}^{k} \beta_i z_i}{\sum_{i=1}^{k} \beta_i} \qquad \text{(Eq. 4)}$$

The great advantage of the TSK model is its descriptive ability, and it is capable of describing a highly non-linear system using a small number of rules.

3.6.2 Mamdani-type fuzzy model

Initially, the Mamdani FIS (Mamdani and Assilian, 1975) was the most widely used in fuzzy systems and fuzzy control for which the implications were that both the input and output of the if-then rules consisted only of fuzzy sets. In contrast, the TSK is related to rules according to a special format, one characterized by functional-type consequents instead of the fuzzy consequents used by Mamdani.

3.7 Cluster Validity

One of the cluster validity techniques is to use the silhouette value in order to evaluate the quality of a clustering allocation, independently of the clustering technique that is used. Thus, silhouette values were used defined as the similarity of each point to points in its own cluster with points which belongs to other clusters. The Silhouette coefficient varies from +1 to -1. If it is close to zero this indicates that the points are not distinct to any given cluster and when it is close to one this means the points are assigned to a very appropriate cluster and finally when it is near to -1, this represent of misclassifying and the point is merely somewhere in between the clusters (Shie and Chen 2008). For this aim, the silhouette coefficient is calculated by the Equation 5.

$$S_i = \frac{min(b_i^k) - a_i}{max(a_i, min(b_i^k))} \quad \text{(Eq. 5)}$$

Where a_i is the average distance from the i^{th} point to the other points in its cluster, b_i^k is the average distance from the i^{th} point to points in another cluster k. In order to find the optimum value of the k, it should be increased until the new mean silhouette coefficient becomes less than the previous one. At that point, adding further clusters begins to reduce accuracy in the clustering, and the addition process must stop. In this way, the mean silhouette coefficient between 0.7 and 1.0 indicates a strong

structure, between 0.5 and 0.7 means a reasonable structure, between 0.25 and 0.5 points out a weak structure and less than 0.25 indicates an insubstantial structure (Kononenko & Kukar 2007).

3.8 Hierarchical Clustering

Clustering is fundamentally a collection of methods of data exploration. Hierarchical clustering procedures use the method of summarizing data structure. Hierarchical clustering can be categorized as an agglomerative or divisive algorithm (Jain at el. 1999, Jiang at el. 2004). The agglomerative hierarchical algorithm is used as an explanatory statistical technique to determine the number of clusters of datasets (Sneath and Sokal, 1973; King, 1967; Guha et al., 1995, 1998; Karypis et al., 1999).

In this study, the agglomerative hierarchical algorithm was employed. The agglomerative algorithm is initiated by assuming that each of *n* objects to be clustered is a unique cluster. The objects were compared, with each other using a Euclidean distance to determine the distance between objects. That process was repeated until the number of clusters was obtained. The average linkage method defined in Equation 6 was applied for comparing the clusters in each stage between all pairs of objects and deciding which of them should be combined.

$$d(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} \left\| x_{ri} - x_{sj} \right\| \quad \text{(Eq. 6)}$$

Here, x_{ri} is the i^{th} object in cluster *r* and n_r is the number of objects in cluster *r*. This methodology partitions data by identifying natural groupings in the hierarchical tree or by cutting off the hierarchical tree at a random point.

A graphical presentation called dendrogram is available showing the clustering results of each stage with regard to 800 training data which belongs to Cyprus data presented in Figure 4. Figure 5 shows a simplified dendrogram in which no more that

12 leaf nodes were formed by collapsing the lower branches of the tree. By this means, cluster 1= [6 9 4 8], cluster 2= [1 3 7 2 5], and cluster 3= [10 12 11]; thus, the number of members comprising the nodes was equal to 800, which was obtained from the summation of 223, 540, and 37.

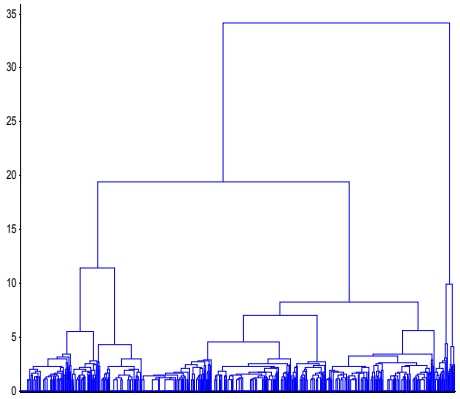

Figure 4: Hierarchical clustering dendrogram with Cyprus data (Aghayan et al., 2013)

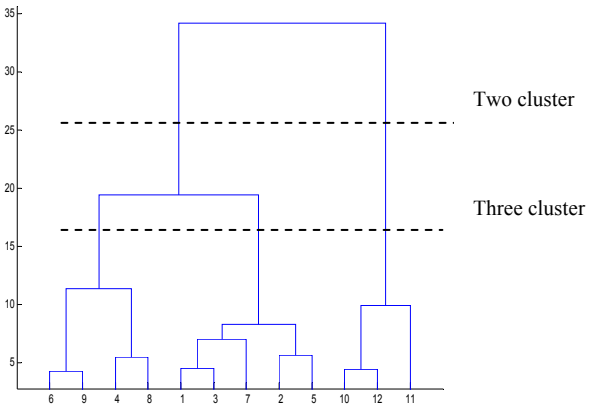

Figure 5: Simplified dendrogram for hierarchical clustering with Cyprus data (Aghayan et al., 2013)

3.8.1 Verifying the Cluster Tree

After the objects in a dataset are linked into a hierarchical cluster tree, the distances in the tree are verified as to whether they accurately reflect the original distances.

In a hierarchical cluster tree, two objects are linked to each other at some level in original data. The distance between two clusters is represented with the height of the link that includes two objects. The height is considered as the cophenetic distance between the two objects. The cophenetic distance is compared to the original distance data in order to find the behavior of generated cluster tree. If the clustering is valid, the linking of objects in the cluster tree should have a strong correlation with the distances between objects in the distance vector. The cophenetic function compares these two sets of values and computes their correlation, returning a value called the cophenetic correlation coefficient (CPCC). The CPCC for a cluster tree is defined as the linear correlation coefficient between the cophenetic distances obtained from the tree and the original distances (or dissimilarities), which varies between 0 and +1. The CPCC value is close to 1 for a high-quality solution. The CPCC between Z, the average linkage method, and Y, the Euclidean distance for all data, is defined by Equation 7:

$$C = \left| \frac{\sum_{i<j}(Y_{ij}-y)(Z_{ij}-z)}{\sqrt{\sum_{i<j}(Y_{ij}-y)^2 \sum_{i<j}(Z_{ij}-z)^2}} \right| \quad \text{(Eq. 7)}$$

Where Y_{ij} is the distance (Y_i , Y_j), Z_{ij} is the cophenetic distance between objects i and j in Z as well as y and z are the average of Y and Z, respectively.

In this study, the CPCC was obtained from the preliminary data related to Cyprus data was 0.842, which indicated that the hierarchical cluster tree was fairly good in terms of accuracy of the clustering solution.

The clustering efficiency was checked by a silhouette plot method. The overall separation parameter was determined by average silhouette values. For example, Figures 6 and 7 show the results of using silhouette values for Cyprus dataset. As shown, traffic crash data was divided into two and three clusters. The mean

silhouette values for the two and three clusters were found to be 0.707 and 0.796, respectively. Also, Figure 6 depicts a few points with negative values which mean that the separation into two clusters was not justified in comparison with the separation into three clusters shown in Figure 7.

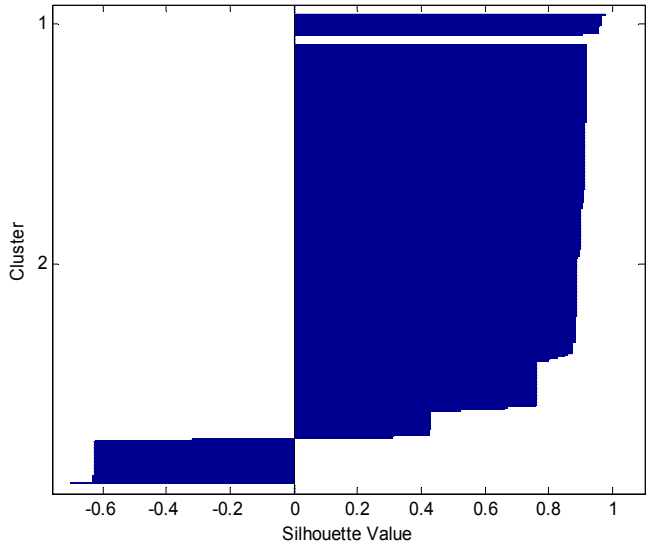

Figure 6: Silhouette values for two clusters in the hierarchical clustering with Cyprus data (Aghayan et al., 2013)

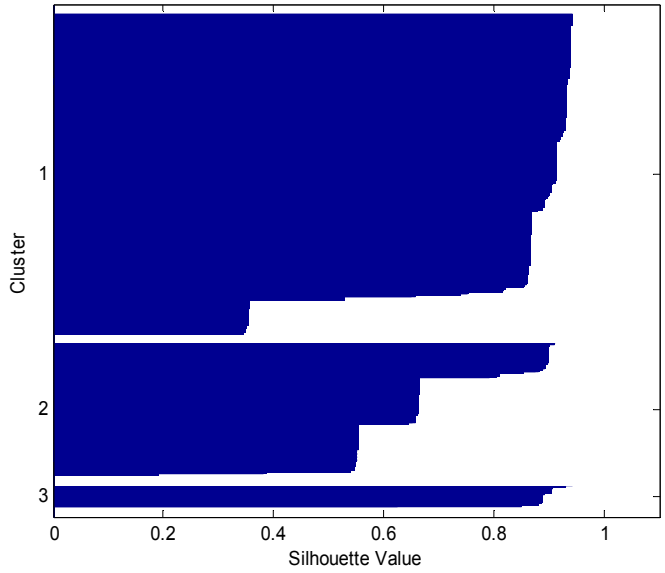

Figure 7: Silhouette values for three clusters in the hierarchical clustering with Cyprus data (Aghayan et al., 2013)

3.9 K-means clustering

The K-means clustering algorithm can be applied as an iterative optimization procedure. Generally, the K-means clustering algorithm begins the clustering process by using a randomly selected set of initial centroid locations. Just as in many other types of numerical minimizations, the solution that K-means reaches sometimes depends on the starting point. It is possible for the algorithm to reach a local minimum, where reassigning any one point to a new cluster would increase the total sum of distances, but where a better solution does exist. However, the parameter is replicated to overcome that problem. When more than one replicate is specified, the K-means algorithm repeats the clustering process starting from different randomly selected centroids for each replication.

K-means uses a two-phase iterative algorithm (batch and online updates) to minimize the sum of point-to-centroid distances, summed over all K clusters. In this study, a modified K-means methodology was employed to reach the local minimum in any circumstance, which was useful for the large number of records. The modified K-means method included batch and online updates in which the first phase entailed reassigning the points to the closest cluster centroid through recalculation of cluster centroids and the second step entailed determining a clustering solution by convergence to a local minimum where points were individually reallocated and cluster centers were recalculated after each reallocation. However, partitioning X into K exhaustive and mutually exclusive clusters $S = \{S_1, S_2, \cdots, S_k\}$, $\cup_{k=1}^{K} S_k = X$, $S_i \cap S_j = \emptyset$ for $1 \leq i \neq j \leq K$ performed by minimizing of the squared-error for the Equation 8 as used as objective function.

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in s_k} \|x_i - c_k\|^2 \qquad \text{(Eq. 8)}$$

Where $X = \{x_1;\ x_2;\ \cdots;\ x_N\} \in \mathbb{R}^{N\times D}$ represents a vector of real numbers, N is the number of data, $C = \{c_1;\ c_2, \cdots; c_K\} \in \mathbb{R}^{K\times D}$ is the corresponding set of centers, K is the number of clusters, and $\|x_i - c_k\|$ is the Euclidean distance between x_i and c_k.

The pseudocode for K-means clustering is given in Algorithm 1.

Algorithm 1: Modified K-means clustering algorithm (Aghayan et al., 2012)

Clustering variables

X: An objects; S_i: The i^{th} cluster; c_i: The centroid of cluster S_i; C: The centroid of all points; N: The number of object in the dataset; K: The number of clusters.

---

input: $X = \{x_1; x_2; \ldots; x_N\} \in \mathbb{R}^{N\times D}$ ($N\times D$ input data set)
output: $C = \{c_1; c_2; \ldots; c_K\} \in \mathbb{R}^{K\times D}$ (K cluster centers)
%replicates: Number of times to repeat the clustering, with a new set of initial cluster centroid
for (replicates =1:1:rep);
Choose a random subset C of X as the initial set of cluster centers;
while termination criterion is not met: $\{minimize : J(C)\}$
for *(j=1:1:N);*
Assign x_j to the nearest cluster;
for (i=*1:1:K);*
$\mathbf{S_i^t} = \{\mathbf{x_j} : \text{Min}(\|\mathbf{x_j} - \mathbf{c_i^t}\|, \|\mathbf{x_j} - \mathbf{c_{i*}^t}\|$ for i*∈ [1: K]-[i])}
end
end
Recalculate the cluster centers;
for *(k=1:1:K)*
Cluster $\mathbf{S_k^t}$ includes the set of point's x_i that are nearest to the center c_k^{t+1} ; $|\mathbf{S_k^t}| = \{\mathbf{x_i} \mid \mathbf{S_k^t}\}$; the number of data in cluster *i;*
Calculate the new center c_k as the mean of the points that belong to $\mathbf{S_k^t}$; $c_k^{t+1} = \frac{1}{|S_k^t|}\sum\{x_i \in S_k^t\}$
end
end
end
Best replicates: min {total sum of distances: *[1: rep]*}

Silhouette plots are shown in Figures 8 and 9 for two clusters and three clusters with Cyprus data, respectively. As shown, the most points in both clusters had large silhouette values, greater than 0.8, showing that those points were well separated from neighboring clusters. For the two and three clusters, the mean silhouette value

was found to be 0.807 and 0.788, respectively. The number of clusters was increased to find out if K-means could find further grouping structures in the data.

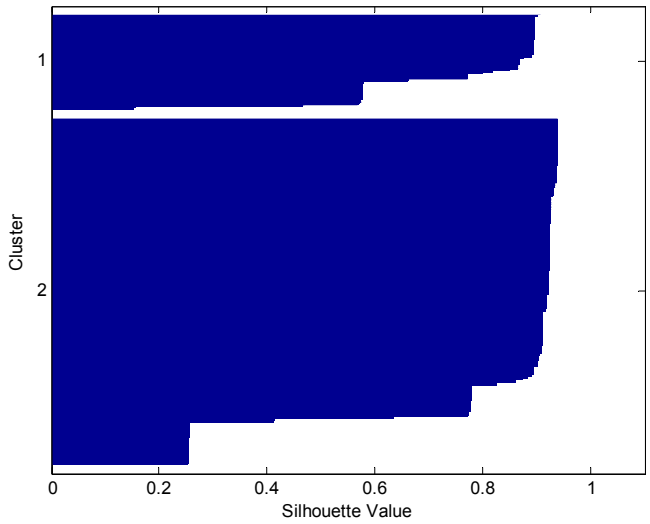

Figure 8: Silhouette values for two clusters in the K-means with Cyprus data (Aghayan et al., 2013)

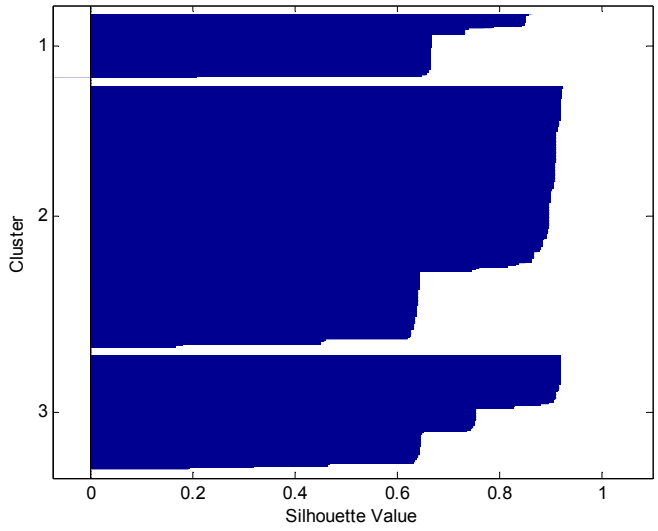

Figure 9: Silhouette values for three clusters in the K-means with Cyprus data (Aghayan et al., 2013)

3.10 Fuzzy C-means clustering

Similar to fuzzy rules, fuzzy clusters, are well suited as a means for building a classification model. Clusters are often considered as fuzzy rules to initialize a fuzzy rule system that is then optimized. The essential procedure of FCM is to find clusters such that the overall distance from a cluster prototype to each datum is minimized. The FCM algorithm is defined by the objective function:

$$J_{FCM}(U,V;X) = \sum_{k=1}^{n}\sum_{i=1}^{c} U_{ik}^{m}\,\|x_k - v_i\|^2 \qquad \text{(Eq. 9)}$$

Where $d_{ik}^2 = \|x_k - v_i\|^2$, and $\|x_k - v_i\|$ is the Euclidean distance between the centroids that characterizes the k^{th} data point and i^{th} cluster. Moreover, n is the number of data points, c is the number of cluster, x_k is the k^{th} data point, v_i is the i^{th} cluster center, m is weighting exponent on each fuzzy membership function, and U_{ik} is the degree of membership of the k^{th} data point in the i^{th} cluster. The parameter m controls the fuzziness of the resulting partition varying in the range [1, ∞). The cluster center v_i and the degree of MF, U_{ik} that are used in $J_{FCM}(U,V;X)$ are defined by:

$$U_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{\|x_k - v_i\|}{\|x_k - v_j\|}\right)^{\frac{2}{(m-1)}}} \qquad \text{(Eq. 10)}$$

$$v_i = \frac{\sum_{k=1}^{n} U_{ik}^{m} x_k}{\sum_{k=1}^{n} U_{ik}^{m}} \qquad \text{(Eq. 11)}$$

In this study, a modified FCM clustering was employed. For this procedure, the initial FCM partition was defined and set with the number of clusters equal to 3, the exponent for the partition matrix equal to 2, the maximum number of iterations equal to100 and minimum improvement equal to 1e-10. Based on this, the initial fuzzy

cluster centers were calculated through the generation of the initial fuzzy partition. To improve the FCM clustering, the cluster centers and the membership grade points were updated, and the objective function defined in Equation 9 was minimized to find the best location for each cluster. This procedure was terminated when the maximum number of iterations or minimum amount of improvement were reached.

Figures 10, 11, 12 and 13 show the results of silhouette values and the objective function values for two and three clusters with Cyprus data, respectively. After 26 iterations for two clusters, the objective function and the mean silhouette value were equal to 28645.730 and 0.799, respectively. After 39 iterations for three clusters, the objective function and the mean silhouette value were equal to 13531.845 and 0.788, respectively.

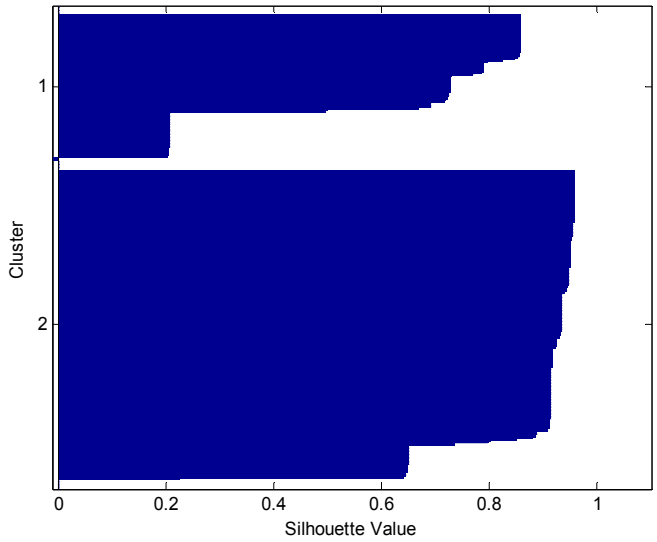

Figure 10: Silhouette values for two clusters in the FCM with Cyprus data (Aghayan et al., 2013)

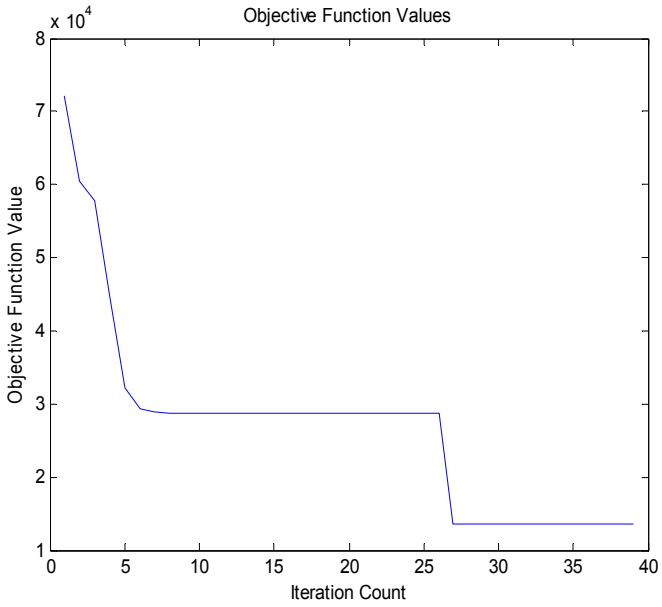

Figure 11: The objective function values at each iteration in the FCM for Cyprus data (Two clusters)

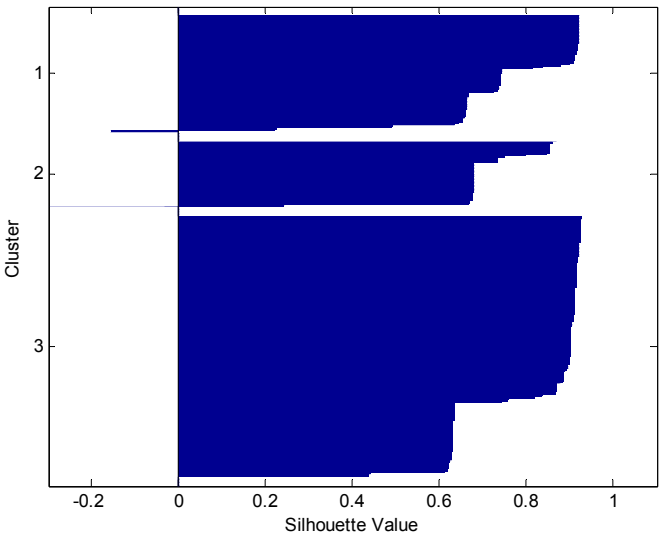

Figure 12: Silhouette values for three clusters in the FCM with Cyprus data (Aghayan et al., 2013)

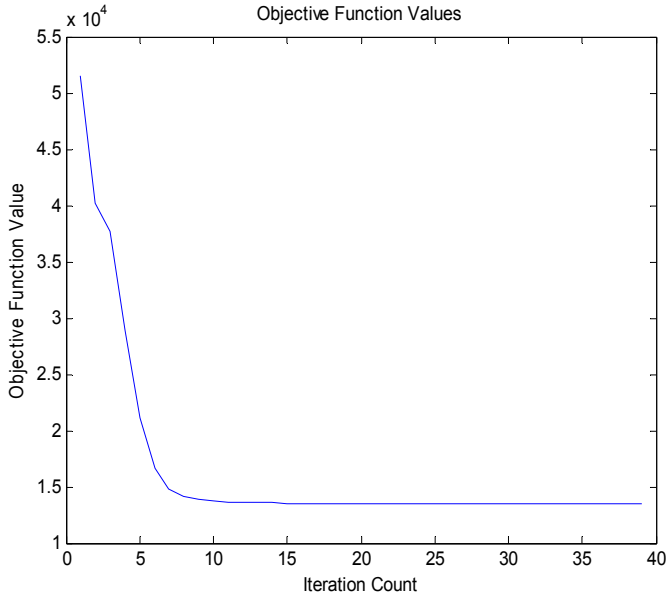

Figure 13: The objective function values at each iteration in the FCM for Cyprus data (Three clusters)

Figure 14 depicts the relationship between iterration count and objective function value in FCM clustering with regard to the number of clusters.

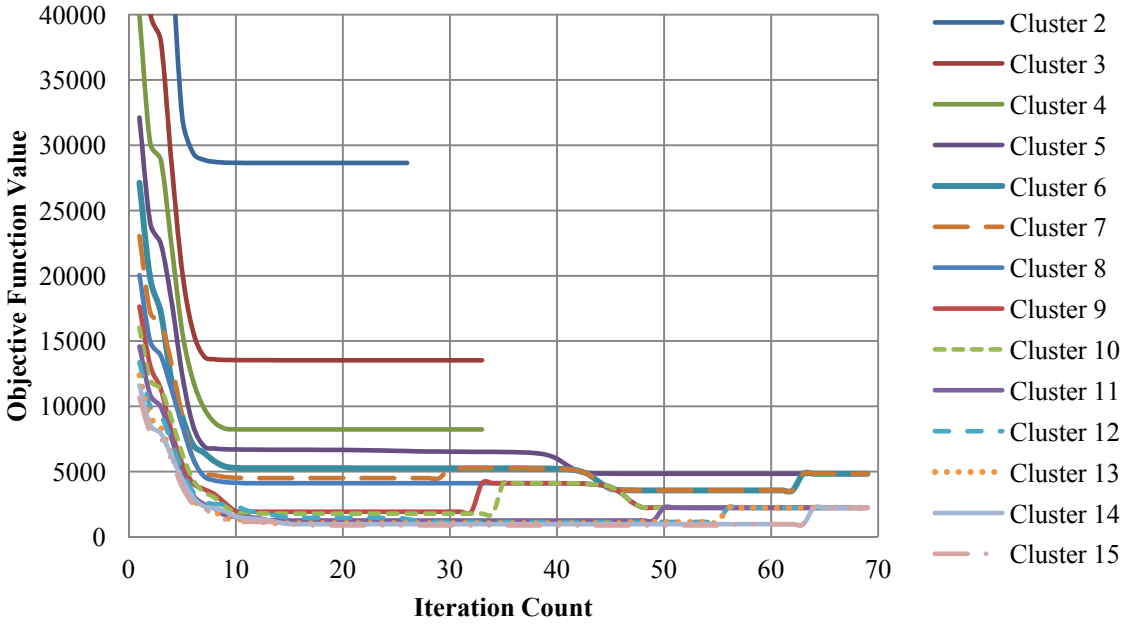

Figure 14: The relationship between iteration count and objective function value in FCM for Cyprus data (Aghayan et al., 2013)

3.11 Fuzzy subtractive clustering

Subtractive clustering uses data points as the candidates for cluster centers instead of grid points, which means that the computation is related to the problem size (Hammouda and Fakhreddine, 2002). In fact, the cluster centers should be located at the data points to reduce the computation effort. Thus, each data point is a candidate for cluster centers; a potential measure at any point x_i is defined as:

$$P_i = \sum_{j=1}^{N} \exp\left(-\frac{\|x_i - x_j\|^2}{\left(\frac{r_a}{2}\right)^2}\right) \quad \text{(Eq. 12)}$$

Where x_i is the i^{th} data point, N is the total number of data points, and r_a is a positive constant representing a neighborhood radius. Hence, a data point will have a high potential value if it has many neighboring data points. The first cluster center x_1^* is chosen as the point having the largest potential value p_1^*. In order to locate the next cluster center, the influence of the previous identified cluster center and the data points near to the center are reduced by revising the potential measure. This procedure is conducted by subtraction, as shown in Equations 13 and 14.

$$P_k = P_{k-1} - P_{k-1}^* \, exp\left(-\frac{\|x_{k-1}^* - x_i\|^2}{\left(\frac{r_b}{2}\right)^2}\right) \quad \text{(Eq. 13)}$$

$$r_b = \rho * r_a \quad \text{(Eq. 14)}$$

Where ρ is a positive constant greater than 1 and is called squash factor. In addition, r_b is a positive constant which defines a neighborhood. That neighborhood, moreover, has measurable reductions in the potential measure and is somewhat greater than r_a, which helps in avoiding closely spaced cluster centers. Thus, the data points near to the first cluster center x_1^* will have a significantly

reduced potential measure. After revising the potential function, the next cluster center is chosen as the point having the greatest potential value. This process is stopped when a sufficient number of clusters are achieved.

The process of acquiring new cluster centers is based on potential values related to an acceptance threshold ε^{upper}, a rejection threshold ε^{lower}, and the relative distance criterion. A data point with the potential greater than the acceptance threshold is accepted directly as a cluster center. The relative distance equation is defined as:

$$\frac{d_{min}}{r_a} + \frac{p_k^*}{p_1^*} \geq 1 \qquad \text{(Eq. 15)}$$

Where $d_{\min}$ is the shortest distance between the candidate cluster center and all previously found clusters' centers. The pseudocode for subtractive clustering and FS clustering are given in Algorithm 2 and 3, respectively.

Algorithm 2: Subtractive clustering algorithm (Aghayan et al., 2012)

Clustering variables

X: An objects *(N×D);* x_i^**:* Location of i^{th} cluster; P_i^**:* Potential of i^{th} cluster; *C:* The number of clusters; r_a *&* r_b*:* Positive constant*;* ε^{upper}: Accept ratio*;* ε^{lower}: Reject ratio

---

input: $X = \{x_1; x_2; \ldots; x_N\} \in \mathbb{R}^{N\times D}$ ($N\times D$ input data set)
output: $X^* = \{x_1^*; x_2^*; \ldots; x_C^*\} \in \mathbb{R}^{C\times D}$ (C cluster centers)
for (i =1:1:N);
$P_1(x_i) = \sum_{j=1}^{N} exp\left(-\alpha\left\|x_i - x_j\right\|^2\right)$ & $\alpha = \frac{4}{r_a^2}$; %initial potential for each data point
end
$P_1^* = argmax_{(i=1,2,\ldots,N)}(P_1(x_i))$ %potential value for the first cluster center.
$P_1^* = P_1(x_1^*)$ % location of the first cluster center
while ($k \leq C$); k=2;
  $P_k(x_i) = P_{k-1}(x_i) - P_{k-1}^* exp(-\beta\|x_{k-1}^* - x_i\|^2)$ & $\beta = \frac{4}{r_b^2}$ % next cluster center
  if $\varepsilon^{\text{lower}}\, P_1^* < P_k^* < \varepsilon^{\text{upper}}\, P_1^*$
    $d_{min} = min(\|x_k^* - x_{1:k-1}^*\|)$
    if $d_{min}/r_a + P_k^*/P_1^* \geq 1$
      $P_k^* = P_k(x_k^*)$; % location of the next cluster center

continue; k=k+1;%go to the beginning of the loop
else
$P_{k-1}(x_i \mid P^*_{k-1} = P_{k-1}(x_i)) = 0$; % eliminating rejected value by assigning potential value 0.
$P^*_{k-1} = argmax_{(i=1,2,\ldots,N)}(P_{k-1}(x_i))$; % choose next higher potential value
$P^*_{k-1} = P_{k-1}(x^*_{k-1})$; return;
end
end
if $P^*_k > \varepsilon^{upper} P^*_1$
$P^*_k = P_k(x^*_k)$ % location of the next cluster center
continue; k=k+1;%go to the beginning of the loop
elseif $P^*_k < \varepsilon^{lower} P^*_1$
Break; %the algorithm is finished.
end
end

Algorithm 3: Fuzzy subtractive clustering-Rule i^{th} (Aghayan et al., 2012)

Clustering variables

X: An objects *(N×D); X*:* location of data cluster; *Y:* Input*; Y*:* location of input cluster; *Z:* Output*; Z*:* location of output cluster; A^i_j: Gaussian Membership function; B^i_j : Singleton Membership function

$X = \{Y, Z\} = \{x_1; x_2; \ldots; x_N\}$; $Y = \{y_1; y_2; \ldots; y_7\}$; $Z = \{z_1; z_2; z_3\}$
$X^* = \{Y^*, Z^*\} = \{x^*_1; x^*_2; \ldots; x^*_N\}$; $Y = \{y^*_1; y^*_2; \ldots; y^*_7\}$; $Z = \{z^*_1; z^*_2; z^*_3\}$
Rule$_i$: $if\ \{y_1\ is\ A^i_1\ and\ y_2\ is\ A^i_2, \ldots, y_7\ is\ A^i_7\}$ Then $\{\ z_1 is\ B^i_1, z_2 is\ B^i_2, z_3\ is\ B^i_3\}$,
Where:

$$A^i_j(y_j) = exp\left[\frac{-1}{2}\left(\frac{(y_j - y^*_{i,j})}{\sigma^i_j}\right)^2\right] \quad \& \quad B^i_j(z_j) = \{1,\ if:\ y_j = y^*_{i,j}\ or\ 0,\ if:\ y_j \neq y^*_{i,j}\},$$

Therefore:

$$\hat{Z} = \frac{\sum_{i=1}^{c}(\mu_i z^*_i)}{\sum_{i=1}^{c}(\mu_i)} = \frac{\sum_{i=1}^{c}(\prod_{j=1}^{7}(A^i_j(y_j))z^*_i)}{\sum_{i=1}^{c}(\prod_{j=1}^{7}(A^i_j(y_j)))}$$; %Output vector

3.12 Regression Model Goodness-of-Fit Measures

Goodness-of-fit (GOF) statistics are useful for comparing results across multiple studies, for comparing competing models within a single study, and for providing feedback on the extent of knowledge about the uncertainty involved with the phenomenon of interest. Four measures of model GOF are discussed: the sum of

squared error (SSE), root mean square error (RMSE), the correlation coefficient (R), mean absolute error (MAE).

3.12.1 Sum of Squares Due to Error

This statistic measures the discrepancy between the data and an estimation model. It is also called the sum of squared residuals (SSR) or is usually labeled as SSE of prediction by Equation 16 in which y_i is response value (target output) and $\hat{y}_i$ is prediction response value:

$$SSE = \sum_{i=1}^{n} \left(\hat{y}_i - y_i \right)^2 \quad \text{(Eq. 16)}$$

An SSE value closer to 0 indicates that the model has a smaller random error component.

3.12.2 Root Mean Squared Error

This statistic is also known as the fit standard error and the standard error of the regression. RMSE is used as the measure of the differences between values predicted by a model or an estimator and the observed values defined as Equation 17:

$$RMSE = S = \sqrt{MSE} \quad \text{(Eq. 17)}$$

Where *MSE* is the mean squared error, Equation 18:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(\hat{y}_i - y_i \right)^2 \quad \text{(Eq. 18)}$$

Where, *n* denotes the size of predicting sample, y_i and $\hat{y}_i$ represent the measured and estimated values, respectively. An MSE value closer to 0 indicates a fit that is more useful for prediction.

3.12.3 Mean Absolute Error (MAE)

The average error of an estimator $\hat{y}_i$ with respect to the estimated parameter y_i is defined as the mean of the absolute difference between the estimator and the real value, Equation 19:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \quad \text{(Eq. 19)}$$

3.12.4 Correlation coefficient (R)

Correlation is a criterion used for measuring if an attribute is relevant to others in dataset and the relevant to classes. The strength and the direction of a linear relationship between two variables can be measured by correlation coefficient (R) defined by Equation 20:

$$R(i,j) = \frac{C(i,j)}{\sqrt{C(i,i)C(j,j)}} \quad \text{(Eq. 20)}$$

Where R value is the correlation coefficient between variables i and j, $C(i,j)$ is the covariance matrix defined by Equation 21:

$$COV = \frac{\sum_{i=1}^{n}(X_i - \bar{x})(Y_i - \bar{y})}{n-1} \quad \text{(Eq. 21)}$$

The correlation coefficients range from -1 to 1, where values close to 1 suggest that there is a positive linear relationship between the data columns, values close to -1 suggest that one column of data has a negative linear relationship to another column of data (*anticorrelation*), and values close to or equal to 0 suggest that no linear relationship exists between the data columns.

3.12.5 Data Normalization

Some data for each variable should be normalized due to dissimilar units and magnitudes. Data normalization leads to improving the data fitting and prediction accuracy. The normalization can be conducted by using the following formula:

$$X_n = \frac{(X - X_{\min})}{(X_{\max} - X_{\min})} \quad \text{(Eq. 22)}$$

Where X_n is the normalized value within [0, 1], X is the original value, $X_{\min}$ and $X_{\max}$ are an instance of the minimum and the maximum values of the vector to be normalized.

3.13 Models Used For Analysis with Iranian Data

3.13.1 Multilayer Perceptron Neural Networks

The MLP model consisted of two layers that each layer had a weight matrix *W*, a bias vector *b*, and an output vector p^i that $i > 1$. Figure 15 shows the selected final model for each of these layers in the MLP model. The number of the layer was appended as a superscript to the variable of interest. Superscripts were used to identify the source (second index) and the destination (first index) for the various weights and other elements of the network.

The weight matrix connected to the input vector p^1 was labeled as an input weight matrix ($IW^{1,1}$) having a source 1 (second index) and a destination 1 (first index). Elements of layer 1, such as its bias, net input *(n),* and output have a superscript 1 to represent that they were associated with the first layer.

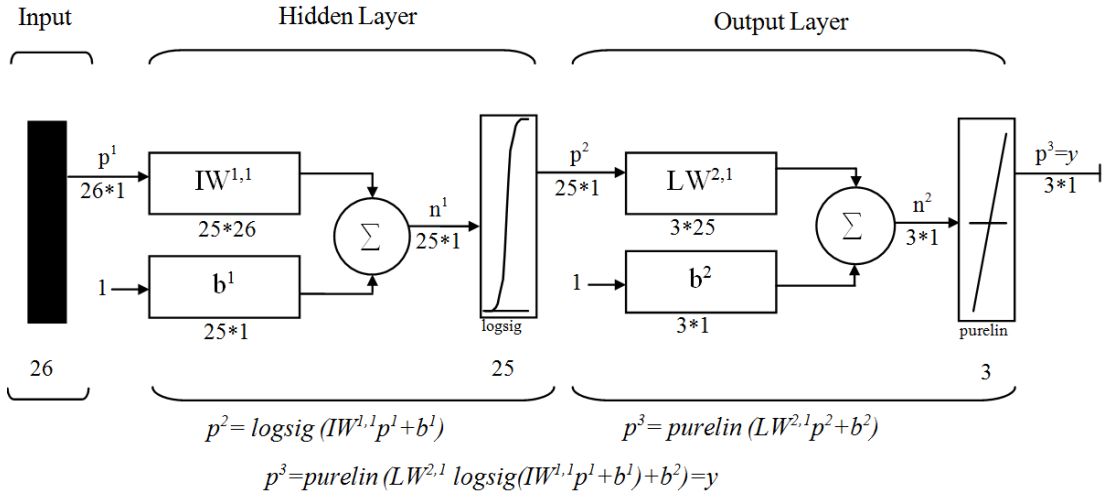

Figure 15: The structure of the final MLP model for Iranian data (Kunt et al., 2011)

Layer weight *(LW)* matrices and input weight *(IW)* matrices were used in the MLP model. The data were randomly divided into three parts: training, testing, and validating. The MLP model had 12 inputs, 25 neurons in the first layer, 3 neurons in the second layer. The output layer of the MLP model consisted of three neurons representing the three levels of injury severity. Seventy percent of the original data were used in the training phase, and the validation and test data sets each contained 15% of the original data.

A constant input 1 was fed to the bias for each neuron with regard to the outputs of each intermediate layer that were the inputs to the following layer. Therefore, layer 2 can be analyzed as a one-layer network with 25 inputs, 3 neurons, and a 3*25 weight matrix W^2; in such circumstances, the input layer 2 is p^2. All vectors and matrices of layer 2 have been determined and it can be considered as a single-layer network on its own. The layers of a multi-layer network play different roles in prediction process. This type of two-layer network was applied in back-propagation algorithm. In this study, it was assumed that the output of the second layer, p^3, was the network output of interest, and this output was labeled as y.

However, the objective of this network is to reduce the error *e* through the least mean square error (LMS) algorithm. The perceptron learning rule calculates the desired changes (target output) to the perceptron's weights and biases, given an input vector p^1 and the associated error *e*. It causes the average of the sum of those errors to be minimized.

The error at the output neuron *j* at iteration *t* can be calculated by the difference between the desired output (target output) and the corresponding real output, $e_j(t) = d_j(t) - y_j(t)$. Accordingly, Equation 23 is the total error energy of all output neurons.

$$\varepsilon(t) = \frac{1}{2}\sum_{J\in c} e_j^2(t) \qquad \text{(Eq. 23)}$$

Referring to Figure 15, the output of the j^{th} neuron in the l^{th} layer can be calculated by Equation 24 in which the transfer function is defined as $f_2 = \log sig$ and $f_3 = purelin$. Log-sigmoid transfer function ($\log sig$) is used in multilayer networks and the linear transfer function ($purelin$) is used in back-propagation networks.

$$y_j^l = f_l\left(\sum_{i=1}^{n^{l-1}} w_{ij}^l . y_i^{l-1}\right) \qquad \text{(Eq. 24)}$$

Where $1 \le l \le 3$, n^l refers to the number of neurons in the layer 1. For the input layer thus holds $l = 1$, $y_j^1 = x_j$ and for the output layer $l = 3$, $y_j^3 = y_j$. The MSE of the output can be computed by:

$$E = \frac{1}{2}\sum_{j=1}^{3}(d_j - y_j)^2 = \frac{1}{2}\sum_{j=1}^{3}\left[d_j - f_3\left(\sum_{i=1}^{25} w_{ij}^3 . y_i^2\right)\right]^2 \qquad \text{(Eq. 25)}$$

The steepest descent of the MSE can be used to update the weights by Equation 26 (Yeung et al. 2010):

$$w_{ij}^{3}(t+1) = w_{ij}^{3}(t) - \eta \frac{\partial E}{\partial w_{ij}^{3}} \quad \text{(Eq. 26)}$$

The mean square error performance index for the linear network is a quadratic function as shown in Equation 25. Thus, the performance index will either have one global minimum, a weak minimum, or no minimum, depending on the characteristics of the input vectors. Specifically, the characteristics of the input vectors determine whether or not a unique solution exists (Hagan et al. 1996).

The results of the MLP model are represented in Table 3 in the form of prediction table for 20 runs. Table 3 depicts the prediction level of injury severity patterns in training, test, validation phases.

Table 3: Prediction table for MLP model with Iranian data (Kunt et al., 2011)

R	No Injury	Evident Injury	Fatality	Overall
Training	0.9091	0.9029	0.8966	0.9125
Validation	0.8187	0.7613	0.6974	0.7863
Test	0.8372	0.6936	0.7587	0.7737
All	0.8849	0.8513	0.8372	0.8731

Figure 16 shows regression plots for the output according to training, validation, and test data. The value of the correlation coefficient (R) for each phase was calculated. The R-value was around 0.87 for the total response in the MLP model.

Figure 17 plots the training, validation, and test errors to find the validation error in the training window. The best validation performance occurred at iteration 7, and the network at this iteration was returned. The plot in Figure 17 shows the mean squared error of the network starting at a large value and decreasing to a smaller value, which means that the network learning was improving. The plot has three lines, because the 1000 input and target vectors were randomly divided into three sets. 70% of the vectors were being used to train the network. 15% of the vectors were used to validate how well the network was generalized. Training on the training

vectors continued as long as the training reduced the network's error on the validation vectors. After the network memorized the training set (at the expense of generalizing more poorly), training is stopped. This technique automatically avoided the problem of over fitting, which plagued many optimization and learning algorithms. Finally, the last 15% of the vectors provided an independent test of network generalization to data that the network has never seen. Figure 18 shows the time response of MLP with regard to number of runs which was around 7.627 seconds.

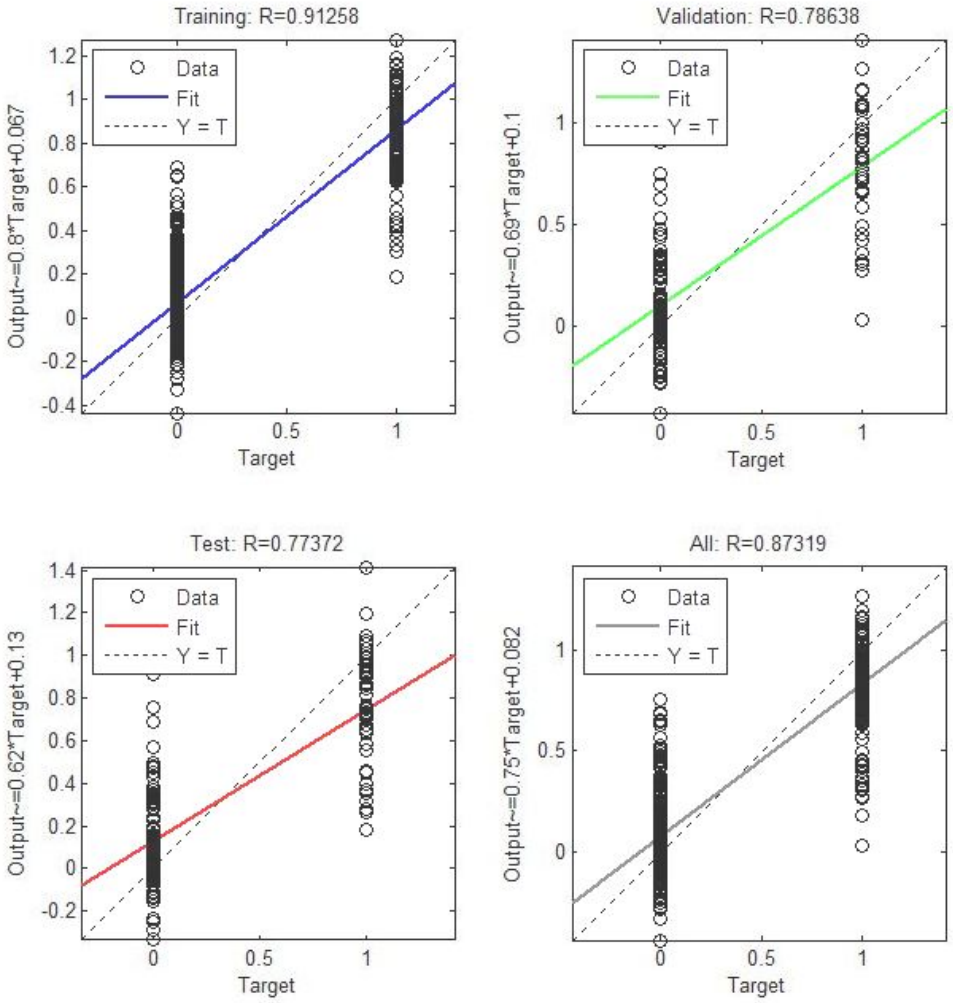

Figure 16: The regression plots for training, test, validation phases and total response in the MLP model for Iranian data (Kunt et al., 2011)

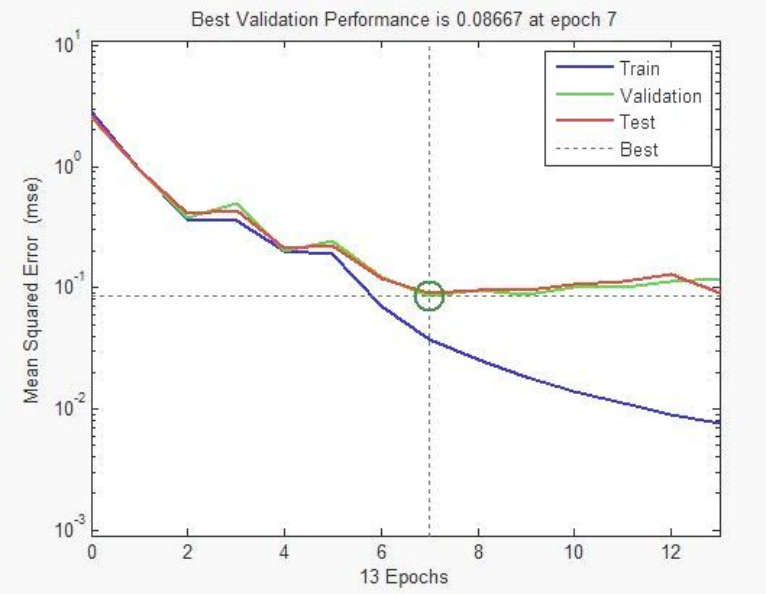

Figure 17: The validation error in the MLP model for Iranian data (Kunt et al., 2011)

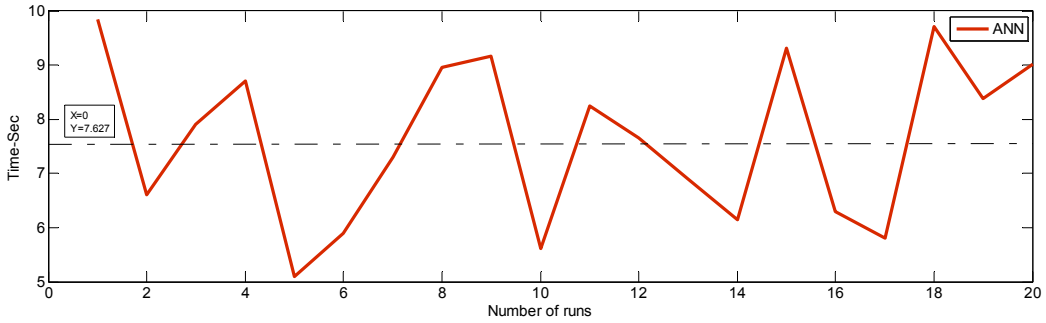

Figure 18: The time response of MLP with regard to number of runs for Iranian data

3.13.2 Genetic Algorithm

The GA is an optimization and search technique based on the principles of genetics and natural selection. The GA starts with a population of solutions (chromosomes) represented by coded strings (typically 0 and 1 binary bits) as the underlying parameter set of the optimization problem. GAs generate successively improved populations of solutions (better generations) by applying three main genetic operators: selection, crossover, and mutation. The selection function chooses parents for the next generation based on their scaled values from the fitness scaling

function, where the stochastic uniform selection function was used in this study. Crossover is achieved by exchanging coding bits between two mated strings. The chromosomal material of different parents can be combined to produce an individual that could get benefit from the strength of both parents. In this case, the applied crossover function used was scattered.

Mutation occasionally provides and recovers useful material for chromosomes through random alteration of the value of a string bit (in the binary case, from 0 to 1 and vice versa). In our case, Gaussian mutation function was used. The following formula was obtained from 1000 police records, and the system was able to modify the formula based on added records. The goal was to find the solution in the set with the highest (optimum) performance according to the GOF. An objective function can be defined to represent the severity of the traffic crash (prediction target), seeking to be optimized. The objective functions were selected by checking the values of R, MAE, RMSE, and SSE shown in Table 4.

Based on the obtained results, the objective function given in Equation 27 had the best results for the GA model, with an R-value around 0.78 because the GA started by creating a random initial population that contains individual vector related to population and this GA process stopped when stopping criteria was met such as maximum number of generation, stall time, stall generation, fitness limit and function tolerance happen. In Table 4, objective function having the higher R was in the first row while it can be changed. To check conclusion function with different initial population vectors and also by stopping criteria then better coefficients related to our model were obtained. After checking multiple of these situations for getting better coefficient results, the improved R-value was around 0.79.

$$F = \sum_{k=1}^{n=1000} \left| X_{13} + \sum_{i=1}^{12} \left(X_{13+i} Sin\left(X_i b_{i,k} \right) \right) + \sum_{i=1}^{12} \left(X_{24+2i} Sin(X_{25+2i} b_{i,k}) \right) - Out_k \right| \quad \text{(Eq.27)}$$

Where X is the coefficient of the objective function that was optimized, b and Out parameters were related to the input and output variables, respectively.

Table 4: Objective functions used in the GA model for Iranian data (Kunt et al., 2011)

F	***R***	***MAE***	***RMSE***	***SSE***
$w_0 + \sum_{i=1}^{12} Sin(w_i X_i) + \sum_{i=1}^{12} Sin(v_i X_i)$	**0.78689**	**0.33002**	**0.43949**	**178.308**
$w_0 + \sum_{i=1}^{12} Sin(w_i X_i) + \sum_{i=1}^{12} Cos(v_i X_i)$	0.74474	0.34955	0.48068	209.6778
$w_0 + \sum_{i=1}^{12} w_i X_i$	0.60020	0.44124	0.57711	302.2494
$w_0 + \sum_{i=1}^{12} w_i X_i^{P_i}$	0.70776	0.39912	0.51465	240.3676
$w_0 + \sum_{i=1}^{12} e^{w_i X_i}$	0.46653	0.53863	0.64319	375.4189
$(w_0 + \sum_{i=1}^{12} w_i X_i) \ / (v_0 + \sum_{i=1}^{12} v_i X_i)$	0.58782	0.45016	0.59606	322.4268
$(w_0 + \sum_{i=1}^{12} Sin(w_i X_i)) / \ (v_0 + \sum_{i=1}^{12} Sin(v_i X_i))$	0.76533	0.34574	0.46290	197.1453
$(w_0 + \sum_{i=1}^{12} Sin(w_i X_i)) / \ (v_0 + \sum_{i=1}^{12} Cos(v_i X_i))$	0.74999	0.34192	0.47364	203.5874
$w_0 + Sin(w_{13} + \sum_{i=1}^{12} w_i X_i)$	0.46702	0.52028	0.70868	455.7767
$w_0 + Cos(w_{13} + \sum_{i=1}^{12} w_i X_i)$	0.41690	0.54515	0.75001	510.4594
$2 + Sin(\frac{1}{\exp(-(1 + \sum_{i=1}^{12} w_i X_i))})$	0.408693	0.48124	0.70213	447.3826

Table 5 presents the modified coefficients for the objective function. Figure 19 depicts the best and mean values of the fitness function at each generation. In

addition, the best and mean values in the current generation are shown at the top of Figure 19. Figure 20 shows the time response of the GA with regard to number of running which was around 0.687.

Table 5: Modified coefficients for objective function in the GA model with Iranian data (Kunt et al., 2011)

x_1	-0.10386	x_8	-1.61021	x_{15}	-0.1684	x_{22}	0.67988	x_{29}	-0.14108	x_{36}	0.07376	x_{43}	-1.41873
x_2	-1.18334	x_9	1.24933	x_{16}	-1.84944	x_{23}	0.26354	x_{30}	0.13037	x_{37}	4.31879	x_{44}	0.16222
x_3	0.30521	x_{10}	-0.63851	x_{17}	0.79854	x_{24}	-0.97961	x_{31}	-0.57707	x_{38}	0.91677	x_{45}	-0.29329
x_4	0.80627	x_{11}	0.20228	x_{18}	0.43804	x_{25}	-0.20209	x_{32}	-0.26776	x_{39}	-0.28983	x_{46}	0.64982
x_5	-0.61428	x_{12}	-0.40444	x_{19}	0.41867	x_{26}	0.78213	x_{33}	0.86287	x_{40}	0.69897	x_{47}	0.15646
x_6	0.55561	x_{13}	0.04129	x_{20}	0.87691	x_{27}	0.49914	x_{34}	-1.98046	x_{41}	2.90065	x_{48}	0.2271
x_7	0.81175	x_{14}	2.74527	x_{21}	-2.6484	x_{28}	0.20184	x_{35}	0.10735	x_{42}	-0.04085	x_{49}	0.17168

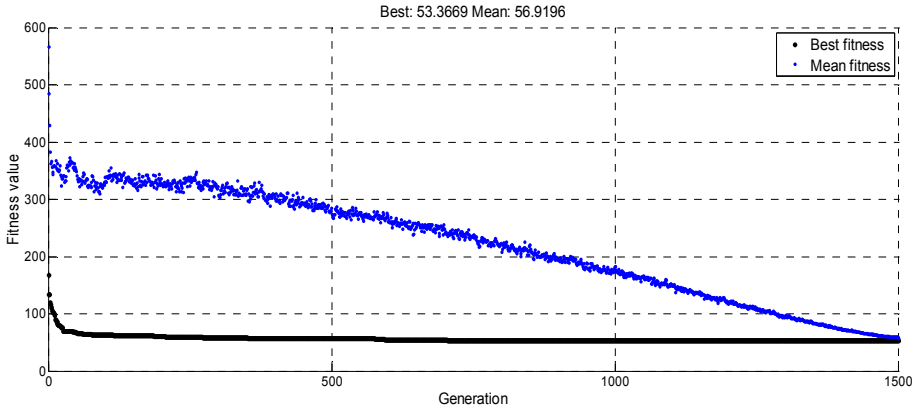

Figure 19: The best and mean values of the fitness function at each generation in the GA model for Iranian data (Kunt et al., 2011)

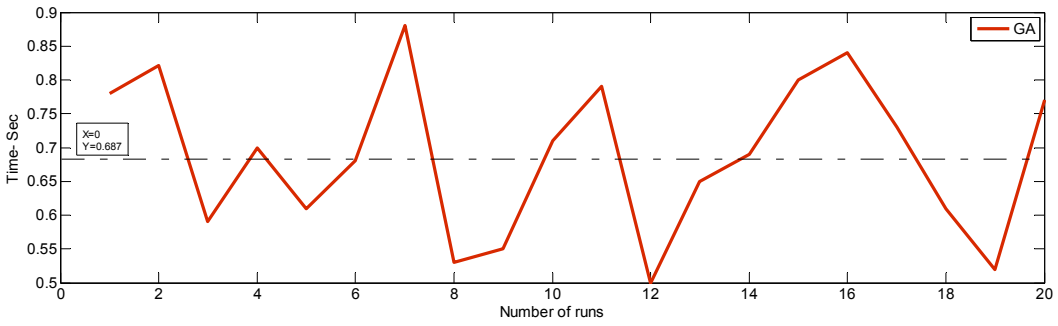

Figure 20: The time response of the GA with regard to number of runs for Iranian data

3.13.3 Combination of Genetic Algorithm and Pattern Search

The GA and PS models were combined to determine whether this combined method would achieve better results than the genetic algorithm. This research was based on GPS Positive Basic 2N, which enhanced the performance of pattern search algorithms.

The initial point of this method was obtained from the optimum point of the GA shown in Table 5. Table 6 depicts the modified coefficients of the combined model. The combined GA and PS model had an R-value of around 0.79.

Table 6: Modified coefficients for the objective function in the combined GA-PS model with Iranian data (Kunt et al., 2011)

x_1	-0.10374	x_8	-1.61021	x_{15}	-0.17632	x_{22}	0.67988	x_{29}	-0.14139	x_{36}	0.06993	x_{43}	-1.69779
x_2	-1.18334	x_9	1.24933	x_{16}	-1.84944	x_{23}	0.26354	x_{30}	0.12699	x_{37}	4.31879	x_{44}	0.18301
x_3	0.30910	x_{10}	-0.62458	x_{17}	0.79854	x_{24}	-0.97961	x_{31}	-0.57707	x_{38}	0.91677	x_{45}	-0.32155
x_4	0.80627	x_{11}	0.20228	x_{18}	0.43804	x_{25}	-0.20335	x_{32}	-0.27866	x_{39}	-0.28983	x_{46}	0.64787
x_5	-0.60150	x_{12}	-0.40445	x_{19}	0.41916	x_{26}	0.78213	x_{33}	0.86268	x_{40}	0.71681	x_{47}	0.15646
x_6	0.55622	x_{13}	0.07327	x_{20}	0.87691	x_{27}	0.49914	x_{34}	-1.98046	x_{41}	2.90065	x_{48}	0.21438
x_7	0.81175	x_{14}	2.74527	x_{21}	-2.64840	x_{28}	0.18443	x_{35}	0.10735	x_{42}	-0.03879	x_{49}	0.17168

Figure 21 shows the objective function value at the best point of each iteration for Iranian data. Typically, the value of the objective function was improved in the early iterations and then level off as they approached the optimal value. The initial point of this graph was the optimum final result of the GA.

The convergence curve in Figure 21 is typical of PS algorithms. The initial convergence occurred after the first 800 iterations, followed by progressively slower improvements as the optimal solution was approached.

Figure 22 displays the mesh size at each iteration for Iranian data. The mesh size increased after each successful iteration and decreased after each unsuccessful one. The best point did not change following an unsuccessful poll. Thus, the algorithm halved the mesh size with a contraction factor set to 0.5. The computed objective

function value at iteration 2 was less than the value at iteration 1 in Figure 22, which indicated that the poll at iteration 2 was successful. Thus, the algorithm doubled the mesh size with the expansion factor set to 2. The poll at iteration 4 was unsuccessful. As a result, the function value remained unchanged from iteration 3, and the mesh size was halved.

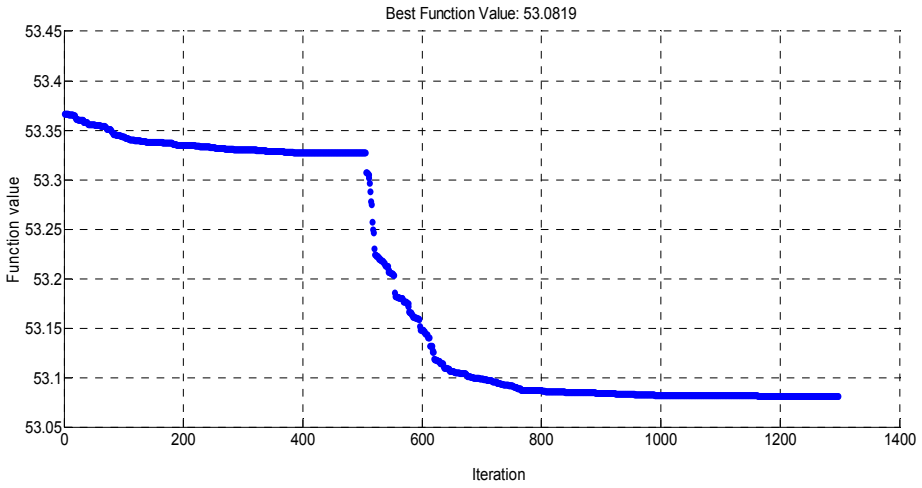

Figure 21: The function value at each iteration in the combined GA-PS model for Iranian data (Kunt et al., 2011)

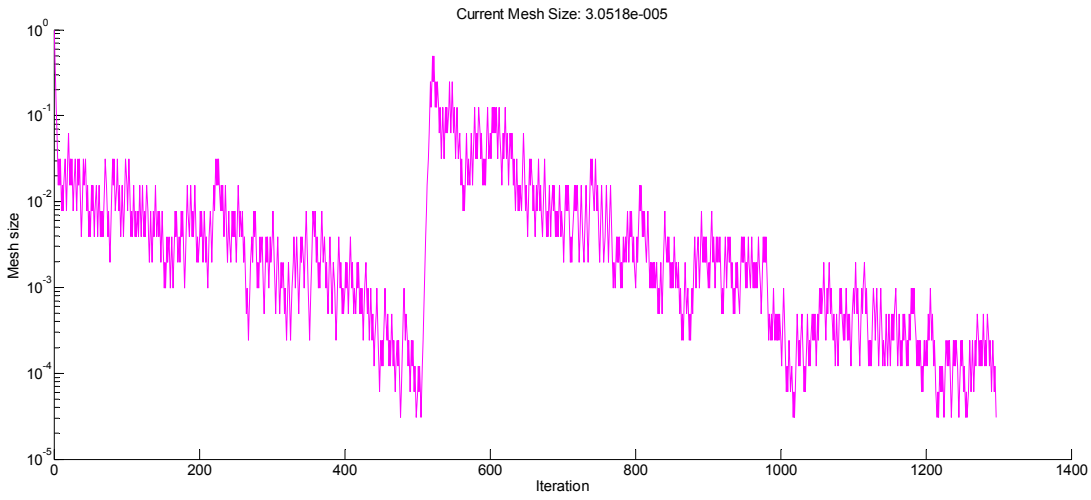

Figure 22: Mesh size at each iteration in the combined GA-PS model for Iranian data (Kunt et al., 2011)

In Figure 23, after 1297 iterations were completed, the PS algorithm performed approximately 98,000 function evaluations to locate the most promising region in the solution space containing the global minima.

Figure 24 shows the time response of GA-PS with regard to number of running which was around 0.975.

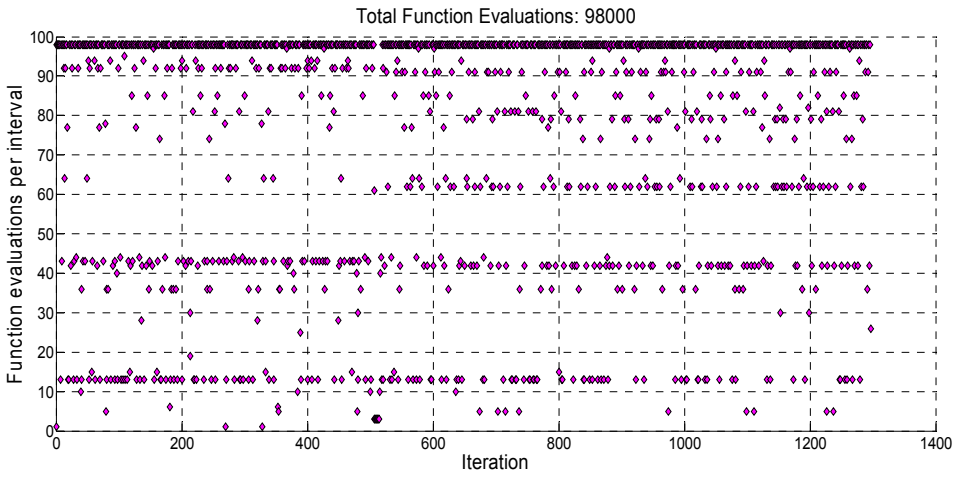

Figure 23: Function evaluation per interval in the combined GA-PS model for Iranian data (Kunt et al., 2011)

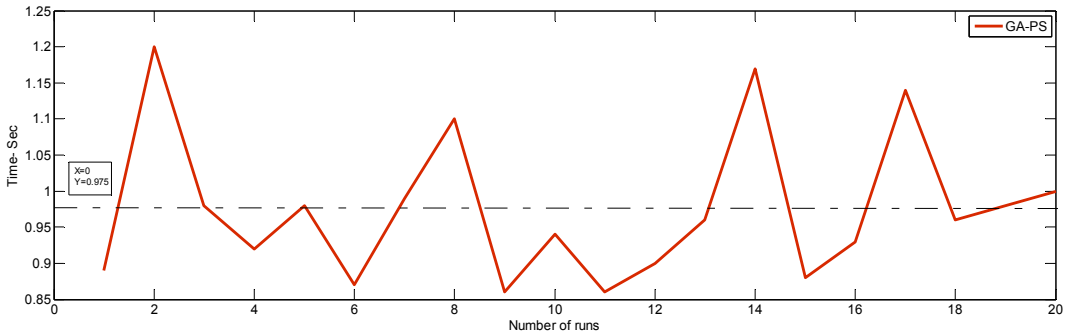

Figure 24: The time response of GA-PS with regard to number of runs for Iranian data

3.14 Models Used for Analysis with Cyprus Data

3.14.1 Multi-layer perceptron neural network

This study used a multi-layer perceptron (MLP) neural network architecture that consisted of a multi-layer feed-forward network with sigmoid hidden neurons and linear output neurons as well as a network that was trained with the Levenberg-Marquardt back-propagation algorithm.

The MLP model consisted of two layers, with each layer having a weight matrix W, a bias vector b, and an output vector p^i, with $i > 1$. Figure 25 shows the selected final prediction model for each layer in the MLP model where the number of the layer was appended as a superscript to the variable. For the different weights and other elements of the network, superscripts were applied to recognize the source (second index) and the destination (first index). Layer weight *(LW)* matrices and input weight *(IW)* matrices were used in the MLP model.

The model was applied to data that were randomly divided into sets for model training, test, and validating. The MLP model had 7 inputs, 20 neurons in the first layer, and 3 neurons in the second layer. The output layer of the MLP model consisted of three neurons representing the three levels of injury severity. Of the original data, 70% were used in the training phase. While the validation and test data sets each contained 15% of the original data. The MLP structure for Cyprus data was almost same with the MLP structure used in Iranian data as mentioned before.

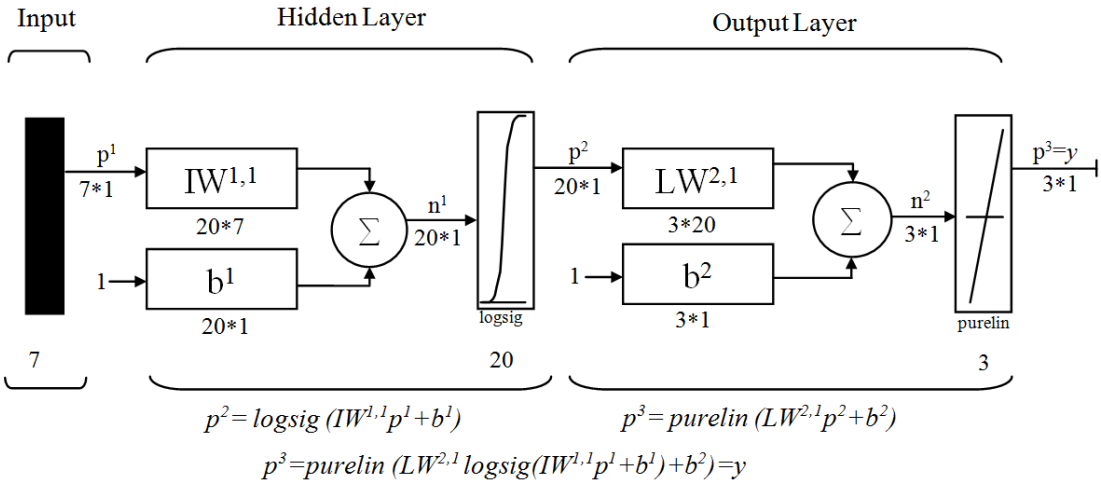

Figure 25: The structure of final MLP model for Cyprus data (Aghayan et al., 2012)

The MLP, which was applied for training, test, and validation, consisted of 7 inputs, 20 neurons in the hidden layers, and 3 neurons in the output layer. The data for training, validation, and test of the MLP application represented 70, 15, and 15 percent of all crash data, respectively. The results of the MLP model are shown in Table 7 for 20 runs, which tabulates the prediction levels of injury severity patterns in the training, test, and validation phases.

Table 7: Prediction table for MLP model with Cyprus data (Aghayan et al., 2012)

R	No Injury	Evident Injury	Fatality	Overall
Training	0.7383	0.8819	0.8805	0.9102
Validation	0.6449	0.7208	0.7408	0.8115
Test	0.5291	0.8259	0.8723	0.8547
All	0.6783	0.8623	0.8673	0.8948

Figure 26 depicts regression plots for the output with respect to training, validation, and test data. The value of the correlation coefficient (R) for each phase was calculated. The R-value was around 0.89 for the total response in the MLP model.

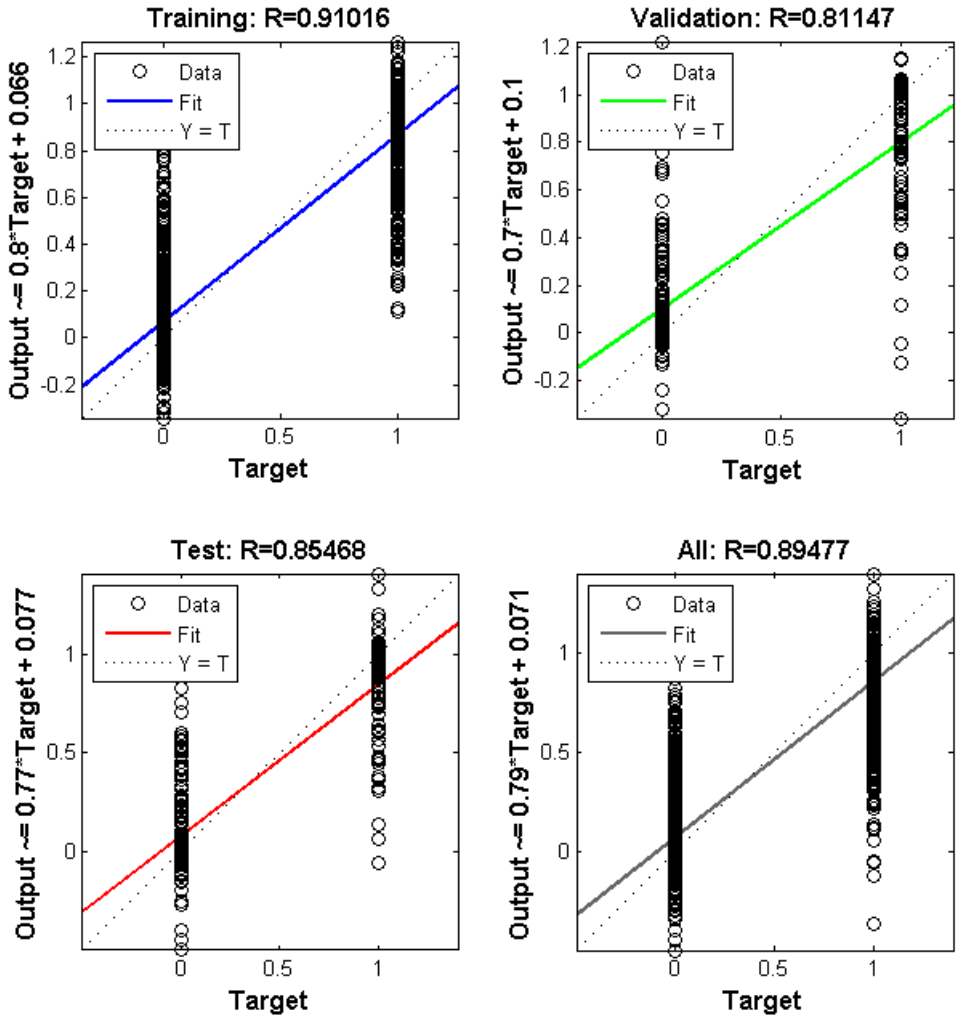

Figure 26: The regression plots for training, test, validation phases and total response in the MLP model with Cyprus data

Figure 27 plots the training, validation, and test errors to determine the validation error in the training window. The best validation performance happened at Iteration 15. Figure 27 shows the mean squared error of the network starting at a large value and decreasing to a smaller value, which means that the network learning was improving. Training on the training vectors continued as long as the training reduced the network's error on the validation vectors. The behavior of that plot and the procedure used is similar to Figure 17 mentioned earlier.

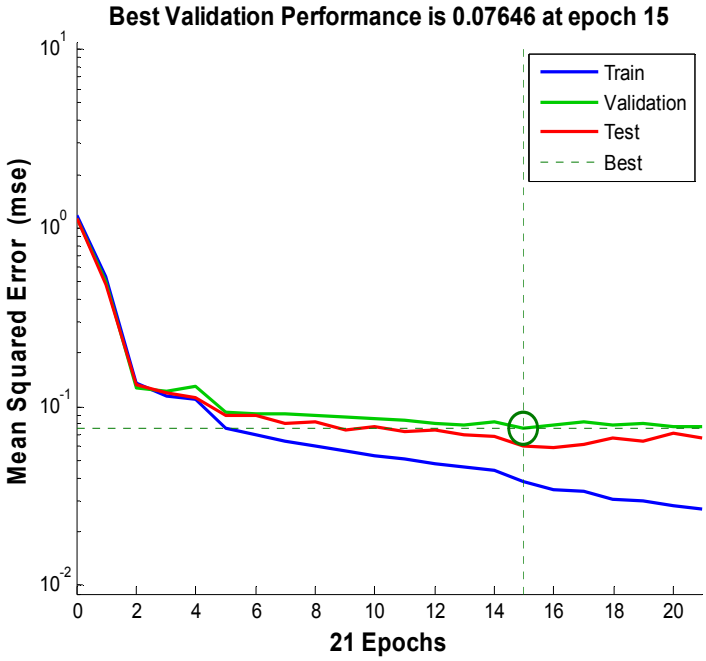

Figure 27: The validation error in the MLP model for Cyprus data

Figures 28 and 29 show the time response and R-value of the MLP model with regard to the number of runs; those values were 2.635 and 0.892, respectively.

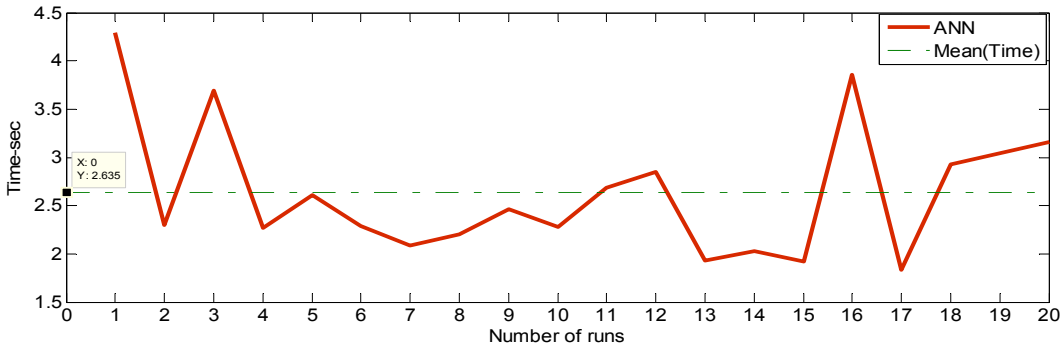

Figure 28: The time response of MLP model with regard to number of runs for Cyprus data

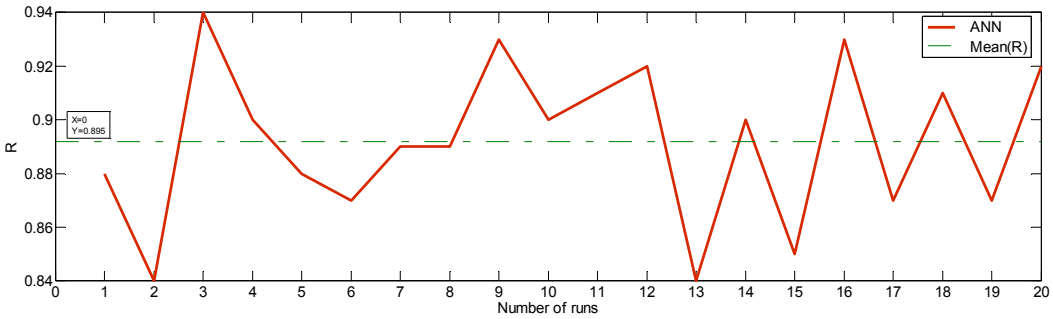

Figure 29: The R-value of MLP model with regard to number of runs for Cyprus data

3.14.2 Comparison of clustering models

Use of a clustering methodology resulted in the optimum number of MFs. Figure 30 depicts the influence of the number of clusters along with the various radii in subtractive clustering. Figure 31 shows the R-value for a given radius in the FS clustering algorithm. In Figures 30 and 31, the minimum number of clusters was 10, and the R-value achieved by the 10 clusters was 0.855. In addition, Figure 32 shows the relationships between the number of clusters and the mean silhouette coefficient. It was found that, when the number of clusters was increased, the mean silhouette coefficient, which represented the overall quality of the clustering measurement, was decreased. In addition, Figure 32 shows that the mean silhouette coefficients for hierarchical, K-means, and FCM clustering converged to 12 clusters. As explained above, by increasing the number of clusters, the R-value increased and the mean silhouette coefficient was decreased. Therefore, to satisfy two different evaluations for the cluster validity, 12 clusters were selected, which was more than the minimum number of 10 clusters obtained from subtractive clustering in Figure 30.

All clustering algorithms identified 12 clusters, which meant that each input and output was characterized by 12 membership functions. Moreover, the number of rules equaled the number of clusters, and hence, 12 rules were created.

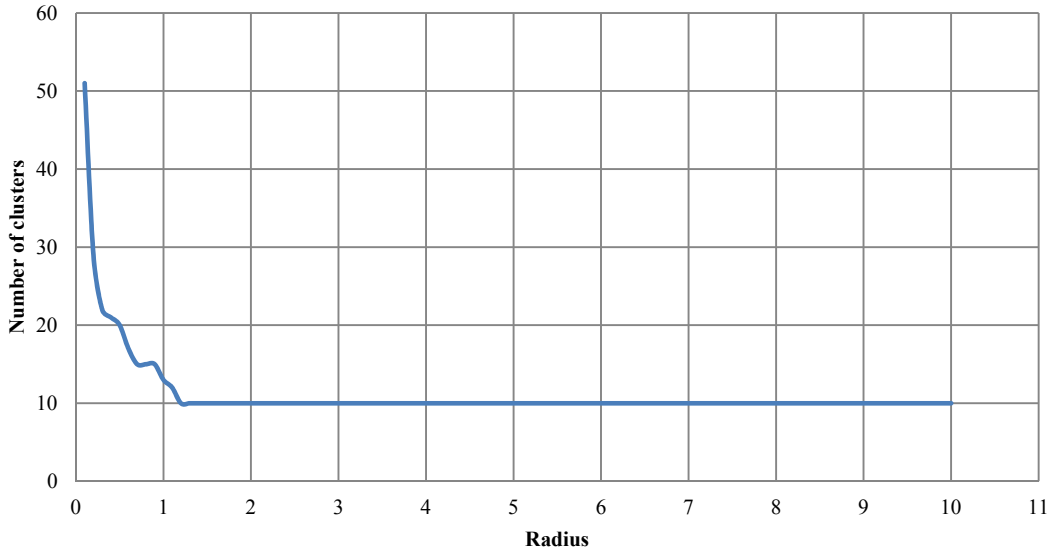

Figure 30: The influence of the number of clusters with given radius in subtractive clustering for Cyprus data (Aghayan et al., 2012)

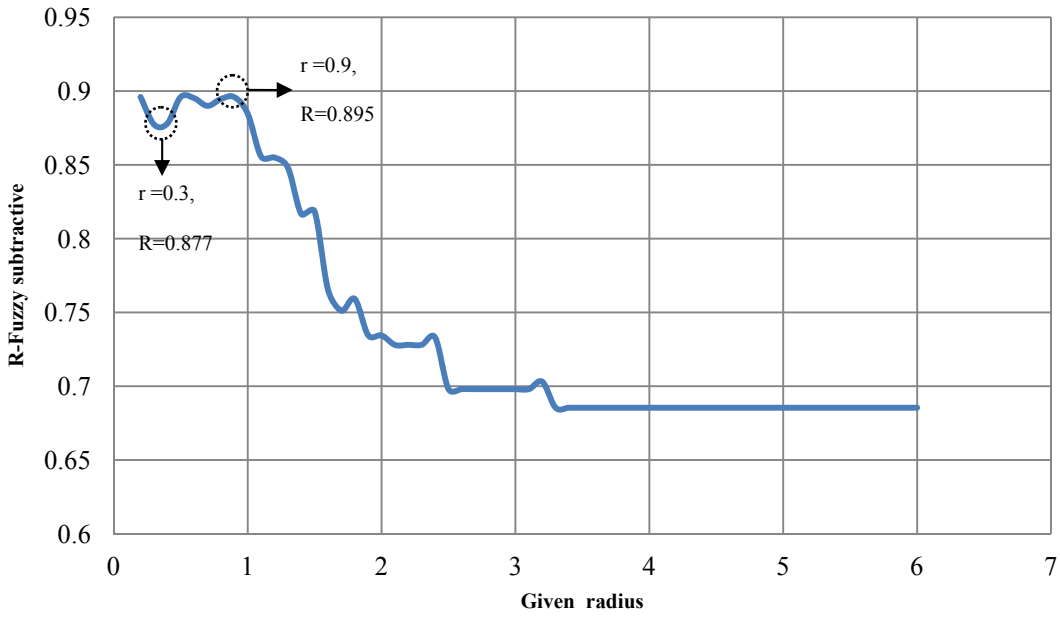

Figure 31: R-values for given radii in FS for Cyprus data (Aghayan et al., 2012)

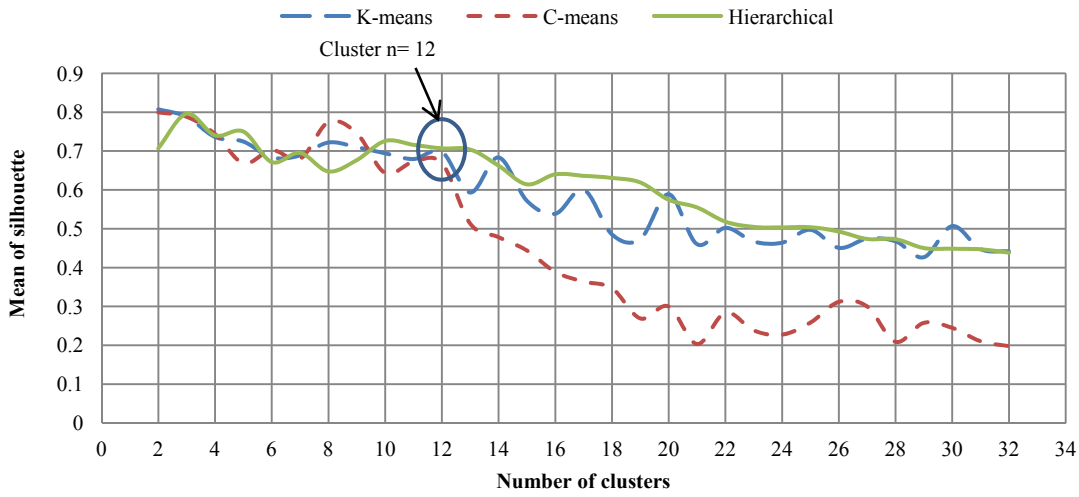

Figure 32: Comparing the mean silhouette values in K-means, hierarchical, and FCM clustering for Cyprus data (Aghayan et al., 2012)

3.14.3 Fuzzy C-Means clustering

Figure 33 shows the structure of the FCM clustering, which consists of the seven input variables, each of which was connected to the 12 membership functions of the clusters. After processing according to the rule base, the MF's output were obtained, which led to the aggregated outputs. Finally, the best aggregated output was selected as the output results.

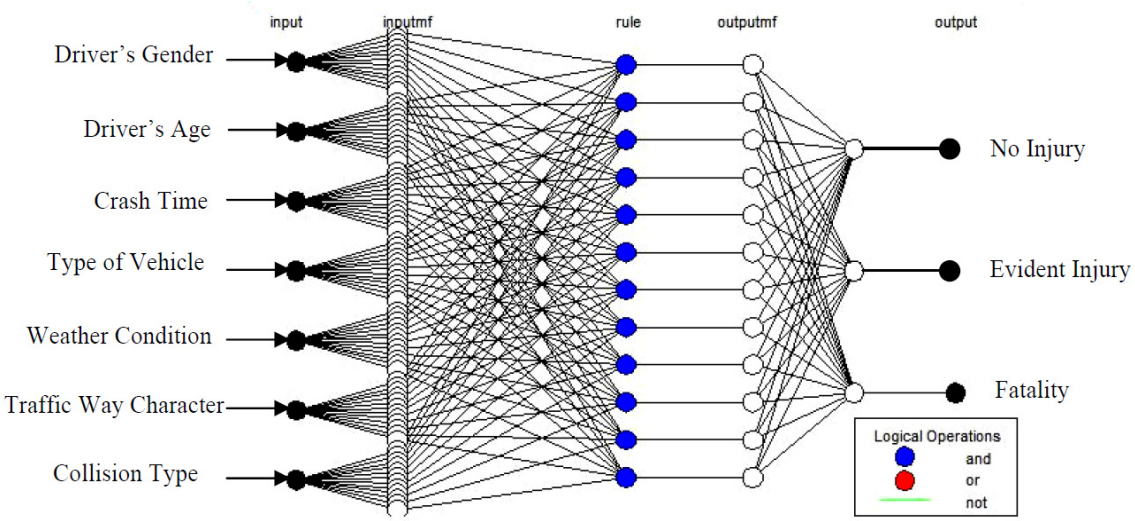

Figure 33: Graphic representation in the FCM clustering algorithm for Cyprus data (Aghayan et al., 2013)

Figures 34 and 35 show an example of the MFs of the collision type and driver's age obtained from the FCM clustering algorithm. The actual values and predicted values for 15% of the training and checking data are shown in Figures 36, 37, respectively. The output values of the three levels (no injury: 1, evident injury: 2, fatality: 3) were either 0 or 1, as shown in Table 2. The mean response time of the FCM approach for 20 runs was 0.474 seconds shown in Figure 38.

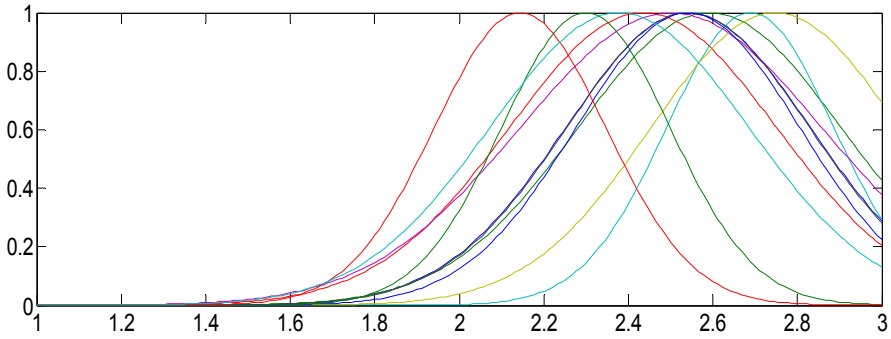

Figure 34: MF for collision type in the FCM clustering with Cyprus data (Aghayan et al., 2012)

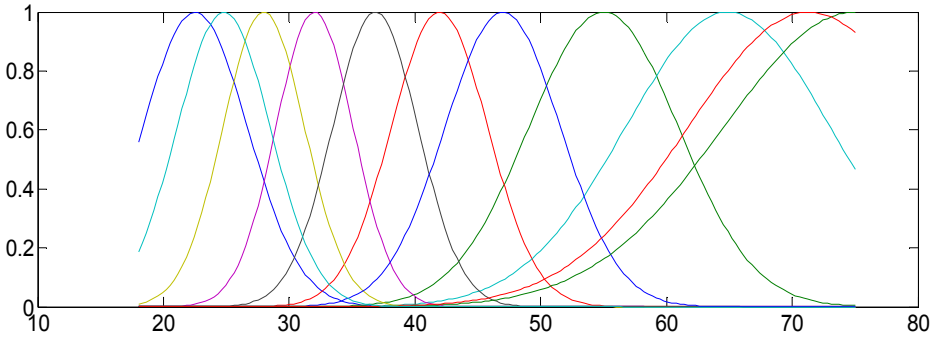

Figure 35: MF for driver's age in the FCM clustering with Cyprus data (Aghayan et al., 2012)

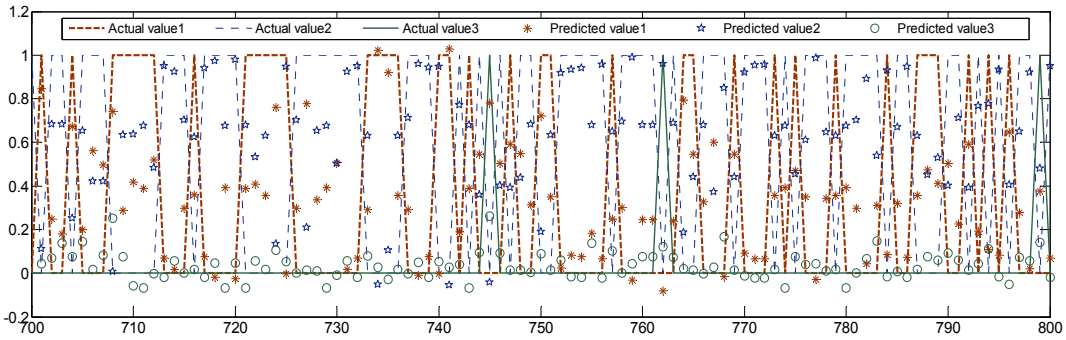

Figure 36: Comparison of actual and predicted values for training data in the FCM clustering with Cyprus data

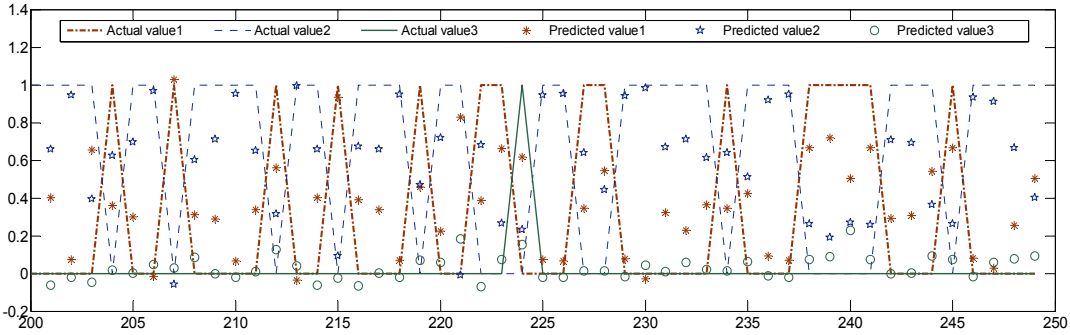

Figure 37: Comparison of the actual and predicted values for checking the data in the FCM clustering with Cyprus data (Aghayan et al., 2012)

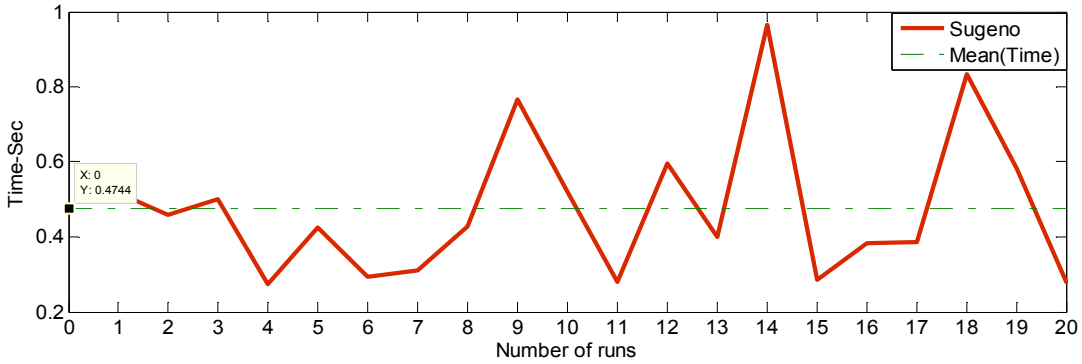

Figure 38: The time response of the FCM clustering with regard to number of runs for Cyprus data (Aghayan et al., 2013)

3.14.4 Fuzzy subtractive clustering

Subtractive clustering is a fast, one-pass algorithm to determine the approximate number of clusters and the cluster centers in the training dataset. However, in FS clustering, both input and output training data generates a Sugeno-type FIS structure.

In this case, the cluster radius indicated the range of influence of a cluster when the data space as a unit hypercube was considered. Specifying a small cluster radius usually yielded many small clusters in the data and resulted in many rules. Specifying a large cluster radius usually yielded a few large clusters in the data and results in fewer rules. Figures 39 and 40 present an example of the MFs of the collision type and the driver's age as used in FS clustering, respectively. When compared with Figures 34 and 35, the figures revealed that the FS clustering algorithm had a lower computational cost than the FCM clustering because of a smaller number of MFs and rules. The actual and predicted values based on the coding variables given in Table 2 are shown in Figures 41 and 42 for 15% of the training and checking data, respectively. In few circumstances, the predicted values were out of boundary condition defined to be between 0 and 1; meanwhile, overall similarity between actual and predicted values was not affected by errors happened in predicted values. The mean response time of the FS clustering for 20 runs was 0.284 seconds shown in Figure 43.

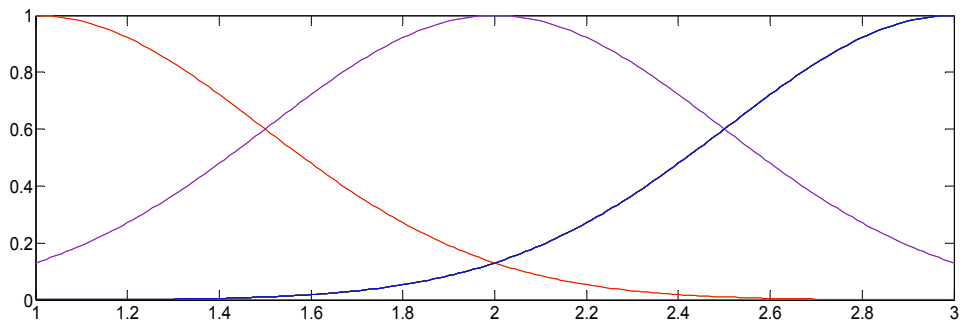

Figure 39: MF for collision type in the FS clustering with Cyprus data (Aghayan et al., 2012)

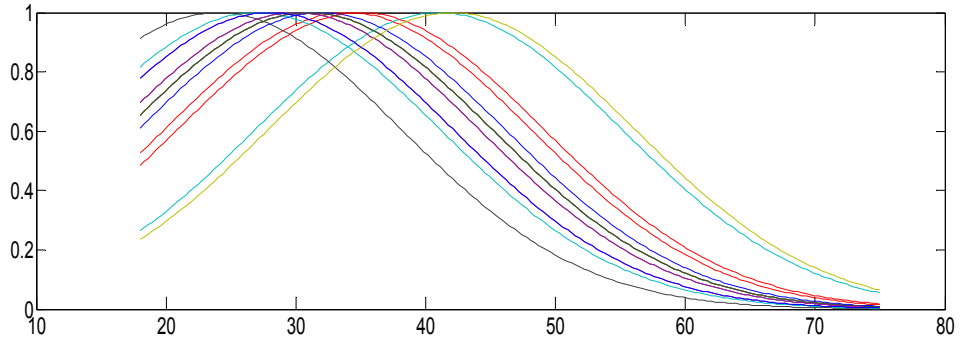

Figure 40: MF for driver's age in the FS clustering with Cyprus data (Aghayan et al., 2012)

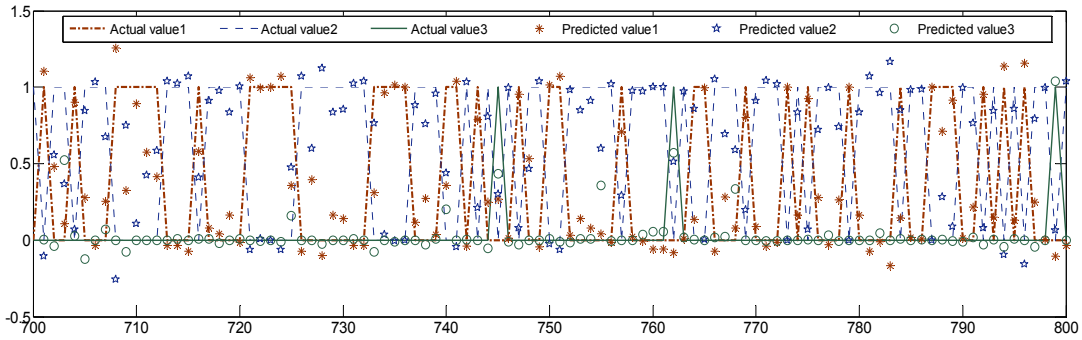

Figure 41: Comparison of actual and predicted values for training data in FS clustering with Cyprus data

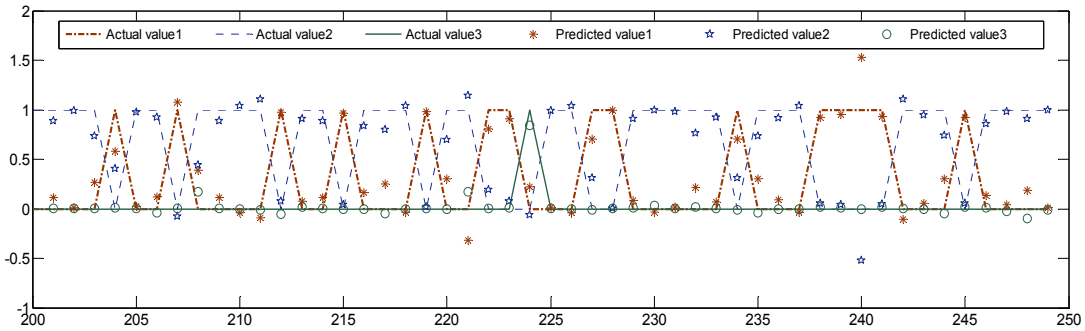

Figure 42: Comparison of actual and predicted value for checking the data in FS clustering with Cyprus data (Aghayan et al., 2012)

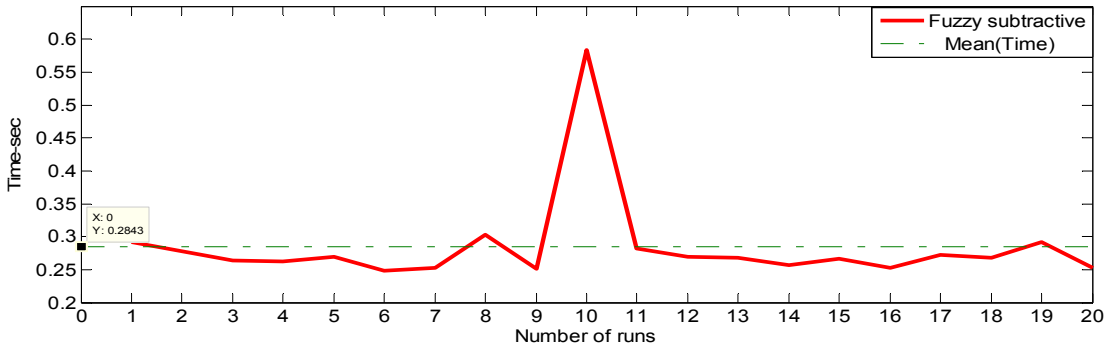

Figure 43: The time response of FS clustering with regard to number of runs for Cyprus data

Chapter 4

RESULTS AND DISCUSSION

4.1 Iranian Data

This study used the ANN, GA, and GA-PS models to predict the severity of traffic crashes. The final results showed that the ANN performed better than the GA and the combined GA and PS models based on the accuracy, while the lowest response time was achieved with the GA.

Table 8 presents the R, MAE, RMSE, SSE, and t values. These results demonstrate that the MLP and GA constructed are promising for modeling traffic injury severity according to accuracy and response time, respectively.

The GA provided the lowest elapsed time, 0.687 seconds, followed by GA-PS, with an elapsed time of 0.975 seconds. GA used the least amount of time with the accuracy that was less than MLP's accuracy, while MLP model used an exhaustive search in the greatest amount of time with the best accuracy. Thus, if a fast prediction model is the goal, GA can be the right choice, but if accuracy is the main concern, then MLP is the best choice.

Table 8: Final results for the objective function in each model with Iranian data (Kunt et al., 2011)

Used Model	MAE	RMSE	SSE	R	T(sec)
GA	0.323436	0.43992	175.628	0.792411	0.687
GA-PS	0.321709	0.437782	173.9248	0.793479	0.975
MLP	0.16178	0.22979	123.4373	0.87319	7.627

Figures 44, 45, and 46 compares the actual output crash severity values with the predicted values for the MLP, GA, and GA-PS models by considering 15% of the data, respectively. This graphical presentation depicts considerable overlap between the actual and predicted graphs, showing that the models successfully predict traffic crash severity with a high accuracy.

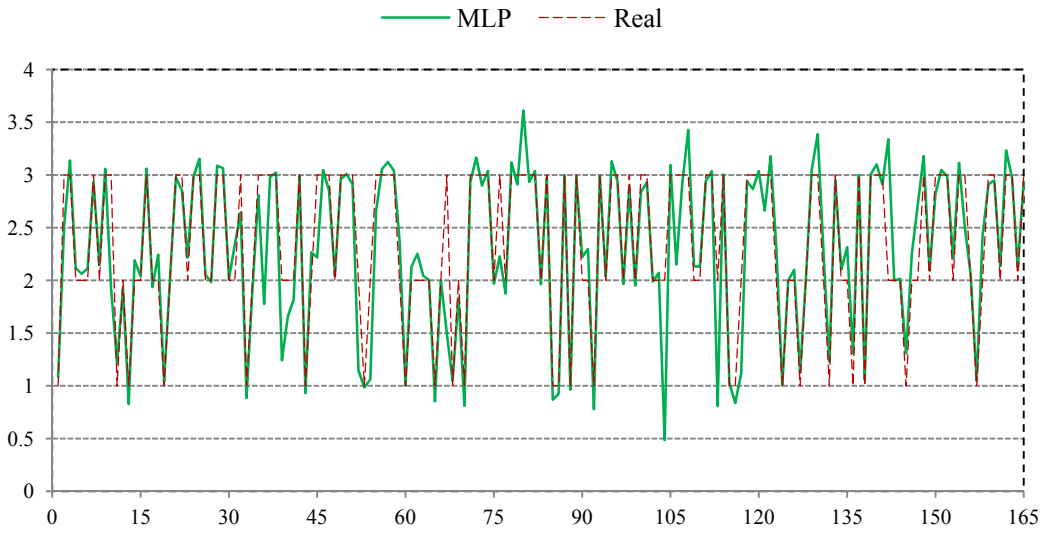

Figure 44: Comparing the actual and predicted values in MLP for Iranian data

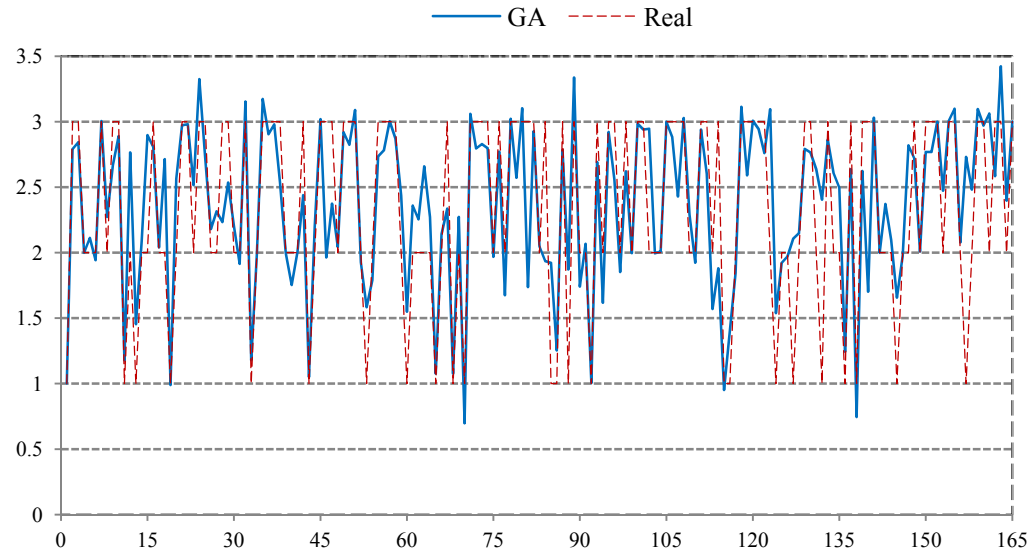

Figure 45: Comparing the actual and predicted values in GA for Iranian data

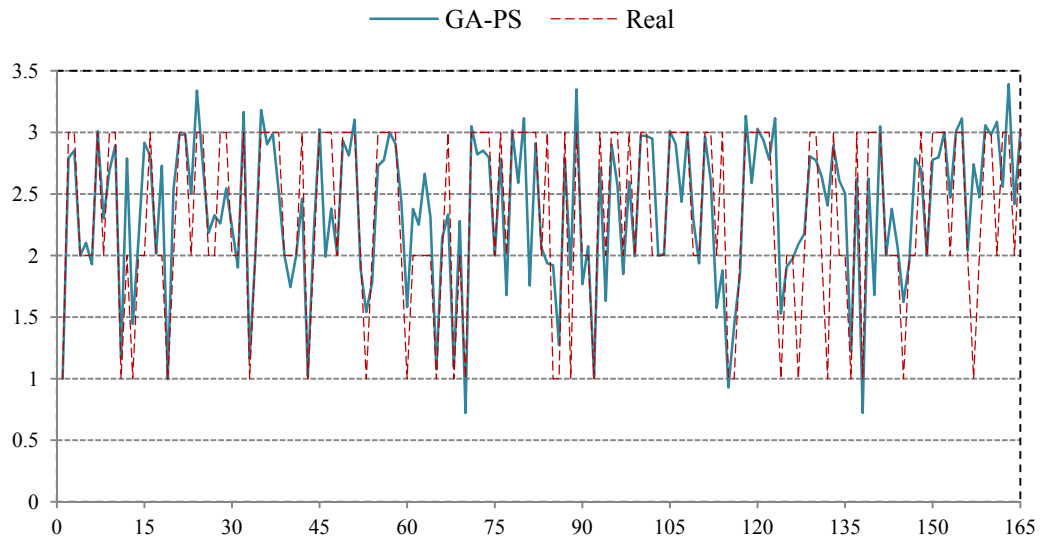

Figure 46: Comparing the actual and predicted values in GA-PS for Iranian data

Figure 47 represents the relationship between the number of program runs and the response time for the MLP, GA, and GA-PS models. The mean response time for GA was less than that for the other models.

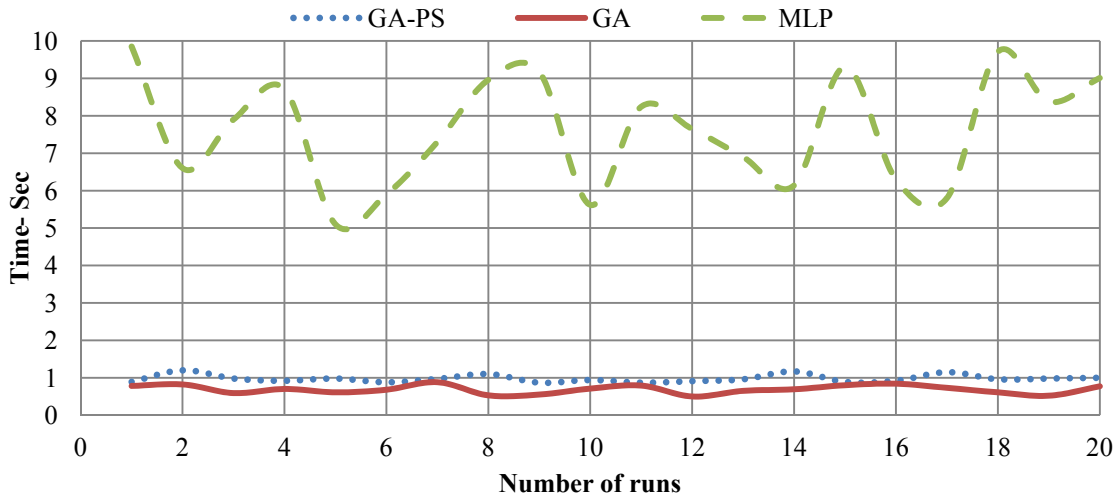

Figure 47: Comparing the response time among the prediction models used for Iranian data

Figure 48 shows regression plots for the output with regard to fatality, evidence injury, and no-injury in moreover, the R-value for each crash severity level was estimated in which the R-value of no-injury was more than other models.

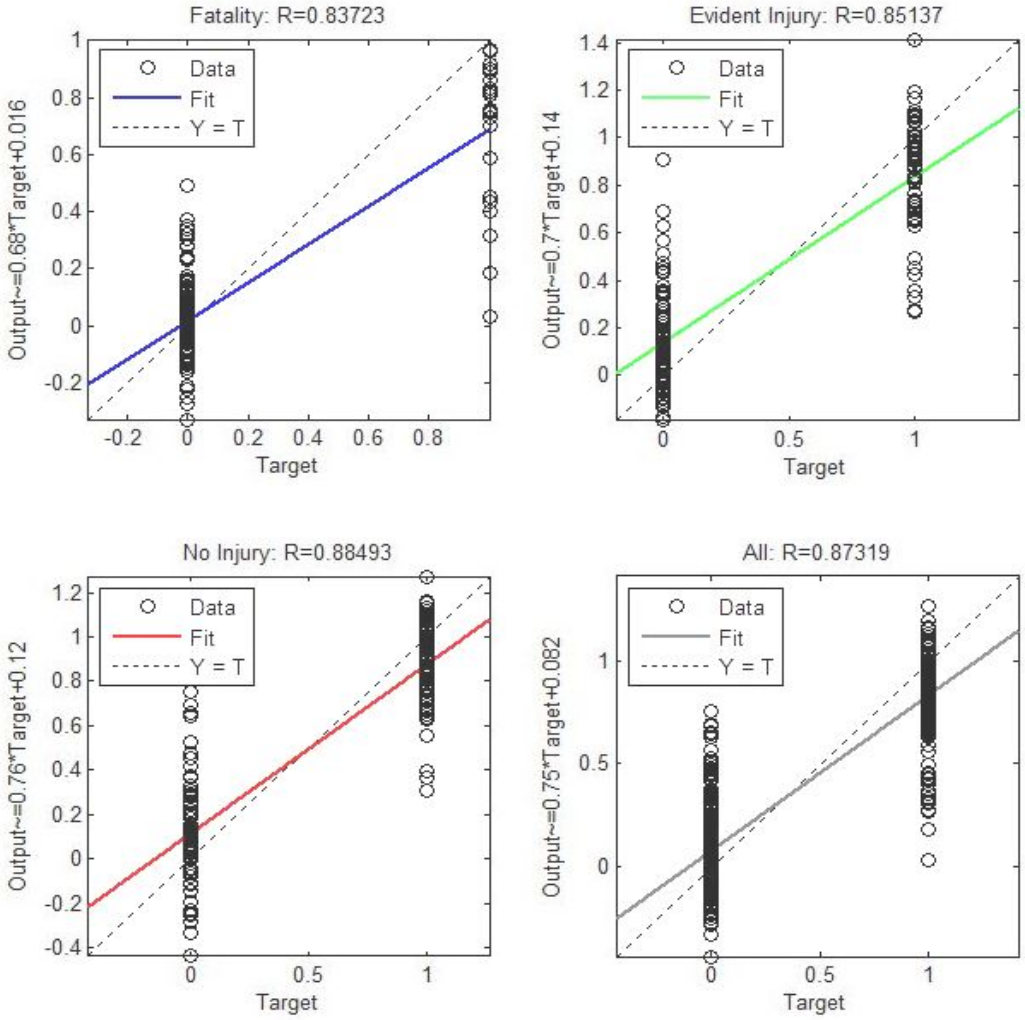

Figure 48: The regression plots for each crash severity level in the MLP model with Iranian data (Kunt et al., 2011)

4.2 Cyprus Data

The relationship between the R-value and the number of clusters for FCM clustering calculated by the Mamdani and Sugeno fuzzy algorithms is represented in Figure 49. With the Sugeno fuzzy algorithm, the R-value was not related to the cluster number; whereas, with the Mamdani fuzzy algorithm the R-value increased approximately with the number of clusters. Figure 50 shows the relationship between the R-value and the number of clusters in the FS clustering. The R-value was 0.855 for the FS clustering with 12 clusters. Figures 49 and 50 show that increasing the

number of clusters typically caused the R-value to increase, but Figure 32 shows that such an increase in the number of clusters caused the mean silhouette coefficient to decrease. Subsequently, 10 was the optimum number of clusters determined by subtractive clustering as shown in Figure 30, the mean silhouette coefficient converged at 12 clusters, as shown in Figure 32. Then, because 12 was greater than 10, 12 clusters were selected for the fuzzy clustering algorithms.

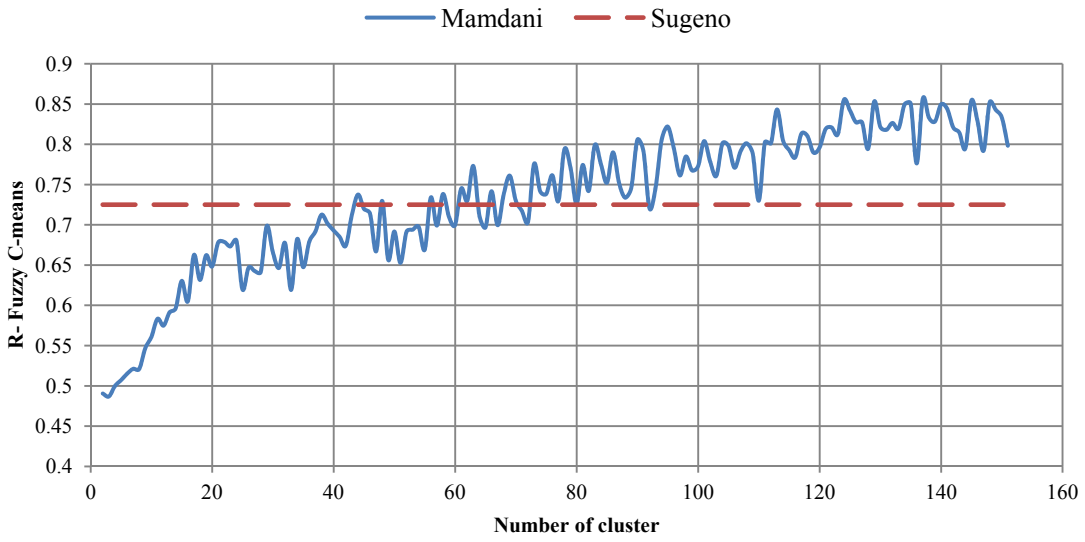

Figure 49: The R-values in FCM for Cyprus data (Aghayan et al., 2012)

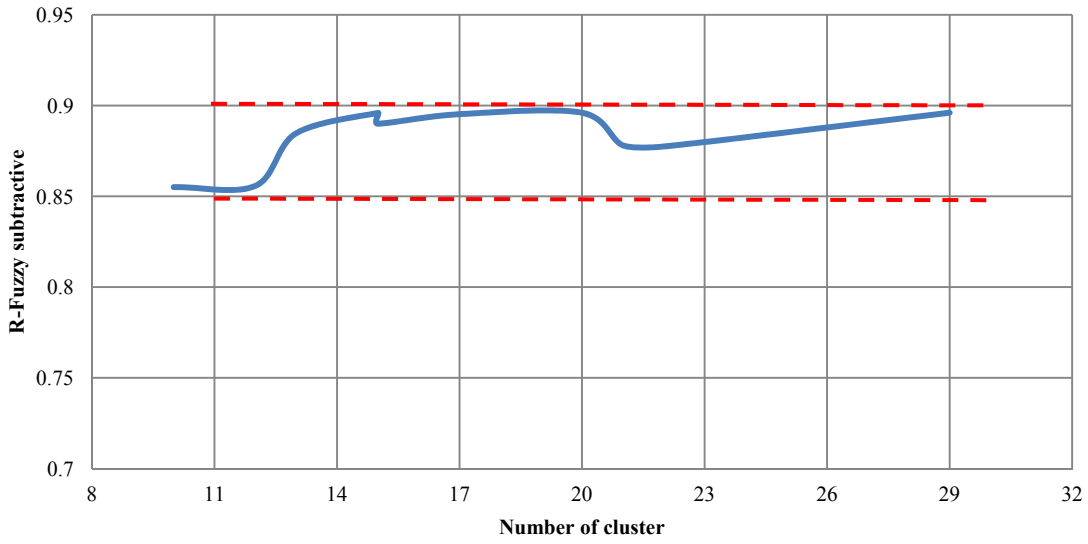

Figure 50: The R-values in FS for Cyprus data (Aghayan et al., 2012)

Figure 51 represents the relationship between the number of program runs and the response time for the MLP, FS clustering and FCM clustering. The mean response time for FS clustering was less than that for the other models.

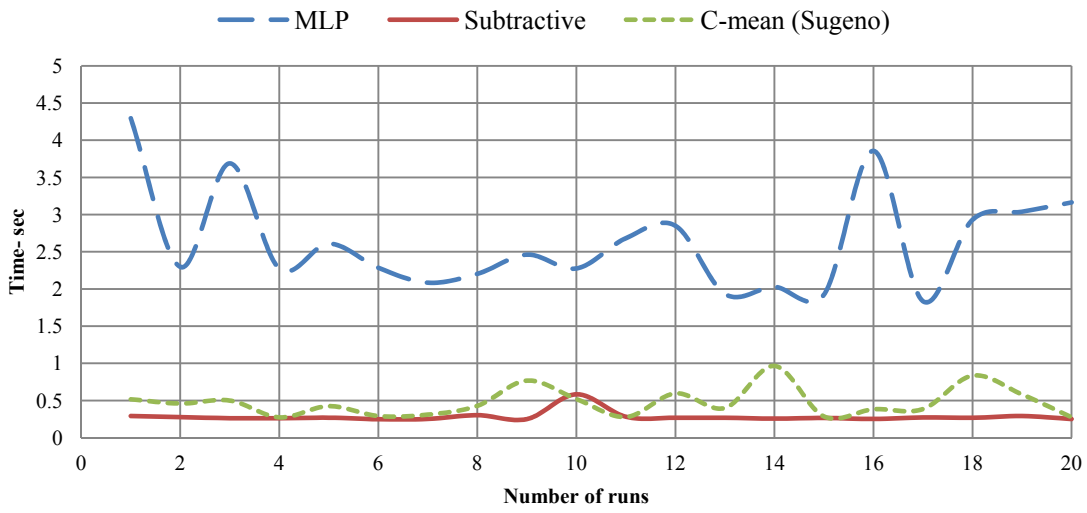

Figure 51: Comparing the response time among the prediction models used for Cyprus data (Aghayan et al., 2012)

The MLP, FCM clustering, and FS clustering were compared on the basis of their R, MAE, SSE, and RMSE values, as shown in Table 9. The FS clustering provided the lowest elapsed time, 0.284 seconds, followed by FCM clustering, with an elapsed time of 0.474 seconds. FS clustering used the least amount of time with the accuracy that was less than MLP's accuracy, while MLP model used an exhaustive search in the greatest amount of time with the best accuracy. Thus, if a fast prediction model was the goal, FS clustering could be the right choice, but if accuracy was the main concern, then MLP was the best choice.

Table 9: Final results for the objective function in each model with Cyprus data (Aghayan et al., 2012)

Used Model	MAE	SSE	MSE	RMSE	R	T(sec)
Multi-Layer Perceptron (MLP)	0.129	132.927	0.044	0.211	0.895	2.635
Fuzzy Subtractive Clustering	0.148	159.62	0.060	0.245	0.855	0.284
Fuzzy C -Means Clustering (TSK)	0.247	245.31	0.120	0.347	0.725	0.474

Figures 52, 53, and 54 compare the actual output crash severity values with the predicted values for the MLP, FS, and FCM models by considering 15% of the data, respectively. This graphical presentation depicts considerable overlap between the actual and predicted graphs, showing that the models successfully predict traffic crash severity with a high accuracy.

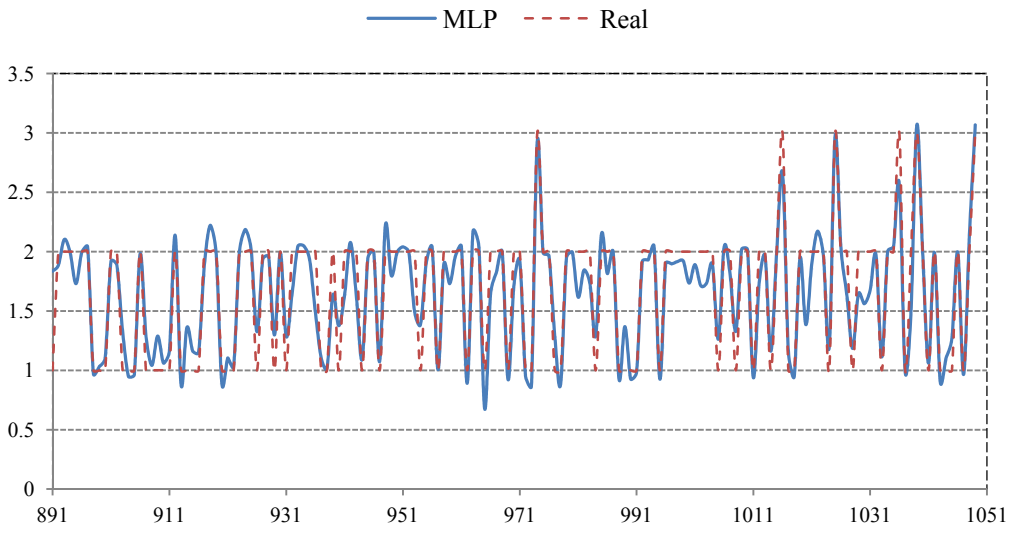

Figure 52: Comparing the actual and predicted values in MLP for Cyprus data

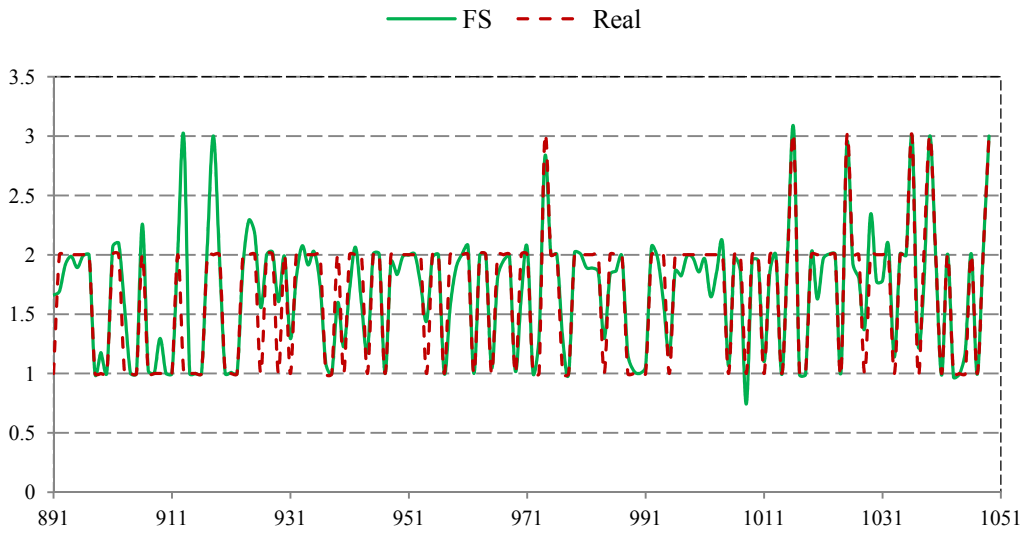

Figure 53: Comparing the actual and predicted values in FS for Cyprus data

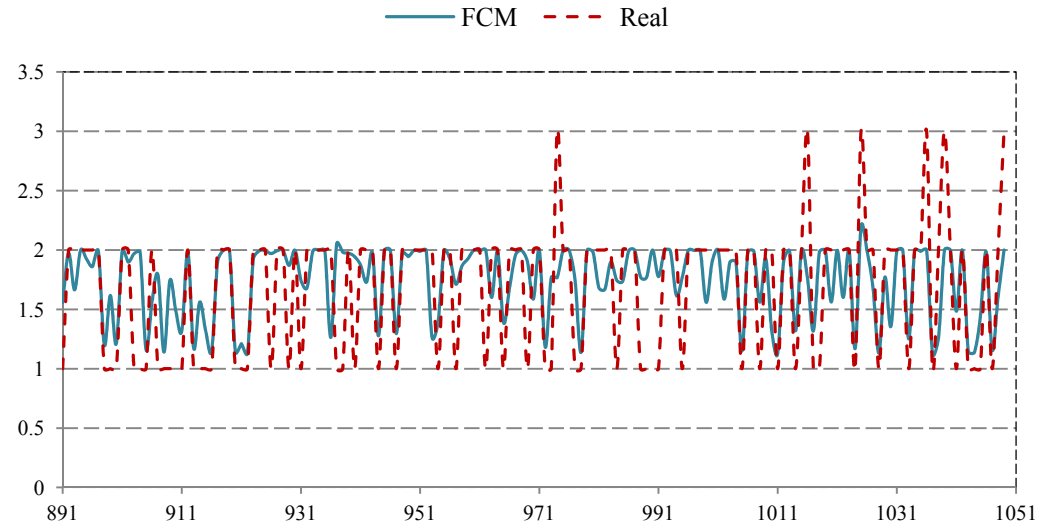

Figure 54: Comparing the actual and predicted values in FCM for Cyprus data

Figures 55, 56, and 57 display the residuals, which were the differences between the actual values and predicted values for each model. In these figures, the abscissa shows 1049 data points while the ordinate shows residuals for each cluster.

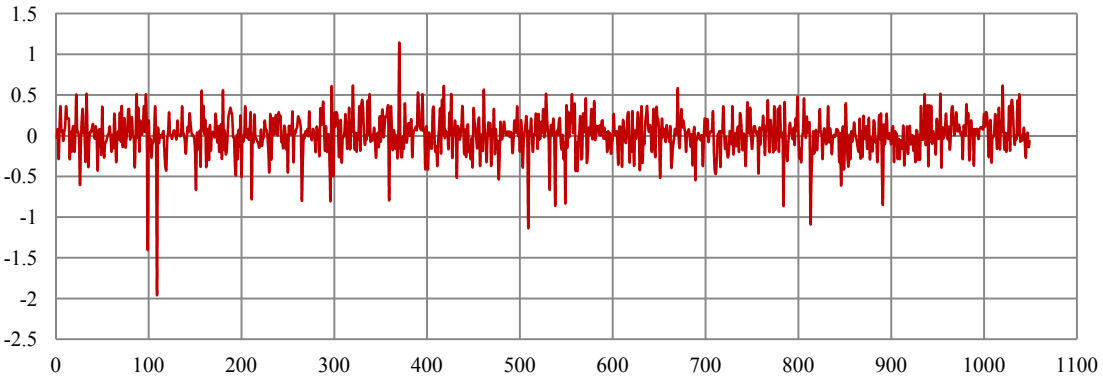

Figure 55: The residuals for MLP model with Cyprus data (Aghayan et al., 2012)

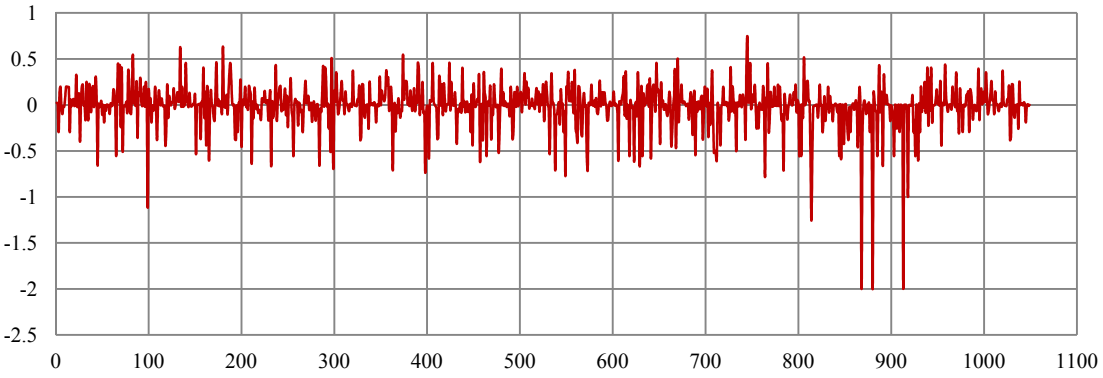

Figure 56: The residuals for the FS clustering model with Cyprus data (Aghayan et al., 2012)

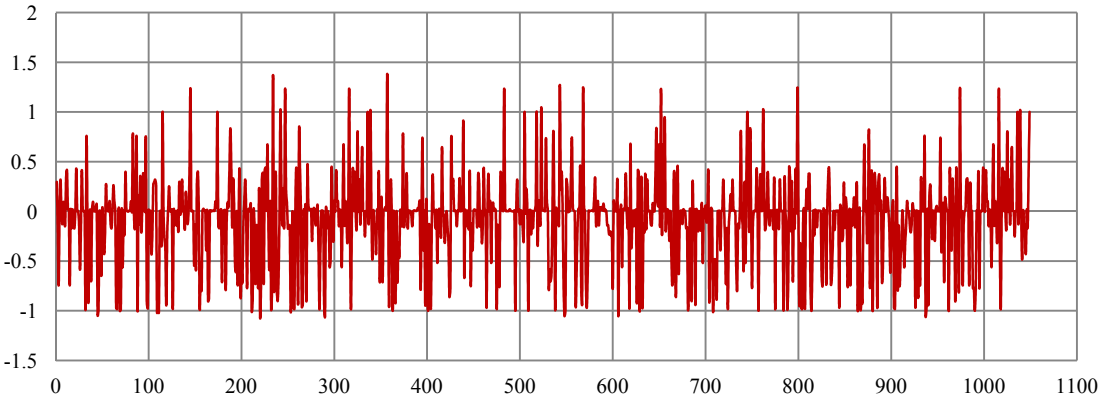

Figure 57: The residuals for the FCM model with Cyprus data (Aghayan et al., 2012)

The regression plots for three levels of crash severity are shown in Figure 58. The amount of R-value which was mentioned above the plot represented the prediction accuracy of MLP model for each level.

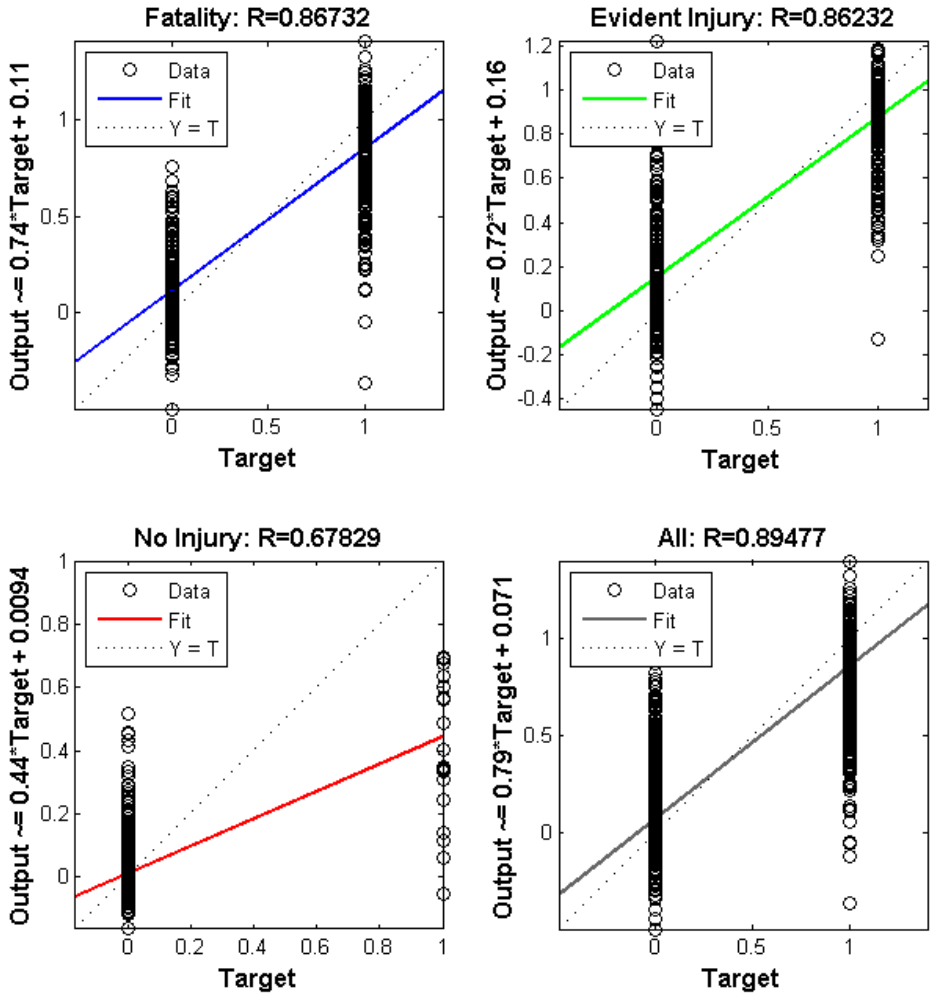

Figure 58: The regression plots for each crash severity level in the MLP model with Cyprus data

Chapter 5

CONCLUSION

Based on Iranian data, the GA, combined GA and PS, and ANN with MLP architecture were used to predict traffic crash severity and to estimate the response time using twelve input parameters and three injury severity levels. The performance of these models was compared to find the most suitable model for the prediction of crash severity and the response time at three levels: fatality, evident injury, and no injury. The following results were obtained from Iranian data.

1. The MLP was applied at training, testing and validation and had 12 inputs, 25 neurons in the hidden layers and 3 neurons in the output layer. The data for training, validation and testing of the MLP application represented 70, 15 and 15 percent of all crash data, respectively. The R-value and response time for ANN was around 0.87 and 7.627, respectively.
2. The GA alone performed as well as the combined GA and PS model in predicting the crash severity. The MLP provided the highest prediction accuracy with an R-value around 0.87, followed by the combination of GA and PS with an R-value around 0.79 and GA at 0.79, while the great amount of time and the lowest elapsed time were achieved from the MLP and GA models, respectively. Therefore, for this dataset, the MLP constructs a better relationship between the twelve model input parameters and the crash severity. The advantage of using the GA or the

combined GA and PS model was that the functions and coefficients of the relationships were known; meanwhile, the limitation of the MLP is that to include the black box approach. Thus, each model had its own advantage, and using more than one method may provide a better understanding of the relationship between the input and output variables.

3. The models were constructed to be able to incorporate additional data. In addition, the optimum coefficients of the objective function are the initial optimum vector in the combined GA and PS model. For reaching optimum results with the MLP model, a new weight and bias were calculated from the preliminary weight matrix and bias vector.
4. The use of more than one model investigated in this research provided a complete understanding of the relationship between input and output variables and allowed for high prediction accuracy (MLP) as well as low response time (GA).

Based on Cyprus data, this study compared the FS clustering, FCM clustering, and MLP models to identify the model best suited for predicting traffic crash severity by using seven input parameters at three levels, fatality, evident injury, and no injury, as well as estimating the response time for processing traffic crash data. The following results were obtained from Cyprus data.

1. Twelve clusters were obtained from four clustering algorithms - hierarchical, K-means, FCM, and subtractive clustering - as the optimum number of clusters, as at this value, the mean silhouette coefficient and R-value converged in the clustering algorithms. Clustering should be applied to the input and output of the training records, which comprised approximately 800 records of the overall used data. The optimum number

of clusters and the number of rules should be equal; therefore, 12 rules were created. In addition, each input and output was characterized by 12 membership functions. This number of clusters was applied to the FS clustering and FCM clustering.

2. Our procedure was able to identify the best two models based on accuracy (R) and response time (t). MLP model via exhaustive search took the greatest amount of time (2.635 seconds) with the best accuracy (R-value of 0.89). However, FS clustering took the least amount of time (0.284 seconds) with accuracy that R-value was 0.85. Thus, if a faster modeling time was desired, then FS clustering can be the right choice, but if accuracy is the goal, then MLP can be selected. In certain circumstances, MLP model could give higher accuracy, but MLP model would take more than 2.635/0.284 = 9.28 times longer to yield an answer than FS clustering.

3. The comparison of multiple models in this research provided a complete understanding of the relationship between input and output variables and allowed for identification of models yielding the highest prediction accuracy (MLP) and lowest response time (FS).

Overall, the findings showed that more than one model can be suitable, depending on the selected criterion (accuracy and response time). The MLP can be the best model to predict the traffic crash severity regardless of the variables involved with crash data in which the accuracy was the important criterion. While high accuracy resulted in better prediction levels of the crash severity, low response time can allow the developed system to assist agencies in performing real-time prediction with data

from detectors and/or real-time traffic data. In addition, the system may be implemented in an incident management to warn drivers in advance or it can prevent the traffic crash or secondary traffic crash. The model adjusted itself by incorporating additional data, which means that the determined models based on each criterion were modified with added data through induction procedure.

REFERENCES

Abdel-Aty, M., Pande, A. (2005). "Identifying crash propensity using traffic speed conditions.", *Journal of Safety Research*, 36 (1): pp. 97–108.

Abdelwahab, H.T., Abdel-Aty, M.A. (2001). "Development of Artificial Neural Network Models to Predict Driver Injury Severity in Traffic Accidents at Signalizes Intersection.", *Transportation Research Record,* 1746, Paper No. 01-2234, pp. 6–13.

Aghayan, I., Noii, N. and Kunt, M. (2013). "Fuzzy C-means clustering based on clustering algorithms for traffic crash data.", *Advances in Civil Engineering and Building Materials*, Edited by Zhao, J., Taylor & Francis Group, London (In press).

Aghayan, I., Noii, N. and Kunt, M. (2012). "Extended Traffic Crash Modeling through Precision and Response Time Using Fuzzy Clustering Algorithms Compared with Multi-Layer Perceptron.", *Promet Traffic & Transportation*, 24(6), pp. 455-467.

Akiyama, T., Sho, C. F. (1993). "Fuzzy mathematical programming for traffic safety planning on an urban expressway.", *Transportation Planning and Technology,* 17, pp. 179-190.

Audet, C., Dennis Jr, J.E. (2003). "Analysis of generalized pattern searches.", *SIAM Journal on Optimization,* 13 (3), pp. 889–903.

Bedard, M., Guyatt, G. and Stones, M. and Hirdes, J. (2002). "The independent contribution of driver, crash, and vehicle characteristics to driver fatalities.", *Accident Analysis and Prevention,* 34 (6), pp. 717–727.

Bezdek, J. (1981). "Pattern Recognition with Fuzzy Objective Function Algorithms.", New York: Plenum Press.

Cameron, M. (1997). "Accident Data Analysis to Develop Target Groups for Countermeasures.", Monash University Accident Research Centre, Reports 46 & 47.

Ceylan, H., Bell, M.G.H. (2004). "Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing.", *Transportation Research Part B: Methodological*, 38 (4), pp. 329–342.

Chang, N.B., Chen, W.C. (2000). "Prediction of PCDDs/PCDFs emissions from municipal incinerators by genetic programming and neural networking modeling.", *Waste Management and Research,* 18, pp. 341–351.

Chen JQ, Xi YG., Zhang, Z.J. (1998). "A clustering algorithm for fuzzy model identification." *Fuzzy Sets System*, 98, pp. 319–29.

Chiu, S.L. (1994). "Fuzzy model identification based on cluster estimation.", *Intelligent & Fuzzy System,* 2 (3).

De Jong, K.A. (1975). “An analysis of the behavior of a class of genetic adaptive systems.” *Dissertation Abstracts International,* vol. 36, No. 10, 5140B (University Microfilms No. 76-9381).

Delen, D., Sharda, R. and Bessonov, M. (2006). “Identifying significant predictors of injury severity in traffic accidents using a series of artificial neural networks.”, *Accident Analysis and Prevention*, 38, pp. 434-444.

Deschaine, L.M., Francone, F.D. (2004). “Comparison of Discipulus (Linear Genetic Programming software with Support Vector Machines, Classification Trees, Neural Networks and Human Experts).”, from http://www.rmltech.com/ Comparison.White.Paper.pdf ,accessed 7.02.08.

Donnell, E.T., Mason Jr, J.M. (2006). “Predicting the frequency of median barrier crashes on Pennsylvania.”, *Interstate Highways. Accident Analysis & Prevention*, 38 (3), pp. 590–599.

Dunn, J.C. (1974). “A fuzzy relative of the ISODATA process and its use in detecting compact, well-separated clusters. J. Cybernet.”, 3, pp. 32–57.

Frigui, H., Nasraoui, O. (2004). “Unsupervised learning of prototypes and attribute weights. Pattern Recognition.”, 37, pp. 567–581.

Fukunaga, K. (1990). “Introduction to Statistical Pattern Recognition.”, San Diego: Academic Press.

Guha, S., Rastogi, R. and Shim, K. CURE. (1998). “An efficient clustering algorithm for large databases.”, In Proc. of 1998 ACM-SIGMOD Int. Conf. on Management of Data, pp. 73–84.

Guha, S., Rastogi, R. and Shim, K. ROCK. (1995). “A robust clustering algorithm for categorical attributes.”, In Proc. of the 15th Int’l Conf. on Data Eng., pp. 512–521.

Hadji Hosseinlou, M., Aghayan, I. (2009). “Prediction of traffic accident severity based on fuzzy logic.”, *8th International Congress on Civil Engineering, Shiraz, Iran. 65p.*

Hagan, M.T., Demuth, H.B. and Beale, M.H. (1996). “Neural Network Design.” Boston, MA: PWS Publishing.

Hammouda, K., Fakhreddine, K. (2002). “A Comparative study of data clustering techniques.”, SYDE 625: Tools of Intelligent Systems Design. Course Project. University of Waterloo, Ontario, Canada.

Hayajneh, M., Hassan, A. (2008). “Modelling the machinability of self-lubricated aluminium/alumina/graphite hybrid composites synthesized by the powder metallurgy method using a fuzzysubtractive clustering-based system identification method.”, *International Journal of Machining and Machinability of Materials* , 3(3-4), pp. 252–271.

Hensher, D.A., Ton, T.T. (2000). “A comparison of the predictive potential of artificial neural networks and nested logit models for commuter mode choice.”, *Transportation Research Part E,* 36 (3), pp. 155–172.

Holland, J.H. (1975). “Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor, USA.

Huang, H., Abdel-Aty, M. (2010). “Multilevel data and Bayesian analysis in traffic safety.”, *Accident Analysis & Prevention*, Doi:10.1016/j.aap.03.013.

Ishak, S.S., Al-Deek, H.M. (1998). “Fuzzy art neural network model for automated detection of freeway incidents.”, *Transportation Research Record* 1634, pp. 56–63.

Ivan, J.N., Wang, C. and Bernardo, N.R. (2000). “ Explaining two lane highway crash rates using land use and hourly exposure”, *Accident Analysis and Prevention*, 32, (6), pp. 787-795.

Jain, A.K., Murty, M.N. and Flynn, P.J. (1999). “Data clustering: a review”, ACM Computing Surveys, 31 (3), pp. 264–323.

Jiang, D., Tang, C. and Zhang, A. (2004) “Cluster analysis for gene express data: a survey”, IEEE Transactions on Knowledge and Data Engineering, 16 (11), pp. 1370–1386.

Jin, X., Cheu, R.L. and Dipti, S. (2002). “Development and adaptation of constructive probabilistic neural network in freeway incident detection.”, *Transportation Research Part C* 10 (2), pp. 121–147.

Kamijo, S., Matsushita, Y., Ikeuchi, K. and Sakauchi, M. (2000). “Traffic monitoring and accident detection at intersections.”, *IEEE Transactions on Intelligent Transportation Systems*, 1 (2), pp. 108–118.

Karlaftis, M.G., Golias, I. (2002). “Effects of road geometry and traffic volumes on rural roadway accident rates.”, *Accident Analysis and Prevention*, 34 (3), pp. 357–365.

Karypis, G., Han, E.H. and Kumar, V. Chameleon. (1999). “A hierarchical clustering algorithm using dynamic modeling.”, *IEEE Computer*, 32(8), pp. 68–75.

Khan, S., Ahmad, A. (2004). “Cluster centre initialization algorithm for K-means clustering.”, *Pattern Recognition Letters*, 25 (11), pp. 1293–1302.

King, B. (1967). “Step-wise clustering procedures.”, *Journal of the American Statistical Association,* 69, pp. 86–101.

Kononenko., Kukar. M. (2007). “Machine learning and data mining.”, Chichester, UK: Horwood Publishing.

Kunt, M.M., Aghayan, I. and Noii, N. (2011). "Prediction for traffic accident severity: comparing the artificial neural network, genetic algorithm, combined genetic algorithm and pattern search methods." *Transport*, 26(4), pp. 353-366.

Lanser, S.H., Hoogendoorn, S. A. (2000). "Fuzzy genetic approach to travel choice behavior in public transport networks.", *Transportation Research Board 79th Annual Meeting, TRB, Washington, DC.*

Lewis, R.M., Torczon, V. (1999). "Pattern search algorithms for bound constrained minimization.", *SIAM Journal on Optimization,* 9 (4), pp. 1082–1099.

Lewis, R.M., Torczon, V. (2002). "A Globally Convergent Augmented Lagrangian Pattern Search Algorithm for Optimization with General Constraints and Simple Bounds.", *SIAM Journal on Optimization*, Vol 12, 4, pp. 1075–1089.

Lord, D., Manar, A. and Vizioli, A. (2005). "Modeling crash-flow-density and crash-flow V/C ratio relationships for rural and urban freeway segments.", *Accident Analysis and Prevention,* 37, (1), pp. 185-199.

Lord, D. (2008). "Methodology for estimating the variance and confidence intervals of the estimate of the product of baseline models and AMFs.", *Accident Analysis & Prevention,* 40 (3), pp. 1013–1017.

MacQueen, J.B. (1967). "Some methods for classification and analysis of multivariate observations. Proceedings of Fifth Berkeley Symposium on

Mathematical Statistics and Probability.", (1), pp. 281–297, Berkeley: University of California Press.

Mahal, D., Hakkert, A. S. and Phrasker, J. N. (1982). "A system for the allocation of safety resources on a road network.", *Accident Analysis and Prevention*, 14 (1), pp. 45-56.

Mamdani, E., Assilian, S. (1975). "An experiment in linguistic synthesis using a fuzzy logic controller.", Int J Man Mach Stud, 7(1), pp. 1–13.

Mussa, R.N., Upchurch, J.E. (2002). "Simulator evaluation of incident detection by transponder-equipped vehicles.", *Transportation,* 29 (3), pp. 287–305.

Mussone, L., Rinelli, S. and Reitani, G. (1996). "Estimating the accident probability of a vehicular flow by means of an artificial neural network.", *Environment and Planning B*, 23, p. 667–675.

Mussone, L., Ferrari, A. and Oneta, M. (1999). "An analysis of urban collisions using an artificial intelligence model.", *Accident Analysis and Prevention*, 31 (6), pp. 705–718.

Ng, K-S., Hung, W-T and Wong W-G (2002). "An algorithm for assessing the risk of traffic accidents.", *Journal of Safety Research*, 33, pp. 387-410.

Ng, J. C. W., Sayed, T. (2004). “Effect of geometric design consistency on road safety.”, *Canadian. Journal of Civil Eng*, 31 (2), pp. 218– 227.

Niitymaki, J. (2001). “General fuzzy rule base for isolated traffic signal control: rule formulation.”, *Transporation Planning and Technology*, 24 (3), pp. 237–247.

Park, B., Messer, C.J. and Urbanik II, T. (2000). “Enhanced genetic algorithm for signaltiming optimization of oversaturated intersections.”, *Transportation Research Record*, 1727, pp. 32–41.

Pena, J.M., Lozano, J.A. and Larranaga, P. (1999). “An empirical comparison of four initialization methods for the K-means algorithm.”, *Pattern Recognition Letters,* 20, pp. 1027–1040.

Polat, K., Durduran, SS. (2012). “Automatic determination of traffic accidents based on KMC based attribute weighting.”, *Neural Computing & Applications*, 21, pp. 1271–1279.

Redmond, S.J., Heneghan, C. (2007) “A method for initializing the K-means clustering algorithm using kd-trees.”, *Pattern Recognition Letters,* 28, pp. 965–973.

Ruspini, E. H. (1969). “A new approach to clustering, Information and Control.”, 15(1), pp. 22-32.

Shie, Jen-Da., Chen, Shyi-Ming. (2008). “Feature subset selection based on fuzzy entropy measures for handling classification problems”, *Application Intelligent Journal*, Vol 28, pp. 69-82.

Sneath, P.H., Sokal, R.R. (1973). “Numerical Taxonomy.”, London, UK:Freeman

Subba Rao, P.V., Sikdar, P.K. and Krishna Rao, K.V. (1998). “Another insight into artificial neural networks through behavioral analysis of access mode choice.”, *Computers, Environment and Urban Systems*, 22 (5), pp. 485–496.

Sohn, S., Lee, S. (2003). “Data fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic accident in Korea.” *Safety Science*, 41 (1), pp. 1–14.

Sugeno, M., Yasukawa, T. (1993). “A fuzzy-logic-based approach to qualitative modeling.”, *IEEE Trans Fuzzy System*, 1, pp. 7–31.

Teklu, F., Sumalee, A. and Watling, D. (2007). “A genetic algorithm approach for optimizing traffic control signals considering routing.”, *Journal of Computer Aided Civil and Infrastructure Engineering,* 22 (1), pp. 31–43.

Tong, H.Y., Hung, W.T. (2002). “Neural network modeling of vehicle discharge headway at signalized intersection: model descriptions and the results.”, *Transportation Research Part A*, 36 (1), pp. 17–40.

Torczon, V. (1997). “On the convergence of pattern search algorithms.”, *SIAM Journal on Optimization*, 7 (1), pp. 1–25.

Valent, F., Schiava, F., Savonitto, C., Gallo, T., Brusaferro, S. and Barbone, F. (2002). “Risk factors for fatal road accidents in Udine, Italy.”, *Accident Analysis and Prevention*, 34 (1), pp. 71–84.

Vythoulkas, P.C., Koutsopoulos, H.N. (2003). “Modeling discrete choice behavior using concepts from fuzzy set theory, approximate reasoning and neural networks.”, *Transportation Research Part C*, 11 (1), pp. 51–73.

Wang, X., Wang, Y. and Wang, L. (2004). “Improving fuzzy C-means clustering based on feature-weight learning.”, *Pattern Recognition Letters*, 25, pp. 1123–1132.

Washington, S., Karlaftis, M. and Mannering, F. (2003). “Statistical and Econometric Methods for Transportation Data Analysis.”, Chapman and Hall/CRC, Boca Raton, FL.

WHO (2004). “World Report on Road Trafic Injury Prevention.”, M. Peden, R. Scurfield, D.Sleet, D. Mohan, A. Hyder, E. Jarawan and C. Mathers (eds.). World Health Organization, Geneva, Switzerland.

Wright, C. C., Abbess, C. R. and Jarrett, D. F. (1988). “Estimating the regression to mean effect associated with road accident black spot treatment: Towards a more realistic approach.”, *Accident Analysis and Prevention*, 20 (3), pp. 199-214.

Yager, R.R., Filev, D.P. (1994). “Approximate clustering via the mountain method.”, *IEEE Transactions On Systems*, Man, and Cybernetics, 24 (8), pp. 1279–1284.

Yeung, D. S., Cloete, I., Shi, D. and W.Y. Ng, W. (2010). “Sensitivity Analysis for Neural Networks.”, Springer- Verlag Berlin Heidelberg.

Yin, H., Wong, S.C., Xu, J. and Wong, C.K. (2002). “Urban traffic flow prediction using fuzzy-neural network.”, *Transportation Research Part C*, 10 (2), pp. 85–98.

Yuan, F., Cheu, R.L. (2003). “Incident detection using support vector machines.”, *Transportation Research Part C*, 11 (3-4), pp. 309–328.

Zadeh, L. (1965). “Fuzzy sets. Information and Control.”, 8, pp. 338-353.

Zhang, J., Lindsay, J., Clarke, K., Robbins, G. and Mao, Y. (2000). “Factors affecting the severity of motor vehicle traffic crashes involving elderly drivers in Ontario.”, *Accident Analysis and Prevention*, 32 (1), pp. 117–125.

Zhang, H.M., Ritchie, S.G. and Jayakrishnan, R. (2001). “Coordinated traffic-responsive ramp control via nonlinear state feedback.”, *Transportation Research Part C*, 9 (5), pp. 337–352.

Zhong, M., Lingras, P. and Sharma, S. (2004). “Estimation of missing traffic counts using factor, genetic, neural and regression techniques.”, *Transportation Research Part C*, 12 (2), pp. 139–166.