UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Muhammad Uzair

Soft Computing Techniques For Forest Fires Prediction

Master's Thesis (30 ECTS)

Supervisor: Dr. Stefania Tomasiello

Tartu 2021

Soft Computing Techniques For Forest Fires Prediction

Abstract:
Forests provide fresh air and they are the source of livelihood to a huge number of people around the world. The forest fires affect the human lives in different ways such as economic, environment and health and these forest fires are usually caused by human negligence, and other environmental factors. The forest fire problem is divided into categories, i.e. fire risk, forest fire detection, and the prediction of burnt area. Developing a system to cater for solutions to this problem can help to use the resources in an efficient manner and can also save human lives and wildlife. In this thesis, some soft computing techniques are discussed, to predict the burnt area from forest fires. The considered soft computing techniques are ANFIS, ANN, SVM. In particular, a new variant of ANFIS is presented. Two publicly available datasets were used: one from Portugal and one from Canada. The techniques were applied separately on both datasets and results were collected.

Keywords:
Machine learning, Fuzzy inference systems, Fractional regularization

CERCS: P170 Computer science, numerical analysis, systems, control

Metsatulekahjude ennustamise pehmed arvutamistehnikad

Lühikokkuvõte:
Metsad pakuvad värsket õhku ja need on elatusallikaks paljudele inimestele kogu maailmas. Metsatulekahjud mõjutavad inimelusid mitmel viisil, näiteks majanduslikus, keskkonna- ja terviseküsimustes ning need metsatulekahjud on tavaliselt põhjustatud inimlikust hooletusest ja muudest keskkonnateguritest. Metsatulekahju probleem jaguneb sisekategooriateks, st tuleoht, metsatulekahjude avastamine ja põlenud ala prognoosimine. Selle probleemi lahendamiseks mõeldud süsteemi väljatöötamine võib aidata kasutada ressursse tõhusal viisil ning päästa ka inimelusid ja elusloodust. Selles lõputöös käsitletakse mõningaid pehmeid arvutustehnikaid, et ennustada põlenud ala metsatulekahjude põhjal. Käsitletavad pehmete arvutitehnikad on ANFIS, ANN, SVM. Eelkõige esitleti ANFISe uut varianti. Kasutati kahte avalikult kättesaadavat andmekogumit: ühte Portugalist ja Kanadast. Meetodeid rakendati eraldi nii andmekogumites kui ka kogutud tulemustes.

Võtmesõnad:
Masinõpe, Hägusad järeldamissüsteemid, Murdosa seadustamine

CERCS: P170 Computer science, numerical analysis, systems, control

Acknowledgements
A sincere thanks to my supervisor for her support during the preparation of this thesis and Mr. Mati Valgepea, leading specialist in forest statistics, Estonian Environment Agency (ESTEA), for the useful information about forest fires in Estonia.

List of abbreviations and terms:

ANN	Artificial Neural Network
ANFIS	Adaptive Neuro-Fuzzy Inference System
CCN	Cascade Correlation Network
MPNN	Multi-Layer Perceptron Neural Network
PNN	Polynomial Neural Network
RBF	Radial Basis Function
SVM	Support Vector Machine
SVR	Support Vector Regressor
DENFIS	Dynamic Evolving Fuzzy Inference System
FPO	Flower Pollination Optimization
PSO	Particle Swarm Optimization
ESTEA	Estonian Environment Agency
RMSE	Root Mean Squared Error
MSE	Mean Squared Error
MAE	Mean Absolute Error
GFD	Greek forest department
WSN	Wireless Sensor Networks

List of Tables

List of Figures

Contents

1 Introduction

1.1 Problem Overview

Forests are the source of air we breathe and wood we use, and they are source of livelihood to around 300 million people including 60 million native people [10]. Forest fire has a huge impact on the environment and it negatively affects different aspects of life such as economics, natural environment and health [7]. The forest fire can be caused by human negligence, lightnings, environmental factors [15, 25]. Every year, millions of forest hectares (*ha*) are destroyed all around the world. For instance, Portugal is a highly affected country [12]: in 2007, there were 64 deaths and a loss of 5 billion euros [18]. In China, from 1952 to 2012, there were 798,500 forest fires and 38,060,000 hectares of forest destroyed and a loss $22 million a year [20]. Planet Earth lost 129 million hectares of forest area from 1990 to 2015 [10].

The forest fire problem can be categorized into different classes: forest fire detection [28, 10], fire risk, burnt area prediction [39, 10]. There have been many systems developed for the prediction and the early detection of forest fires. Soft Computing (SC) techniques including CCN, MPNN, PNN, RBF, and SVM were used in [7] to predict the burned area from forest fire. SVM, Random forests, decision tree, ANNs and multiple regressions were used in [12]. SVM and ANNs were used in [18] to detect forest fire in Greek Forest Departments. Fuzzy systems with wireless sensors network to predict forest fire was employed in [20]. Different ensemble learning approaches were discussed in [39] to forecast forest fire. ANNs have been used in [30] for forest fire prediction.

While the goal of fire risk algorithms is to prevent ignition, the prediction of the burnt area during forest fires is aimed at supporting firefighting strategies (e.g. to properly allocate firefighting resources).

In this thesis the focus is on burnt area prediction. This is a kind of regression problem.

The regression is a special form of function approximation. Let $\mathbf{v} = (v_1, \ldots, v_n)$ be a vector of independent variables and z a response variable, the following relationship is considered:

$$z = f(v_1, \ldots, v_n) + \epsilon, \tag{1}$$

where ϵ is an error term to account for uncertainties. Given a data set $\mathbf{D}$, with N instances, the regression problem consists of finding an estimation of f giving the minimum error.

It is worth mentioning that in Estonia the main reason of forest fires is the improper human action. Data on forest resources and fires in Estonia are published annually in the statistical yearbook "Forest" (e.g. see [1]). Unfortunately, information such as time and exact location of forest fires are not available, both for technical and legal issues.

1.2 Motivation and Main Goals

As presented in a recent survey [5], there have been many techniques applied to the forest fires prediction problem. Among them, Neural Networks (NNs) and Support Vector Regression (SVR) are the most popular approaches in the field while the Adaptive Neuro-Fuzzy Inference System (ANFIS) and some variants of it for the prediction of the burnt area have been investigated in few papers. In [15], ANFIS and Dynamic Evolving Fuzzy Inference System (DENFIS) were numerically investigated. In [37], the authors compared the results by Takagi-Sugeno-Kang (TSK) fuzzy inference system against some kinds of functional networks. In [6], a comparative analysis was performed on the results by ANFIS, whose parameters were optimized by different evolutionary techniques, such as Flower Pollination Optimization (FPO), Genetic Algorithm (GA-ANFIS), Particle Swarm Optimization (PSO). All these variants have a certain computational cost, as the standard ANFIS. The latter is based on grid partitioning [17], which divides the input space into grid partitions, each one representing a fuzzy rule. In this method, the number of rules increases exponentially by increasing the dimension. This leads to the curse of dimensionality. Unlike the grid partitioning, the scatter partitioning divides the input space into a number of arbitrary partitions, often in the form of clusters obtained by means of clustering methods. Among them, subtractive clustering is the most popular one, in spite of the computational cost [40]. This clustering algorithm is based on the improved Mountain method of data partitioning. The resulting rule base size depends on the cluster neighborhood radius. A small value leads to a large rule base and therefore high model complexity and computational cost, while a large value leads to a small rule base that is a poor model. Besides, the standard ANFIS uses a hybrid learning algorithm with backpropagation and least-squares (LS) method. Recently, it has been proposed to use just the LS method in order to lower the computational time [27].

Anyway, the generalized inverse in LS methods could not be computed for ill-conditioned problems. In order to address such issue, there are regularization techniques. One of these methods has been recently proposed in fractional version and successfully applied to different problems [26], [23]. The aim of this thesis is to present an investigation on ANFIS with a kind of scatter partitioning and LS learning algorithm (without backpropagation) based both on standard and fractional regularization method.

2 Background

This section is divided into two parts. Subsection 2.1 describes the related works, and subsection 2.2 describes different parameters used to solve the problem.

2.1 Related Works

Different soft computing (SC) techniques were considered in [7] to predict the burned area from a fire. These SC techniques include cascade correlation network (CCN), multi-layer perceptron neural network (MPNN), polynomial neural network (PNN), radial basis function (RBF) and support vector machine (SVM). The performance of these models was evaluated using five quality measures including root mean squared error (RMSE), mean squared error (MSE), relative absolute error (RAE), mean absolute error (MAE), and information gain (IG). SVM exhibited the best performance by outperforming RBF, MPNN, PNN, and CCN models.

Different Data mining (DM) techniques were applied to measure the burned area of small fires [12]: multiple regression (MR), decision tree (DT), random forests (RF), neural network (NN), and support vector machine (SVM). There was some preprocessing done for MR, NN and SVM models. The nominal variables were transformed into a *1-of-C* encoding. SVM outperformed the other DM techniques used in the study.

In [18], Greek wild fires from 1983-1997 were discussed to forecast the annual burned area. The regression ability of SC approaches was discussed in this study. There were models developed using Artificial Neural Network (ANN) and e-Regression Support Vector Machine (SVM) which were local to a distinct Greek forest department (GFD). There were 51 GFD in total. ANN models completely failed in two forest departments while SVM models worked on all of 51 GFDs. Root mean squared error (RMSE) was used for the evaluation. The ANN showed low R^2 score for only 2 out of 49 cases and SVM showed low R^2 score for 13 out 51 cases, thus, outperforming the ANN models.

Fuzzy inference system and rechargeable wireless sensor networks have shown promising results in forest fire prediction [20]. These sensors collect a huge amount of data for weather information on daily basis which can support a very accurate status of forest environment. The study [20] was conducted in Nanjing City, China. There can be many factors in forest fire and these factors were divided into three categories which include weather factors, human behavioral factors and environmental factors which are input to the fuzzy system. Triangular membership functions were chosen in the fuzzy rule system in this study. The output of the fire probability were: low, moderate, high, very high, extreme.

Ensemble learning approaches can be helpful in forest fire forecasting and prevention and it also can help with forest fire management by providing forest firefighting decision making [39]. Ensemble learning consists of multiple learners which work together to

solve a problem and eventually to improve the results of the problem. Random Forests (RFs) and Extreme Gradient Boosting (EGB) are some of the most popular ensemble learning methods and they were adopted in [39]. Two experiments were conducted, one to predict the burned area, and another one to predict large-scale fires. In the first experiment, there was a comparison made between default random forests (RFd), tuned random forests (RFt), default gradient boosting machines (GMBd), tuned gradient boosting machines (GMBt), default generalized linear models (GLMd), tuned generalized linear models (GLMt) and deep learning (DL). The RFt outperformed other models on the basis of two evaluation metrics, i.e. root mean squared error (RMSE) and mean absolute error (MAE). The second experiment dealt with logit regression (LR), decision tree (DT), random forests (RF), support vector machine (SVM), neural network (NN), extreme gradient boosting (EGB), and deep learning (DL). EGB achieved the highest prediction accuracy among all other predictors at 72%.

Artificial Neural Networks (ANNs) were also used for the prediction of forest fire [30]. The ANN is a mathematical model that can be easily implemented as a software simulation that tries to simulate two essential properties of the human brain in relation with its high capabilities of parallel information processing. The first property concerns our ability to learn from examples, and the second one our ability to generalize the knowledge we acquire through the learning process to new and unseen examples [30]. The burnt area was predicted using the multi layer perceptron (MLP) architecture. Different number of hidden layers were introduced and the error rate was calculated for each scheme of hidden layers. The number of hidden layers was determined heuristically. Backpropagation algorithm was used to converge to a local minimum of the global error observed. The results of this study were satisfactory and can be tuned with the size of the hidden layers.

A comparative study of Data Mining (DM) models is presented in [10] which includes Cascade Correlation (CC), Decision Tree Forest (DTF), Gene Expression Programming (GEP), Group Method of Data Handling Polynomial Network (GMDH-PN), Linear Regression (LR), Multilayer Perceptron Neural Network (MLP), Radial Basis Function Neural Network (RBF), Epsilon Support Vector Regressor (e-SVR) and Nu Support Vector Regressor (Nu-SVR). The evaluation metrics includes correlation coefficient (R), Normalized Mean Squared Error (NMSE), Maximum Error (ME), Root Mean Squared Error (RMSE), Mean Squared Error (MSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). GEP outperformed other models in 4 out of 7 chosen accuracy metrics. Two SVR models (e-SVR) and (NuSVR) were close in giving good predictions but they still got beaten by GEP. The study suggests that GEP models is suitable to predict the burned area using different features.

Different machine learning algorithms have been used to assess the forest-fire susceptibility (FFS) in Fars province, Iran [25]. The machine learning algorithms used are boosted regression tree (BRT), general linear model (GLM), and mixture discriminant

analysis (MDA). The dataset used contains historical satellite images of forest fires of 358 different locations. The evaluation metrics used are ROC curve, accuracy, overall accuracy, True-skill statistic (TSS), F-measures, corrected classify instances (CCI), and K-fold cross-validation (4-fold). The results show that BRT and MDA are more effective than GLM for FFS mapping.

The images collected from forest fires usually have complex backgrounds which can lead to false or missing alarm due to the weak generalization of the image recognition [21]. Deep learning based multi-level forest fire detection method has been used to solve the aforementioned problem. In particular, the following steps were followed:

- General Advanced Networks (GAN) was used to generate more samples in case of less samples,
- Adaboost classifier based on Histogram of Oriented Gradients (HOG) was used to make primary predictions,
- Convolutional neural networks (CNN) and Support Vector Machine (SVM) was used to make the final prediction.

The GAN was used to create high quality forest fire images to solve the generalization problem. In total 500 forest fire and 1500 non-fire images were created using GAN. The proposed method achieved higher recognition rate and lower false alarm rate. The recognition rate of forest fire is up to 97.6%, missed alarm rate is 1%, and false alarm rate is 1.4%.

Fuzzy c-means is a well-established clustering algorithm and an extension of fuzzy c-means, by utilizing a flexible termination criterion, was proposed in [19]. The new system was named FUCEMFUTEC (fuzzy c-means with fuzzy termination criterion). In fuzzy c-means each data point belongs to only one specific cluster, while the proposed approach puts each data point into all of the determined clusters with a degree of membership. The approach was applied to the Greek forest departments according to their fire risk. The risks include number of forest fires and annual burned forested areas. There were two clusters, i.e. "Very Risky" and "Risky" and each forest department was assigned a degree of membership to each cluster. The results show that the estimation of forest fire risk is 35-40%, which does not seem very good, but can support the stakeholders decision making.

Long-term planning of high or low fire risk depends on long-term prediction [16]. The existing approaches for long-term forest fire prediction have a problem of applying specific cluster boundaries. Triangular and trapezoidal membership functions can be used to predict long-term forest fire risk. The developed decision system uses various aspects of fuzzy sets and fuzzy machine learning. This study has been applied to Greek Forest Department and the results shows an accuracy of 60% in prediction of ten of the most risky areas.

A competitive spatial prediction model has been proposed in [22] to detect early forest fire using geodata. This model can predict forest fires, identify risk zones, provide real time alerts to authorities and present maps based on geographical treatments. Machine learning algorithms were used to detect the early forest fire, with the support of the geographical information system (GIS). The machine learning algorithms include Random Forest (RF), Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). RF achieved an overall accuracy of 100% while SVM and KNN achieved 35%, 45% for small fire and 29%, 45% for large fire. The overall accuracy for RF is 100%, SVM 74%, and KNN provides 58%.

Topographical and meteorological data of Vietnam was used in [35] for forest fire prediction; the approach was the Extreme Learning Machine (ELM). This approach is based on a single hidden neural network, overcoming the problems caused by the gradient descent in the backpropagation and resulting in a shorter training time. ELM is designed to work with single hidden layer feed forward (SLFN) networks. These SLFNs include Support Vector Machine (SVM), polynomial network, Radial Basis Function (RBF), and single and multi-layer neural networks. The dataset was divided into three partitions for experiment. The training-testing data were 648-432 for the first partition, 540-540 for the second one, 756-324 for the third one. There were 5 activation functions used and SinC function achieved 85.42% accuracy with 540-540 scheme, radial basis function achieved 84.95%, and hardlim achieved 77.08%.

A novel approach for fire prediction combining Internet of Things (IoT) and machine learning is proposed in [29]. The machine learning algorithms used include Boosted Decision Trees, Decision Forest Classifier, Decision Jungle Classifier, Averaged Perceptron, 2-Class Bayes Point Machine, Local Deep Support Vector Machine (SVM), Logistic Regression and Binary Neural Network. A camera was deployed on the field to capture the real time images of the forest area and those images were compared, on the basis of the training dataset, to classify fire or no fire. The IoT sensors collect data for temperature, relative humidity and Carbon mono-oxide (CO). This hybrid model eliminates false alarms because if one system fails, the other will switch to hot-standby with weight 1. Boosted decision tree model achieved the highest Area Under Curve (AUV) of 78% and proved to be the most suitable candidate for forest fire prediction.

In [8], the authors used ELM to predict the burnt area (from the Portuguese dataset considered in this thesis), by using 10-fold cross validation and trying a different number of neurons in the hidden layers, i.e. 10, 20, 40, 60, 80. They compared their approach with Linear Regression, Random Forest Regression, and Support Vector Regression (SVR), the latter with linear, polynomial, RBF, and sigmoid kernels. They found the best performance by ELM with linear activation function.

In [37], experiments on the same dataset were performed by using Chebyshev polynomial neural network, Hermite polynomial neural network, Radial Basis Function neural network, Takagi-Sugeno-Kang (TSK) fuzzy inference system, by finding almost the

same results.

A comprehensive survey of the machine learning algorithms based forest fires prediction and detection systems can be found in [5].

2.2 Problem parameters

There are various parameters related to the forest fire such as vegetation type, density, topography, altitude, slope, relative humidity, wind speed and air temperature, season, date, time [7, 12, 18, 20, 25]. Other parameters may include geographic and temporal information [12]. In this section several parameters used in different studies are presented.

The features used to predict the burned area in [7, 12, 39, 30, 10] are given in Table 1 in Section 4. There are the geographic features (X and Y) which show the x and y axis location of the fire. Month and day are the temporal variables and indicate when the fire occurred. Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Drought Code (DC) are the Fuel Moisture Codes [12]. The Initial Spread Index (ISI) belongs to the Fire Behaviour Indexes. The **area** is the burned area which is predicted using the other features.

There are nine independent parameters considered in [18] to determine the total burned area in Greek Forests and those parameters include: *average altitude* at the spatial and temporal point of the forest fire breakout (STPFFB), *average relative humidity* STPFFB, *average air temperature* STPFFB, *average wind speed* STPFFB, *average slope* STPFFB, *average vegetation density* STPFFB, *average grassland density* STPFFB, *intervention time* (elapsed time from the moment of fire detection till the arrival of fighting forces) and *type of wild fire*. The importance of the selected features was determined by using the Rough Set Theory in the context of attribute reduction [18].

The factors affecting the forest fire were divided into three categories in [20]: weather factors, human behavioral factors and environmental factors. Weather factors include temperature, humidity, wind, rainfall. Human behavioral factors include month, date, time, history of forest fire. Environmental factors are road density, population density, fuel types, combustible concentration.

There are three categories for factors affecting the forest fire in [25]. These categories are topographical factors, climate factors, and other factors. Topographical factors include elevation, slope, aspect, topographical wetness index. Climate factors include rainfall and temperature. Land use, distance from road, distance from river, and distance from urban area belongs to the other factors.

In [35], there were 10 parameters selected to predict the forest fire. Those parameters are slope, aspect, elevation, NDVI, distance to road, distance to residential area, land use, temperature, wind speed, rainfall and forest fire occurrence.

In [29], the authors used images and different parameters like temperature, relative humidity, and carbon Mono-oxide (CO). The images were captured from the advanced cameras deployed at the field while other parameters were collected through sensors.

3 Methodology

3.1 A WSN based monitoring system

Even though it is beyond the scope of this thesis, it is worth mentioning that the algorithm for forest fires prediction can be realistically part of an architecture in which wireless sensor networks (WSNs) technology is utilised to transmit data and predict both the risk and the evolution of ignition in the forest. Both problems are influenced by meteorological parameters, such as temperature, relative humidity, wind speed, and daily precipitation, which can be measured by the sensors. Wireless Sensor Networks (WSNs) based monitoring systems have shown great potential to reduce forest fires' damaging effects. Examples of WSNs for forest fires prediction were discussed in [20, 32].

A WSN based on Internet of Things (IoT) devices and sensors can be used to perform a real-time environmental monitoring of both the fire risk and the fire detection. Their design and distribution should be done to cover as much forest areas as possible. With respect to this, there may be several challenges, such as the authentication of sensor nodes and the security of wireless communications among distributed sensor nodes, taking into account possible areas out of network coverage.

In monitoring systems designed to evaluate the fire risk, different types of environmental risk factors can be considered. Factors such as vegetation layers, topography, or the frequency of forest fires may be useful to perform a long-term estimation of forest fire risks. Other factors such as meteorological variables, polluting gases, or the oxygen level measured in real time can be used to perform a short-term estimation of forest fire risks.

The detection of forest fires involves a fast check of some parameters. In addition to meteorological variables, the oxygen level and the concentrations of polluting gases (usual indicators of fire outbreaks occurrence and biomass burning process) measured by the sensors are analysed. Unusual environmental changes such as temperature increase, decrease in relative humidity and oxygen values jointly with high concentrations of carbon dioxide and carbon monoxide may indicate the recent occurrence of a forest fire.

Fire spread can be estimated by analysing the values of meteorological variables, wind direction changes, and the oxygen level over nearby forest areas, since they influence fire propagation factors such as dryness of vegetation and organic fuels.

The monitoring platform consists typically of several modules, as described below.

- Data collection module. The measured data are collected for proper calculation. Fire detection is possible by checking some measurements (e.g. oxygen level and the concentrations of polluting gases, as mentioned before).

- Communication module. It is used to route the alarm signal generated by the data collection module to the analysis module, by meeting certain quality of service

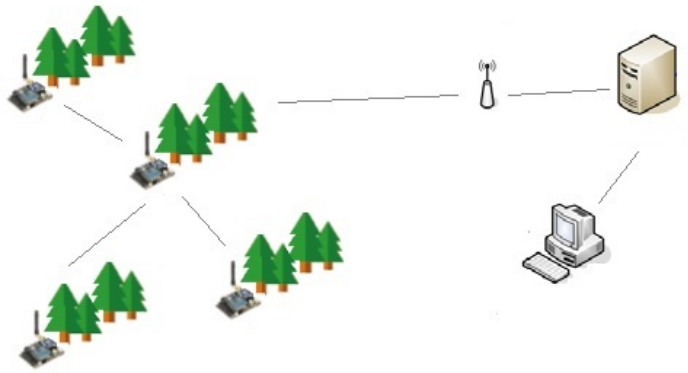

Figure 1. A WSN-based monitoring system

(QoS) requirements such as reliability (i.e. the alarm has to reach the sink safely), time constraint (i.e. the signal must arrive within a reasonable time) and security (the routing path has to be secure against any attack or malicious behavior).

- Analysis module. In this module the received alarm signal is properly checked, to avoid false alarms (e.g. by using the data collected from other sensors nodes). Then the fire spread is evaluated.

3.2 Soft Computing Approaches

3.2.1 ANFIS

Before introducing ANFIS, it is useful to recall briefly the notion of fuzzy set.

A fuzzy set is a generalization of the classical (crisp) set. While the latter is defined by a characteristic function, which can assume only the values 0 or 1, a fuzzy set is defined by a membership function, a real function which can assume any value in the interval $[0, 1]$. In Figure 2, some typical membership functions (representing continuous fuzzy sets) are depicted.

A general class of intersection operators for fuzzy sets is provided by the t-norm. It aggregates two membership grades in different ways, namely by the minimum operator, algebraic product, bounded product, drastic product.

The Adaptive Network-based Fuzzy Inference System or Adaptive Neuro-Fuzzy Inference System (ANFIS) was introduced by Jang [17]. It is a five-layer network architecture which represents the first-order Sugeno (or TSK) fuzzy inference system. The name adaptive means that the nodes in this network are adaptive, i.e. the outputs of

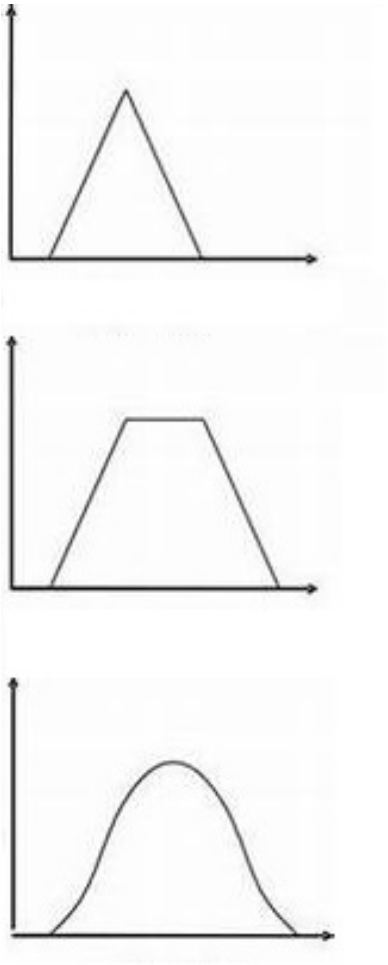

Figure 2. From the top, triangular, trapezoidal and bell-shaped membership functions

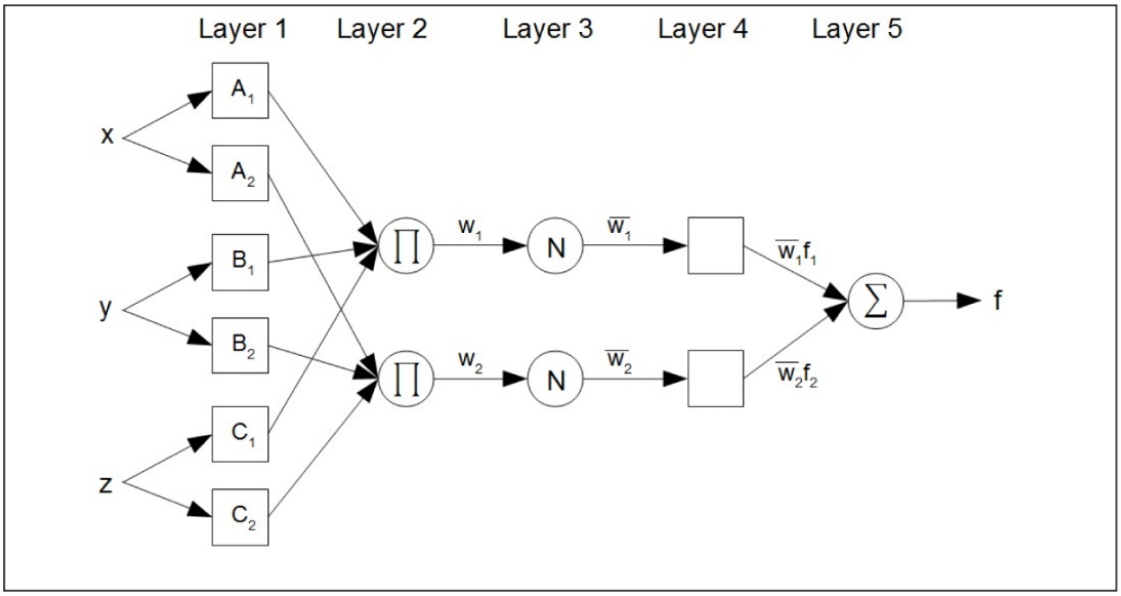

Figure 3. An ANFIS architecture (adapted from mathworks.com)

these adaptive nodes depends on the parameter(s) of that node. The common learning algorithm is based on gradient descent. To minimize the error measure, parameters should be tuned and this tuning is specified by the learning. Backpropagation with least-squares method is usually used to tune the membership function parameters for fuzzy inference system (FIS). An ANFIS architecture is represented in Figure 3.

In general, ANFIS is based on IF-THEN rules with functional consequent part. ANFIS aims at adjusting the antecedents and the consequent parameters.

There are five layers ($S_1, S_2, ..., S_5$) in the ANFIS network. All the outputs from layer S_4 are summed together to get the final output z^0. The output of S_1 is the membership degree of the input ξ to the fuzzy set B_i. Layer S_2 has all the nodes fixed and the output of each node represents the firing strength of a rule. The nodes in S_2 are computed using product-type t-norms (to model AND). For each layer S_i one has:

$S_1: \quad O_i = \mu_{B_{ir}}(\xi_i);$
$S_2: \quad \omega_r = \Pi_{i=1}^{n} \mu_{B_{ir}}(\xi_i);$
$S_3: \quad \overline{\omega}_r = \omega_r / \sum_{j=1}^{R} \omega_j;$
$S_4: \quad \rho_r = \overline{w}_r z_r = \overline{\omega}_r \alpha_{0r} + \alpha_{1r}\xi_1 + \ldots + \alpha_{nr}\xi_n;$
$S_5: \quad z^0 = \sum_{r=1}^{R} \rho_r,$

where α_{ir} are the consequent parameters. Any suitable parameterized membership function (MF) can be a membership function for B_i. One of them is the generalized bell-shaped function:

$$\mu_{B_{ir}}(\xi) = \left(1 + |\frac{\xi_i - c_{ir}}{a_{ir}}|^{2b_{ir}}\right)^{-1}, \tag{2}$$

where $\{a_i, b_i, c_i\}$ is the parameter set. The value changing parameters are referred to as premise parameters. An alternative is the classical bell-shaped function:

$$\mu_{B_{ir}}(\xi) = \exp\left(-\left(\frac{\xi_i - c_{ir}}{a_{ir}}\right)^2\right). \tag{3}$$

Backpropagation and least-squares based hybrid learning algorithm is used in the ANFIS. The output from all the nodes go forward until layer S_4, and least-squares method is used to determine the consequent parameters for fixed antecedent parameters. Matrix equation $\mathbf{M}\delta = \mathbf{Y}$ is obtained using the training data, where δ collects the unknown parameters and $\mathbf{Y}$ the target values. The least-squares (LS) method is formulated as

$$\min_{\delta} \|\mathbf{M}\delta - \mathbf{Y}\|^2 \tag{4}$$

with the solution

$$\delta* = \overline{\mathbf{M}}\mathbf{Y}, \tag{5}$$

where $\overline{\mathbf{M}} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T$ is the pseudoinverse of $\mathbf{M}$. Backpropagation is then used to adjust the antecedent parameters.

3.2.2 ANFIS with regularization

In the proposed variant, the learning algorithm is only LS-based and uses a kind of regularization, both in standard and fractional version as in [26, 23]. The standard regularization method, as a generalization of the standard LS method, is a well-known method for solving discrete ill-posed inverse problems. The minimization problem is formulated as follows:

$$\min_{\theta} \|\mathbf{M}\delta - \mathbf{Y}\|^2 + \frac{1}{C}\|\delta\|^2, \tag{6}$$

where C is a regularization parameter. The solution is:

$$\delta* = (\mathbf{M}^T\mathbf{M} + \frac{1}{C}\mathbf{I})^{-1}\mathbf{M}^T\mathbf{Y}, \tag{7}$$

where $\mathbf{I}$ is the identity matrix.

The fractional method follows the minimization problem

$$\min_{\delta} \|\mathbf{M}\delta - \mathbf{Y}\|_Q^2 + \frac{1}{C}\|\delta\|^2, \tag{8}$$

where $\|\delta\|_Q = (\delta^T \mathbf{Q} \delta)^{0.5}$ and $\mathbf{Q}$ is a symmetric positive semi-definite matrix defined as [26]

$$\mathbf{Q} = (\mathbf{M}^T \mathbf{M})^{\frac{\gamma-1}{2}}, \tag{9}$$

with $\gamma \in (0, 1)$. Let $\mathbf{u}_i$ and $\mathbf{v}_i$ be the vectors (columns of the matrices $\mathbf{U}$ and $\mathbf{V}$) coming from the singular value decomposition (SVD) of the matrix $\mathbf{M}$, that is $\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, with $\mathbf{D}$ the diagonal matrix of singular values σ_i in decreasing arrangement. Then the solution is [26]:

$$\delta* = \sum_{i=1}^{q} \frac{\sigma_i^{\gamma}}{\sigma_i^{\gamma+1} + C^{-1}} (\mathbf{u}_i^T \mathbf{Y}) \mathbf{v}_i. \tag{10}$$

3.2.3 Artificial Neural Networks

Artificial Neural Networks (ANNs) are computing schemes offering a mapping between some input and output (or target) variables. The input variables are also called attributes or features. ANNs are employed both in regression and classification problems.

3.2.3.1 Neuron Neuron (also called unit) (Fig 4) is the basic component of the ANN. An ANN consists of layers with a different number of neurons, namely, input, hidden, and output layers. There could be one or more hidden layers. They are called hidden because the computation is not visible to the user. The input χ of a generic node k is given by the linear sum of incoming signals x_i from the preceding layer, through the weights w_{ik}, plus a bias term b_k, that is

$$\chi_k = \sum_i w_{ik} x_i + b_k$$

. Notice that $x_i = f(\chi_i)$, where f is the activation function.

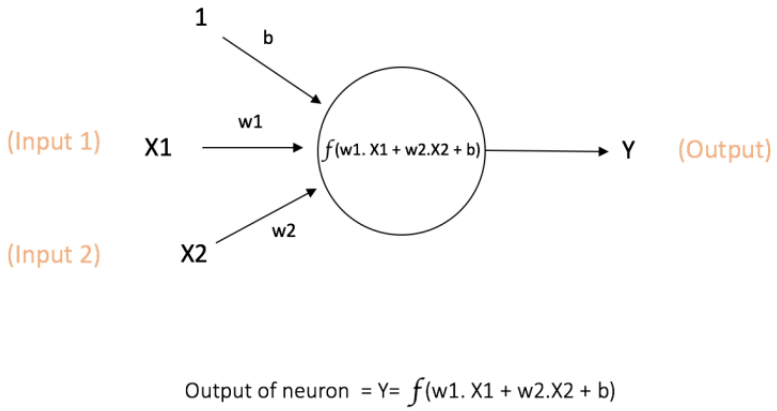

Figure 4. Neuron in Artificial Neural Network [4]

There are different activation functions. The well-known ones are:

- **Linear:** $f(\chi_i) = c\chi_i$, that is the identity function when $c = 1$; it can be used in the output node.
- **Sigmoid (logistic):** $f(\chi_i) = \frac{1}{1+\exp(-\chi_i)}$, whose range is between 0 and 1.
- **Hyperbolic Tangent:** $f(\chi_i) = \frac{2}{1+\exp(-2\chi_i)} - 1$, whose range is between -1 and 1.
- **Rectified Linear Unit (ReLU):** $f(\chi_i) = max\{0, \chi_i\}$, which has the advantage of a lower processing time.

3.2.3.2 Architecture The number of neurons (units) and their distribution in the layers, with a different number of neurons, describe the architecture of an ANN. The architecture of an ANN is usually fixed by the user. In general, there are two main types of architectures:

- Feedforward
- Feedback.

A *feedforward* network has connections only in one direction from the input to the output layer, without connections between the neurons of the same layer (Fig 5). In a *feedback* network, the output from layers can be redirected to the same layer or any other layer. In this class there are recurrent networks.

In a feedforward network, the total number of connections N_c between the nth and $(n+1)$th layer is given by:

$$N_c(n) = units(n) \times units(n+1),$$

where $units(n)$ and $units(n+1)$ are the number of units in the nth and $(n+1)th$ layer respectively. A weight matrix is also associated with the connections between the layers. For instance, $\mathbf{W}^{(n)}$ is the weight matrix referred to the connections between the nth and $(n+1)$th layer.

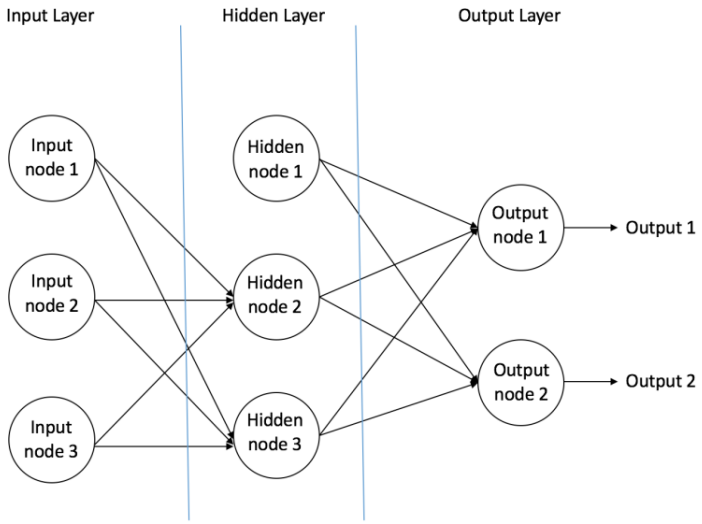

Figure 5. Feed forward network [4]

3.2.3.3 Learning Let us consider the training data set $T = \{\mathbf{v}^{(r)}, \overline{y}^{(r)}\}_{r=1}^{M}$, with $\mathbf{v}^{(k)} \in \mathbb{R}^m$. In a feedforward network, the output is obtained as follows:

- input to hidden layer
$$\chi_1 = f(\mathbf{W}^{(1)}\mathbf{v}^{(k)})$$
- hidden to hidden layer
$$\chi_{l+1} = f(\mathbf{W}^{(l)}\overline{\mathbf{v}}_l), \qquad \forall l \in \{1, \ldots, s-1\}$$
- hidden to output layer
$$\mathbf{y} = f(\mathbf{W}^{(s)}\overline{\mathbf{v}}_s)$$

The formation and modification of synapses is responsible for the learning in human brain. The learning in computing scheme is based on the optimization of a mathematical model. The learning can be supervised or not. In *supervised learning*, a training data set T consisting of M input-output data samples is presented to the ANN. The difference

between the computed output $\mathbf{y}^{(r)}$ and the desired output $\overline{\mathbf{y}}^{(r)}$ is computed to correct the weights so that the actual output could better approximate the desired output. In *unsupervised learning*, the desired outputs are not available and the learning algorithm has to detect independently patterns, features, correlations, and so on, following a process called "self-organization".

As mentioned before, the supervised learning of a neural network foresees a numerical optimization, more precisely the minimization of an error function. The latter depends on the model's parameters, that are weights and biases.

In the gradient descent algorithm, the objective function is minimized by means of gradient values with respect to the parameters. In the Levenberg-Marquardt algorithm, the second-order derivatives appears in the optimization.

Backpropagation is a typical algorithm based on the gradient descent. Weights are updated at the sth iteration (epoch) by the *delta rule*

$$w_{ij}(s) = w_{ij}(s-1) + \Delta w_{ij}(s-1), \tag{11}$$

with $\Delta w_{ij}(s-1) = -\eta \frac{\partial E(s-1)}{\partial w_{ij}(s-1)}$ where $E(s-1)$ is the total error at the epoch $s-1$ and η is the learning parameter.

The backpropagation algorithm can be stopped when the absolute rate of change of the average squared error per epoch is significantly small, e.g. in the range $[0.001, 0.01]$. Initial weights and the learning rate η influence the convergence of the backpropagation algorithm.

3.2.3.4 Learning Rate The learning rate defines how the network's weights are updated. The learning rate can affect both the time the neural network takes to learn a good solution (the number of epochs) and the result. The lower the learning rate, the smaller the changes in the weights from one iteration to the next, the higher the computational time. The higher the learning rate, the worse the stability and the convergence. It is possible to increase the rate of learning without significant instability effects by modifying the delta rule by means of a (usually positive) momentum term α

$$\Delta w_{ij}(s) = \eta \delta_j x_i(s) + \alpha \Delta w_{ij}(n-1),$$

where α is usually a positive number called the momentum constant. Its value ranges between 0.1 and 1. It is possible to allow the learning rate to vary over time, e.g. with exponential decay, or to reduce the learning rate by a certain factor every few epochs.

Backpropagation does not represent the best learning algorithm in general, but it is computationally efficient, because its computational complexity is linear in the total number of weights. For large-scale learning, probably the most popular method is the stochastic gradient descent (SGD). It updates the model parameters by means of a random selection of the training samples.

3.2.4 Support Vector Machines

Support vector machines (SVMs) is a supervised learning algorithm and was proposed by Vapnik in 1992 [9]. SVMs can be used for classification and regression problems. Explicit use of convex optimization, statistical learning theory, and kernel functions are the key innovations of SVM.

3.2.4.1 Classification Given a training set $S = \{(\mathbf{x_1}, \mathbf{y_1}), \ldots, (\mathbf{x_n}, \mathbf{y_n})\}$ of data points x_i from $X \subseteq \mathrm{R}^n$ with corresponding labels y_i from $Y = \{-1, +1\}$, generated from an unknown distribution, the task of the classification is to learn a function $\mathbf{g}$: $X \rightarrow Y$ that correctly classifies new examples (x, y) (i.e. such that $g((x) = y)$ generated from the same underlying distribution as the training set. Generalization means smallest error on unseen/test data, so a good classifier should generate the best possible generalization. The expected generalization error can be calculated using statistical learning theory [38] for a classifier which is known as the capacity of the classifier. The capacity is regulated by SVM in a way that it separates the given training data using the hyperplane according to the labels. A higher dimensional feature space is realized when there is no linear separation is possible. The hyperplane corresponds to the non-linear decision boundary in the input space.

Let us consider $\phi : I \subseteq \mathrm{R}^n \rightarrow F \subseteq \mathrm{R}^n$ a mapping from the input space I to feature space F (Fic 6). A hyperplane is found during the learning phase which has the highest margin. The equation of such a hyperplane is

$$\gamma = \min_{1 \leq i \leq \ell} \quad \mathbf{y_i}(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle - b) = \min_{1 \leq i \leq \ell} \quad \mathbf{y_i g}(\mathbf{x}_i) \tag{12}$$

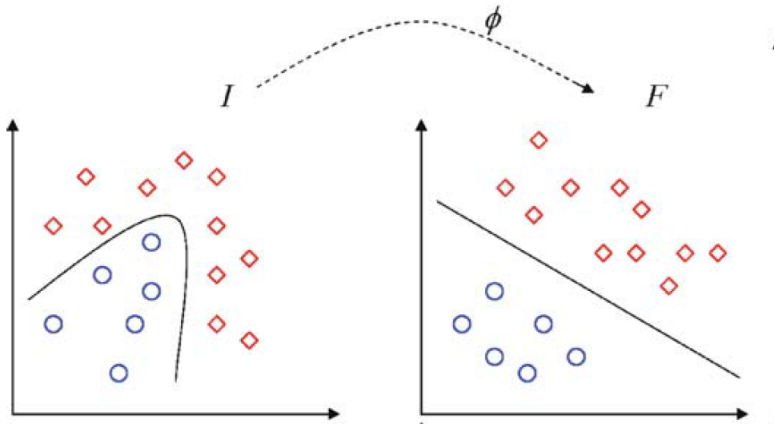

Figure 6. Mapping of input space I to feature space F [13]

where $\mathbf{w}$ is an ℓ dimensional vector of weights, b is the threshold, and $\langle , \rangle$ is the inner product.

The value $(\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle - b)/\|\mathbf{w}\|$ describes the distance of a sample $\mathbf{x}_i$ form the hyperplane. When this value is multiplied to a label $\mathbf{y}_i$, we get a positive or negative value based on the correct or incorrect classification, respectively. A label is assigned to a given data point $\mathbf{x}$ by the following function:

$$\mathbf{g}(\mathbf{x}) = sign(\langle \mathbf{w}, \phi(\mathbf{x})\rangle - b) \tag{13}$$

3.2.4.2 Maximizing the Margin When classes are linearly separable, a hyperplane $(\mathbf{w}, b)$ exists such that:

$$\mathbf{y_i}(\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle - b) \geq \gamma \quad i = 1, \ldots, \ell \tag{14}$$

Imposing $\|\mathbf{w}\|^2$ = 1, the choice of the hyperplane such that the margin is maximized is equivalent to the following optimization problem:

$$\max_{w,b,\gamma} \gamma \quad \text{subject to} \quad \mathbf{y_i}(\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle - b) \geq \gamma \quad i = 1, \ldots, \ell \quad \text{and} \quad \|\mathbf{w}\|^2 = 1 \tag{15}$$

An efficient solution can be found in the dual space by introducing the Lagrange multipliers $\alpha_\mathbf{i}, \mathbf{i} = \mathbf{1}, \ldots, \ell$. The problem (15) can be recast in the following dual form:

$$\max_{\alpha} \sum_{i=1}^{\ell} \alpha_i \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \langle \phi(\mathbf{x_i}), \phi(\mathbf{x_j})\rangle \quad \text{subject to} \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0, \alpha_i \geq 0 \tag{16}$$

This reduces the problem to a quadratic optimization task. The solutions α^* of this kind of problems must satisfy the Karush-Kuhn-Tucker conditions. This ensures that only a subset of training examples needs to be associated to a non-zero α_i. This is called the *sparness* of the SVM solution. In the solution α^*, only a subset of training examples is associated to non-zero α_i. These lie closer to the separating hyperplane and called *support vectors* (Fig 7). The weight vector $\mathbf{w}^*$ for the maximal margin hyperplane is given by linear function of the training points:

$$\mathbf{w}^* = \sum_{\mathbf{i=1}}^{\ell} \alpha_\mathbf{i}^* \mathbf{y_i} \phi(\mathbf{x_i}) \tag{17}$$

Then the decision function (13) can be expressed as:

$$g(\mathbf{x}) = sign(\sum_{i=1}^{\ell} \alpha_i^* y_i \langle \phi(\mathbf{x_i}), \phi(\mathbf{x})\rangle - b) \tag{18}$$

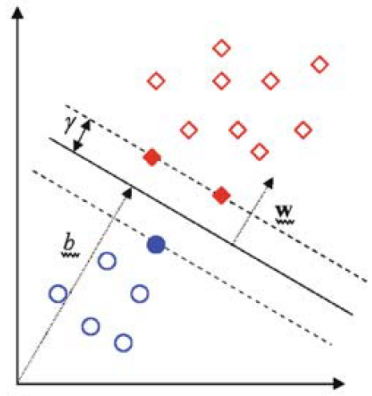

Figure 7. Support Vectors on a hyperplane [13]

For a support vector $\mathbf{x}_i$, it is $\langle \mathbf{w}^*, \phi(\mathbf{x_i}) \rangle - b = \mathrm{y}_i$ from which the optimum bias b^* can be computed. It is usually better to average the values obtained by considering all the support vectors [13]. Quadratic programming (QP) problem (16) and decision function (18) depend on the dot product between data points. The matrix of dot products with elements $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i, \phi(\mathbf{x}_j) \rangle$ is called the *kernel matrix*. In the case of linear separation $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$, but in general, functions that provides non-linear boundaries can also be used. Widely used kernels are the polynomial $K(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + 1)^d$ or the Gaussian $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\left\| x_i - x_j \right\|^2}{\sigma 2}}$ where d and σ are user-defined parameters.

3.2.4.3 Soft Margin Too much noise in the training data can lead the SVM to over-fitting. Some training errors needs to be tolerated in order to get the better generalization. This is called the *soft margin classifier* [11]. Optimal class separation can be obtained by introducing the slack variables $\xi_i \geq 0$:

$$\min_{\mathbf{w},b,\gamma,\xi} -\gamma + C \sum_{i=1}^{\ell} \xi_i \quad \text{subject to} \quad y_i(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle - b) \geq \gamma - \xi_i, \xi_i \geq 0 \quad i = 1, \ldots, \ell \quad \text{and} \quad \|\mathbf{w}\|^2 = 1 \tag{19}$$

The classification errors and the maximization of the margin can be controlled with the user-defined constant C.

3.2.4.4 Regression Support Vector regression (SVR) was proposed in 1996 [14]. A linear algorithm is used in the kernel-induced feature space to construct a function such that the training points are inside a tube of given radius ε. As for classification the regression function only depends on a subset of the training data.

The training dataset $\{(x_i, y_i)\}_{i=1}^{n} \subset R^d \times R$ for a regression problem has x_i as the input vector and y_i as the output vector. The goal of SVR is to find such a function $f(x)$ which does not exceed a certain ε deviation. The ε deviation is the sum of all of the target variables for all the training dataset [15].

The f takes the form in case of linear function:

$$f(x) = \sum_{i=1}^{n} \mathbf{w_i x_i} + \mathbf{b} \quad \text{with} \quad \mathbf{w} \in R, \mathbf{b} \in R \tag{20}$$

By minimizing the norm, the ε deviation can be minimized: $\|\mathbf{w}\|^2 = \sum_{i=1}^{n} \mathbf{w_i}^T \mathbf{w_i}$. The problem can be formulated as a convex optimization problem:

$$\text{minimize} \qquad \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to} \begin{cases} y_i - \sum_{i=1}^{n} \mathbf{w_i x_i} - \mathbf{b} \leq \varepsilon, \\ \sum_{i=1}^{n} \mathbf{w_i x_i} + \mathbf{b} - y_i \leq \varepsilon. \end{cases}$$

3.2.4.5 Speeding up the Quadratic Program (16) is a time consuming task when there are large training sets. Many researchers have worked to solve this problem. Most of the methods decompose the large problems into the series of smaller problems. The Platt[24] method is the most known as Sequential Minimal Optimization.

3.2.4.6 Kernel Methods The learning problem and decision function can be described as the dot products between the data points in SVM. Other methods such as Principal Component Analysis, Canonical Correlation Analysis, Fisher Discriminant posses the same property. There are number of algorithms that solve the non-linear problems with the same complexity as the linear problems using kernel. These are called *kernel methods* [34, 33].

4 Numerical Experiments

4.1 Experimental setup and metrics

Two publicly available datasets were used for the experiments, by using data normalization in the range [0, 1]. There are three variants of ANFIS used here: G-ANFIS with standard regularization, F-ANFIS with fractional regularization, and C-ANFIS with subtractive clustering. G-ANFIS and F-ANFIS with scattered partition have been implemented in Scilab 6.0.2. MATLAB 2018a was used for C-ANFIS and ANFIS with grid partioning. The membership functions for all the ANFIS architectures were the classical bell-shaped function. The sigmoid and the linear functions were the activation functions for the hidden and output units of the NNs respectively. The experiments were performed by means of a CPU clocking in at 2.50GHz. The error measure herein considered is the root mean square error (RMSE), by taking into account the standard error, because of the cross validation. k–fold cross–validation is generally used to obtain reliable results. The idea is that each sample may be used as training data and test data. First, Data are split into k equal subsets, $k-1$ are used k times as training data and one as test data. Each time the test and training data are different. Usual values for k are 5 and 10.

4.2 Datasets

The first dataset is about forest fires in Portugal. It is obtained from UCI machine learning repository [3]. It contains 517 instances with 13 attributes. This is a difficult regression task, where the aim is to predict the burnt area of forest fires, in the northeast region of Portugal, by using meteorological and other data [12]. The attributes are listed in Table 1, where "D" stands for multivalued discrete and "C" for continuous.

variable	description	type
X	x-axis spatial coordinate	D
Y	y-axis spatial coordinate	D
month	-	D
day	-	D
FFMC	fine fuel moisture code	C
DMC	duff moisture code	C
DC	drought code	C
ISI	initial spread index	
temperature	-	C
RH	relative humidity	C
Wind	wind speed	C
Rain	outside rain	C
Area	burnt area	C (TARGET)

Table 1. First benchmark: dataset attributes

The second dataset is publicly available in the Mendeley Data repository [2]. The dataset presents 1714 istances and 3 attributes described in Table 2.

variable	description	type
NDVI	Normalized Difference Vegetation Index	C
LST	Land Surface Temperature	C
Area	burnt area	C

Table 2. Second benchmark: dataset attributes

4.3 Numerical results

4.3.1 First benchmark: Portugal forest fires dataset

A 10-fold cross-validation was performed on the Portugal forest fires dataset [3] in [15]. Two types of experiments were performed in [15]:

- using the actual values of the target variables,
- using the predicted values of the target variable using SVR.

The obtained numerical results were compared against the first experimental setup in [15], as it can be seen in Figure 8. The best results were achieved by F-ANFIS, with 3 terms (MFs), $C = 0.00001$, for any γ, since any change of this parameter does not affect the result significantly.

In Figure 8, the results by the methods CART, SVR + CART, DENFIS, SVR + DENFIS, ANFIS, and SVR + ANFIS are from [15] and results by F-ANFIS and G-ANFIS are from [36].

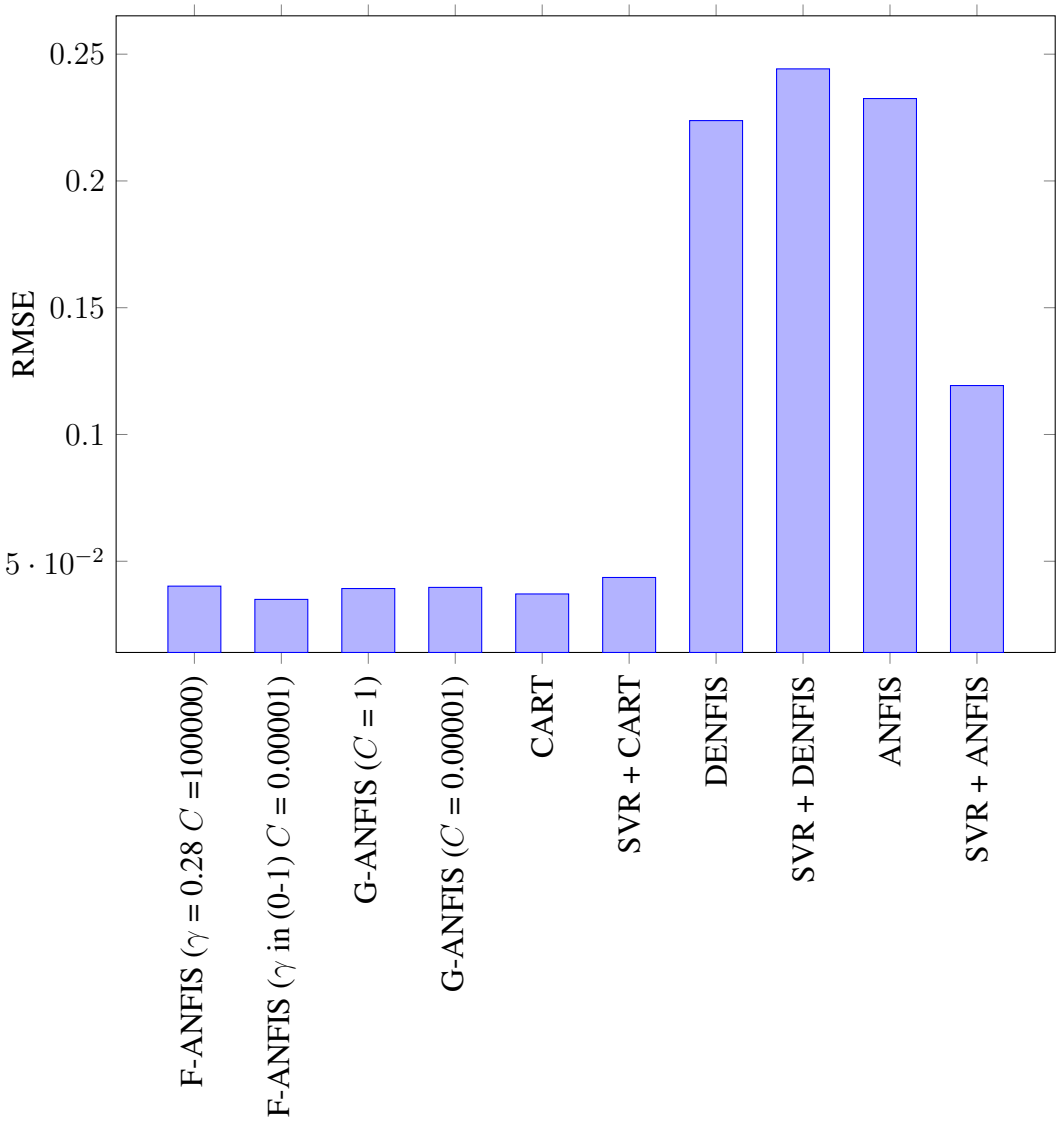

Figure 8. First benchmark: 10-fold cross validation test results

Figures 9-12 refer to the experiments performed with the 5-fold cross validation (in Figures 9-11, results by nu-SVR and C-ANFIS are from [36]). In the comparative analysis, NN and SVR were considered, because they seem to be the common computing schemes for forest fires prediction according to the recently published survey [5]. The best result was by F-ANFIS, with 3 terms (MFs), $C = 0.00001$, for $\gamma = 0.9$, even though

by changing γ the results did not vary significantly. This result is close to one by nu-SVR (linear kernel), which exhibits the shortest run-time. By changing the type of kernel the results did not change significantly. No results by the standard ANFIS with grid partitioning, with Matlab running out of memory.

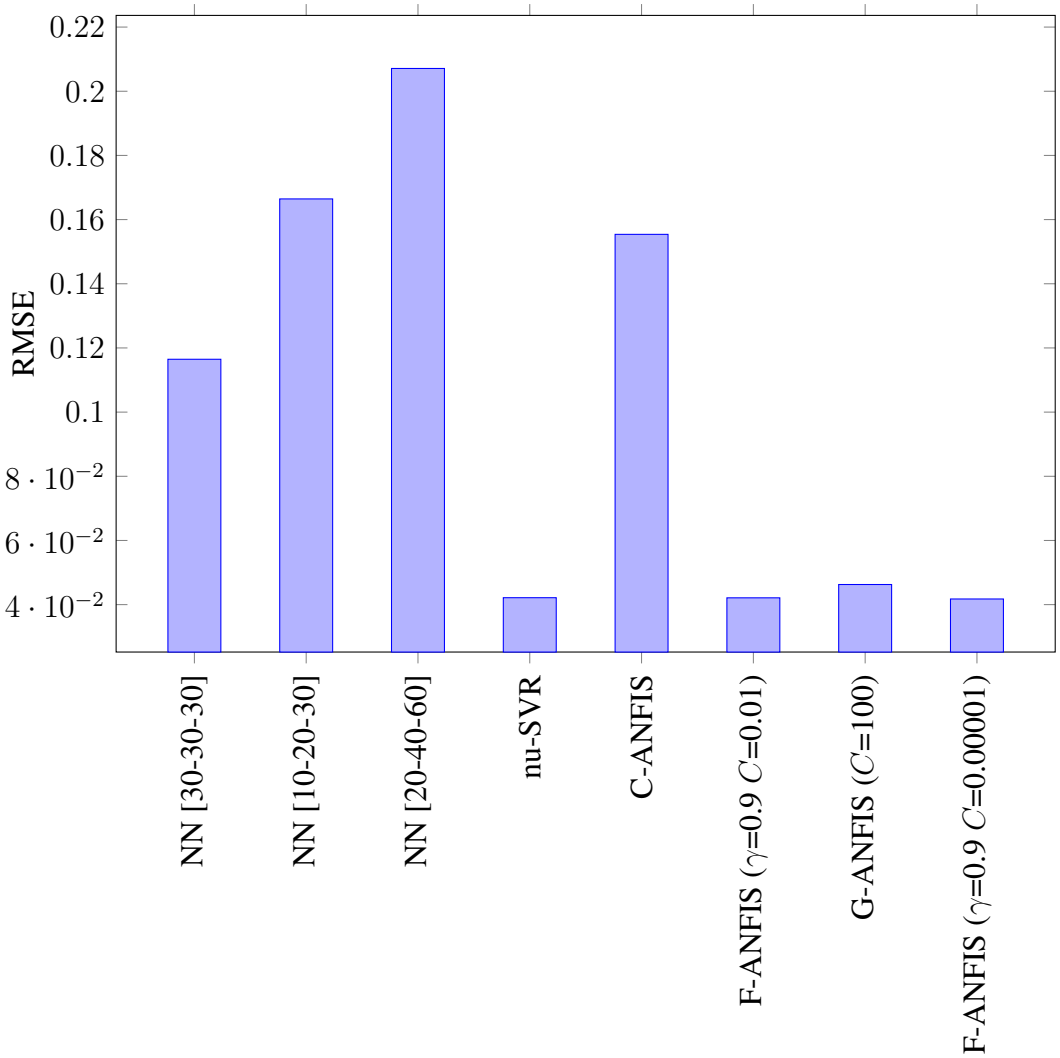

Figure 9. First benchmark: 5-fold cross validation test results

The standard error is presented in Figure 10. It can be seen that the C-ANFIS has the lowest standard error, while nu-SVR, F-ANFIS, and G-ANFIS have similar standard errors. All the NN architectures show a higher standard error.

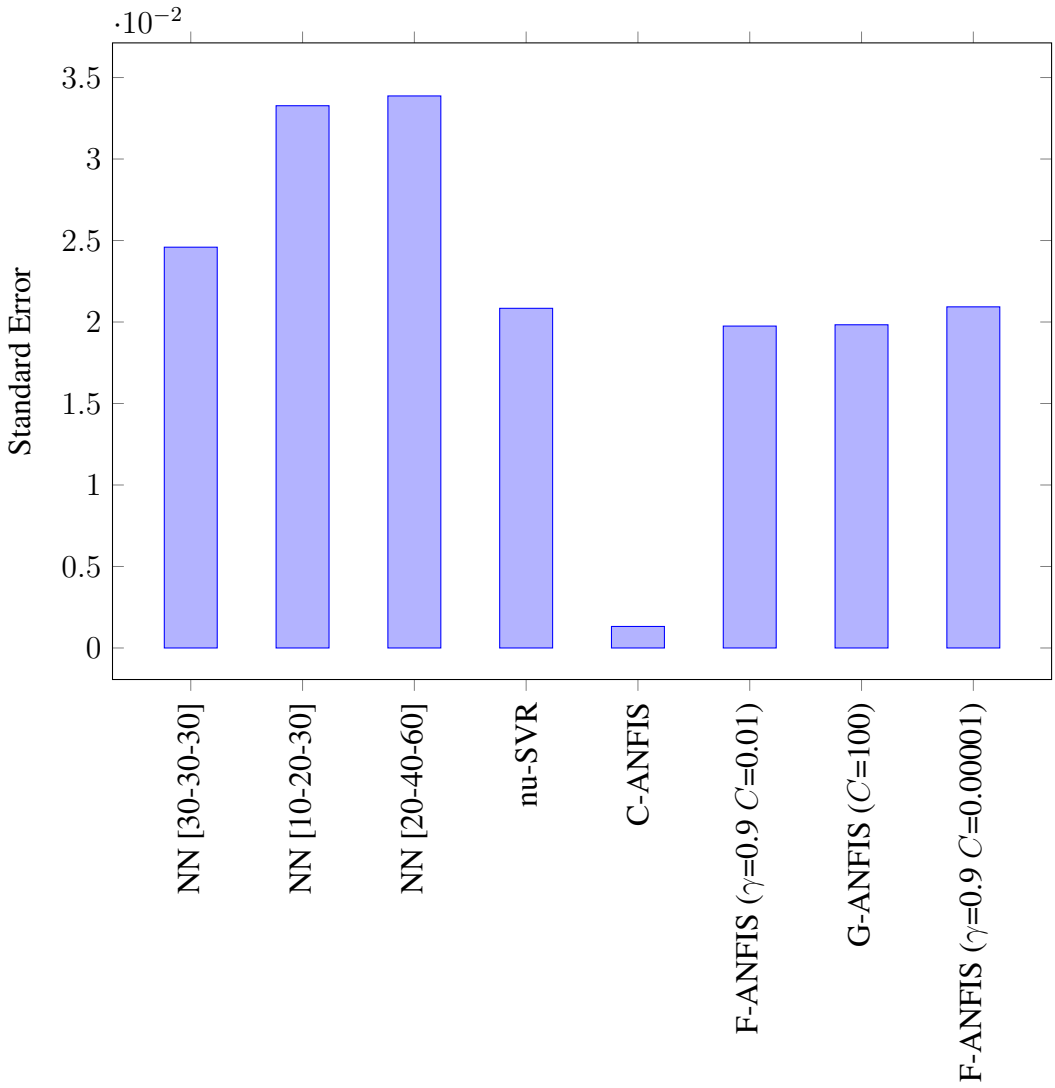

Figure 10. First benchmark: 5-fold cross validation standard errors measure

C-ANFIS shows the least standard error, but it takes a considerable amount of time as compared to other approaches. F-ANFIS, G-ANFIS, and nu-SVR have almost the same elapsed time, with nu-SVR the fastest one. The considered NNs also have less time elapsed as compared to C-ANFIS. The time elapsed by different approaches is shown in Figure 11.

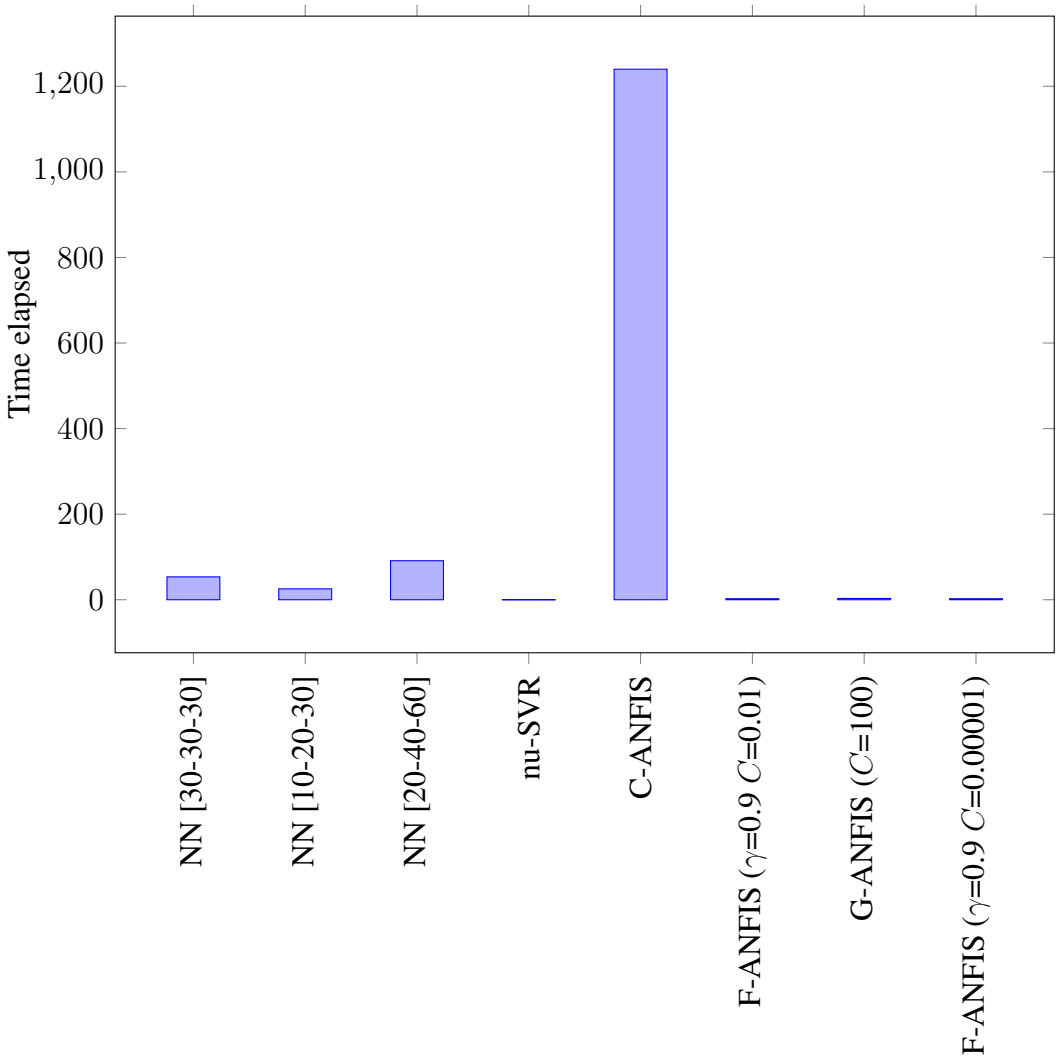

Figure 11. First benchmark: 5-fold cross validation elapsed time for each approach

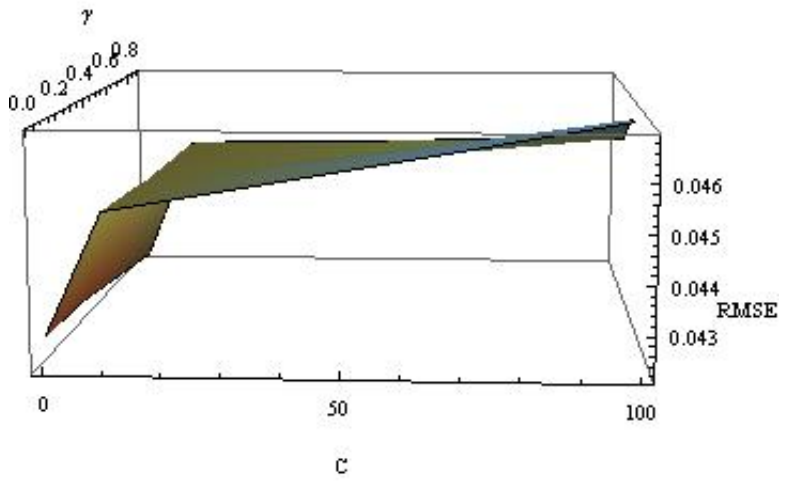

Figure 12. First benchmark: RMSE vs (C,γ) for F-ANFIS (5-fold cross validation test results)

Figure 12 provides the RMSE vs (C, γ) plot for the 5-fold cross-validation test results by F-ANFIS. In the considered range, that is $C \in (0, 100]$, the RMSE exhibits a quasi-linear variation.

4.3.2 Second benchmark: Canada forest fires dataset

The second dataset was retrieved from the Mendeley Data repository [2]. The original dataset presented 1714 istances and 3 input attributes. It was used for a binary classification (fire, non-fire) in [31], but in this thesis the fire instances have been used to predict the burnt area. The resulting dataset consists of 379 instances, whose attributes are shown in Table 2. In Figure 13-15, the results by C-ANFIS and nu-SVR are from [36]. As one can see in Figure 13, the mean RMSE by nu-SVR (linear kernel), C-ANFIS and G-ANFIS with 3 terms (MFs), $C = 100$ (or $C = 10$, as in [36]) is almost the same. By changing the type of kernel in nu-SVR, the results do not change significantly. In this example, the result by F-ANFIS are not very good and the worst ones are again by the NNs.

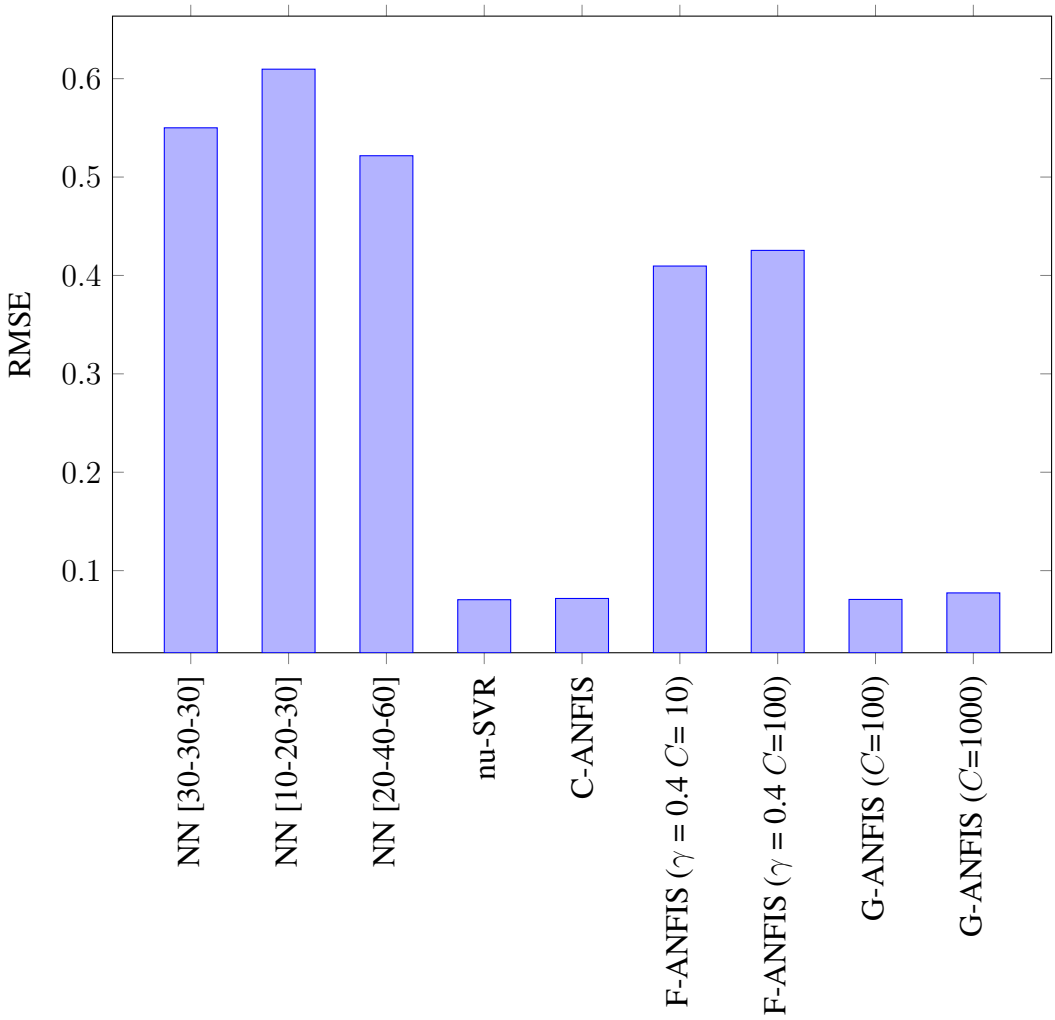

Figure 13. Second benchmark: 5-fold cross validation test results

The standard error is high for neural network with architecture [30 30 30], that is 3 hidden layers with 30 units in each layer. The other NNs have similar standard error with F-ANFIS. C-ANFIS, G-ANFIS, and nu-SVR has the lowest and similar standard error which is in the Figure 14.

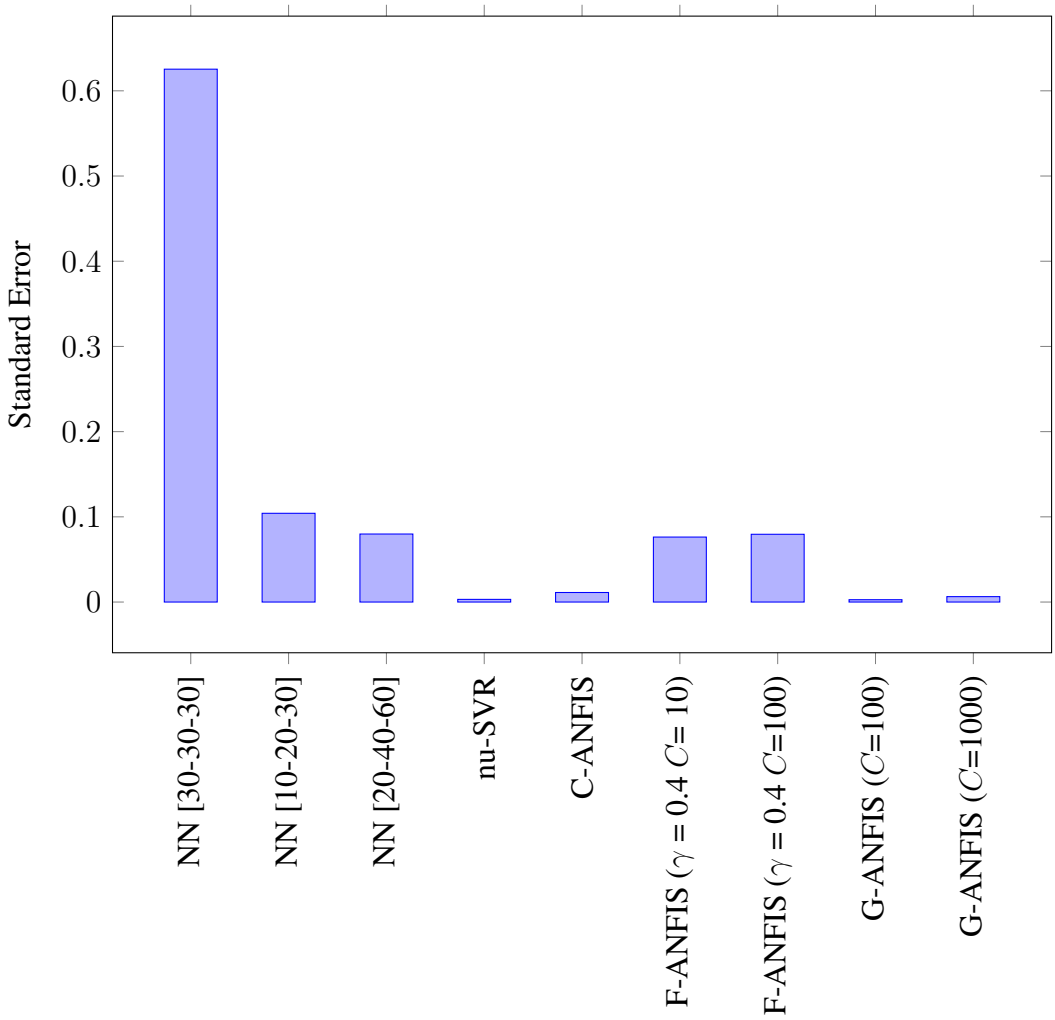

Figure 14. Second benchmark: 5-fold cross validation standard errors measure

The time elapsed by different approaches is shown in Figure 15. All the ANFIS approaches took almost the same time. Again, nu-SVR is the fastest one. All the NNs have a higher run-time, and the NN [20 40 60] takes more time compared to other NNs.

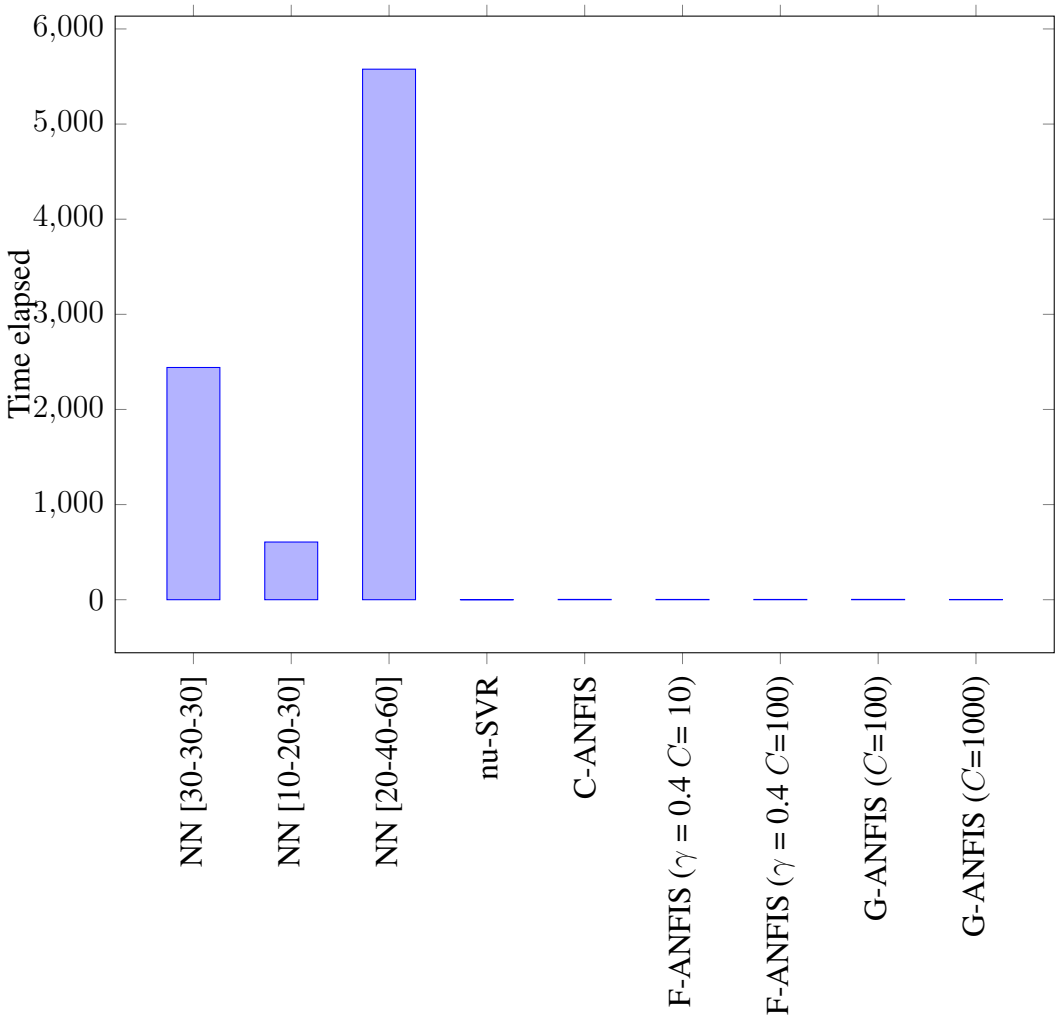

Figure 15. Second benchmark: 5-fold cross validation elapsed time for each approach

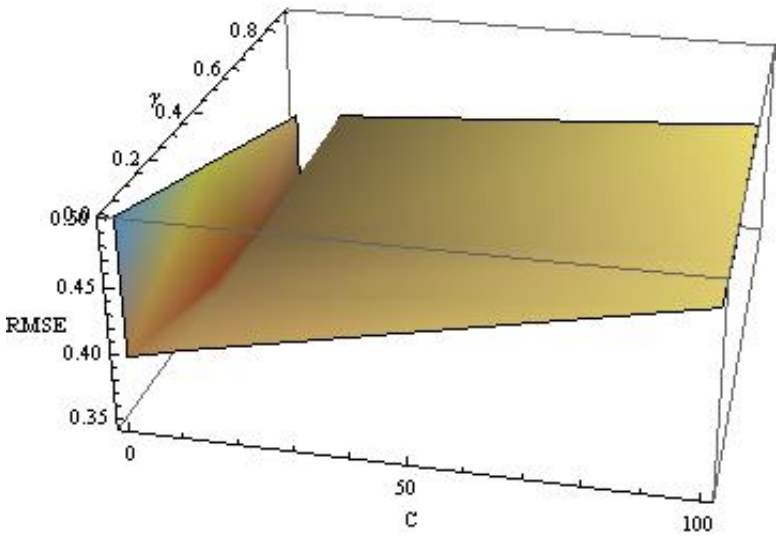

Figure 16. Second benchmark: RMSE vs (C,γ) for F-ANFIS (5-fold cross validation test results)

Figure 16 is the RMSE vs (C,γ) plot for F-ANFIS, which shows similar behaviour as in the previous example.

Even though the classification problem is beyond the scope of this thesis, a few experiments with this Canadian dataset as the original classification problem in [31] were performed. Since the authors in [31] did not provide the dataset with the training and test part, it was not possible to compare our results against those results. Anyhow, 566 samples were used for the training with 133 fire cases, and 238 samples with 60 fires cases were used for the test stage. The following error measures were considered, being TP the true positives, TN the true negatives, FP the false positives, FN the false negatives:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$
$$Precision = TP/(TP + FP)$$
$$Recall = TP/(TP + FN)$$
$$F - measure = 2 * (Precision * Recall)/(Precision + Recall)$$

Confusion matrix:

$$\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix}$$

The best results sought are tabled in Table 3.

Approach	**Accuracy**	**Precision**	**Recall**	**F-measure**	**time (s)**
G-ANFIS (C=100000)	0.7731092	0.6875	0.1833333	0.2894737	0.1006869
ANFIS	0.7650	–	–	–	7.103449
SVM	0.7478992	–	0	–	0.0028625
NN [50-50]	0.7479	0.7478992	–	–	962.53

Table 3. Classification problem: test results

The confusion matrices are

- for G-ANFIS (2 terms)

$$\begin{bmatrix} 11 & 5 \\ 49 & 173 \end{bmatrix}$$

- for ANFIS (2 terms)

$$\begin{bmatrix} 0 & 133 \\ 0 & 433 \end{bmatrix}$$

- for the SVM

$$\begin{bmatrix} 0 & 0 \\ 60 & 178 \end{bmatrix}$$

- for the NN

$$\begin{bmatrix} 0 & 0 \\ 60 & 178 \end{bmatrix}$$

The results are not good in general, but the results by G-ANFIS (with 2 terms) are the best ones. The results by SVM did not change significantly by changing the kernel (here linear).

5 Concluding Remarks

In this thesis, different soft computing techniques to predict the burnt area from a forest fires have been applied. Two publicly available datasets [2, 3] were used. The soft computing techniques that were applied includes ANFIS, ANN, and SVR, which are state-of-the-art techniques. Regarding ANFIS, a new version using both standard and fractional regularization has been discussed. The aim of this variant is to provide reliable results even in case of ill-posedness with a better computational cost with respect to the classical ANFIS, as mentioned in the Motivation and Main Goals section. From the numerical experiments, it has been possible to see that G-ANFIS (equipped with the standard regularization) and F-ANFIS (based on the fractional regularization), with a proper choice of their parameters, outperformed most of the other approaches. For the first benchmark case, the only approach whose results are close to those by F-ANFIS (that are the best ones), is $nu - SVR$, but the advantage against the latter is in the interpretability. While the interpretability of $nu - SVR$ has still to be proved (to our best knowledge), there is a huge literature on the interpretability of ANFIS. For the second benchmark case, the best result is by G-ANFIS, which is close to that by C-ANFIS (ANFIS with subtractive clustering), but the latter has a higher computational cost as confirmed by the elapsed time in the numerical experiments. A proper choice of the parameters of the proposed variant to get a may represent a limitation of it. This is a common problem in many machine learning algorithms. Future work will address this issue.

References

[1] Estonian forest fire data. https://www.keskkonnaagentuur.ee/sites/default/files/mets2017.pdf. Accessed: 2020-10-12.

[2] Mendeley data repository. https://data.mendeley.com/datasets/85t28npyv7/1. Accessed: 2020-11-06.

[3] Portugal dataset. https://archive.ics.uci.edu/ml/datasets/Forest+Fires. Accessed: 2020-10-30.

[4] A quick introduction to neural networks. https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/. Accessed: 2021-01-25.

[5] F. Abid. A survey of machine learning algorithms based forest fires prediction and detection systems. *Fire Technology*, in press, 2020.

[6] K. Ahmed, A. A. Ewees, and A. E. Hassanien. Prediction and management system for forest fires based on hybrid flower pollination optimization algorithm and adaptive neuro-fuzzy inference system. In *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pages 299–304, 2017.

[7] S. Al_Janabi, I. Al_Shourbaji, and M. A. Salman. Assessing the suitability of soft computing approaches for forest fires prediction. *Applied Computing and Informatics*, 14(2):214 – 224, 2018.

[8] M. Anshori, F. Mar'i, M. W. Alauddin, and W. F. Mahmudy. Prediction of forest fire using neural network based on extreme learning machines (elm). In *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*, pages 301–305, 2019.

[9] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144–152, New York, NY, USA, 1992. Association for Computing Machinery.

[10] P. Chaudhary and A. Sinha. *Sustainable Approach for Forest Fire Prediction*, pages 456–469. 04 2020.

[11] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 2004.

[12] P. Cortez and A. Morais. A data mining approach to predict forest fires using meteorological data. 01 2007.

[13] N. Cristianini and E. Ricci. *Support Vector Machines*, pages 928–932. 01 2008.

[14] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In *NIPS*, 1996.

[15] M.A.H. Farquad, V. Ravi, and S. Bapi Raju. Support vector regression based hybrid rule extraction methods for forecasting. *Expert Systems with Applications*, 37(8):5577–5589, 2010.

[16] L.S. Iliadis. A decision support system applying an integrated fuzzy model for long-term forest fire risk estimation. *Environmental Modelling & Software*, 20(5):613 – 621, 2005.

[17] J. S. R. Jang. Anfis: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems Man & Cybernetics*, 23:665–685, 1993.

[18] G. Karapilafis, L. Iliadis, S. Spartalis, S. Katsavounis, and E. Pimenidis. Modeling spatiotemporal wild fire data with support vector machines and artificial neural networks. In Lazaros Iliadis, Harris Papadopoulos, and Chrisina Jayne, editors, *Engineering Applications of Neural Networks*, pages 132–143, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[19] L. Liadis, M. Vangeloudh, and S.H. Spartalis. An intelligent system employing an enhanced fuzzy c-means clustering model: Application in the case of forest fires. *Computers and Electronics in Agriculture*, 70:276–284, 03 2010.

[20] H. Lin, X. Liu, X. Wang, and Y. Liu. A fuzzy inference and big data analysis algorithm for the prediction of forest fire based on rechargeable wireless sensor networks. *Sustain. Comput. Informatics Syst.*, 18:101–111, 2018.

[21] Z. Liu, K. Zhang, C. Wang, and S. Huang. Research on the identification method for the forest fire based on deep learning. *Optik*, 223:165491, 2020.

[22] Z. Mohammed, C. Hanae, and S. Larbi. Comparative study on machine learning algorithms for early fire forest detection system using geodata. *International Journal of Electrical and Computer Engineering (IJECE)*, 10(5):5507 – 5513, 2020.

[23] Shraddha M. N., Ravi Prasad K. Jagannath, and Venkatanareshbabu Kuppili. Fractional tikhonov regularization to improve the performance of extreme learning machines. *Physica A: Statistical Mechanics and its Applications*, 551:124034, 2020.

[24] J. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press, January 1998.

[25] H. R. Pourghasemi, A. Gayen, R. Lasaponara, and J. P. Tiefenbacher. Application of learning vector quantization and different machine learning techniques to assessing forest fire influence factors and spatial modelling. *Environmental Research*, 184:109321, 2020.

[26] J. Prakash, D. Sanny, S. K. Kalva, M. Pramanik, and P. K. Yalavarthy. Fractional regularization to improve photoacoustic tomographic image reconstruction. *IEEE transactions on medical imaging*, 38(8):1935—1947, August 2019.

[27] C.P. Pramod and G.N. Pillai. K-means clustering based extreme learning anfis with improved interpretability for regression problems. *Knowledge-Based Systems*, 215:106750, 2021.

[28] V. Raj, S. Ditipriya, D. Debashis, and D. A. Kumar. Eeffl: energy efficient data forwarding for forest fire detection using localization technique in wireless sensor network. *Wireless Networks*, 26:5177 – 5205, 06 2020.

[29] S. Richa, S. Rani, and I. Memon. A smart approach for fire prediction under uncertain conditions using machine learning. *Multimedia Tools and Applications*, 2020.

[30] Y. Safi and A. Bouroumi. Prediction of forest fires using artificial neural networks. *Applied Mathematical Sciences*, 7:271–286, 01 2013.

[31] Y. O. Sayad, H. Mousannif, and H. A. Moatassime. Predictive modeling of wildfires: A new dataset and machine learning approach. *Fire Safety Journal*, 104:130–146, 2019.

[32] A. Schumann, J. Toledo-Castro, P. John Caballero-Gil, N. Rodríguez-Pérez, I. Santos-González, C. Hernández-Goya, and R. Aguasca-Colomo. Forest fire prevention, detection, and fighting based on fuzzy logic and wireless sensor networks. *Complexity*, 2018:104663, 2018.

[33] B. Schölkopf and A.J. Smola. *Smola, A.: Learning with Kernels - Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, Cambridge, MA*, volume 98. 01 2001.

[34] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[35] B. K. Singh, N. Kumar, and P. Tiwari. Extreme learning machine approach for prediction of forest fires using topographical and metrological data of vietnam. In *2019 Women Institute of Technology Conference on Electrical and Computer Engineering (WITCON ECE)*, pages 104–112, 2019.

[36] S. Tomasiello and M.Uzair. –. In *Proceedings of FUZZ-IEEE 2021 (to appear)*, 2021.

[37] Ioannis A. Troumbis, George E. Tsekouras, J. Tsimikas, C. Kalloniatis, and D. Haralambopoulos. A chebyshev polynomial feedforward neural network trained by differential evolution and its application in environmental case studies. *Environmental Modelling Software*, 126:104663, 2020.

[38] V. N. Vapnik. *The Nature of Statistical Learning Theory*.

[39] Y. Xie and M. Peng. Forest fire forecasting using ensemble learning approaches. *Neural Computing and Applications*, 05 2018.

[40] C. Yeom and K. Kwak. A performance comparison of anfis models by scattering partitioning methods. *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 814–818, 2018.

Appendix

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Muhammad Uzair**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Soft Computing Techniques For Forest Fires Prediction,

   supervised by Dr. Stefania Tomasiello.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work, from 25/03/2024 until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Muhammad Uzair
08/03/2021