Turnaround Time Prediction for a Medical Laboratory Using Artificial Neural Networks

Araştırma Makalesi/Research Article

Mete EMİNAĞAOĞLU*, Alper VAHAPLAR

Faculty of Science, Department of Computer Science, Dokuz Eylül University, İzmir, Turkey
mete.eminagaoglu@deu.edu.tr, alper.vahaplar@deu.edu.tr
(Geliş/Received:17.07.2018; Kabul/Accepted:05.10.2018)
DOI: 10.17671/gazibtd.444826

***Abstract*—** Turnaround time (TAT) or duration between different stages in medical and healthcare services is accepted to be one of the most significant performance measures that can have a great impact on service quality, change management, costs, and strategic decisions. Accurate and reliable prediction or estimation of the turnaround times or elicitation of the underlying causes that affect TAT is known to be a difficult problem. In this study, a heuristic prediction approach is used by designing and implementing a special artificial neural network (ANN) model in order to predict TAT of a specific process in a private hospital. The prediction performance of our ANN model is comparatively analyzed with some alternative linear and nonlinear numerical prediction algorithms. The results show that ANN surpasses all of the other numerical prediction algorithms and ANN might be used by the decision makers as a reliable model to estimate TAT within acceptable error rates.

***Keywords*—** turnaround time, artificial neural networks, numerical prediction, medical laboratory, multilayer perceptron, health services

Yapay Sinir Ağları ile Tıbbi Laboratuvar için İşlem Süresi Kestirimi

***Özet*—** Hastanelerde ve çeşitli sağlık hizmetlerinde tıbbi işlemler / aşamalar arasındaki işlem ya da geri dönüş süresi, hizmet kalitesi, değişim yönetimi, maliyetlerin azaltılması ve stratejik kararlar üzerinde de etkisi olan en önemli performans ölçütlerinden biri olarak kabul edilmektedir. Geri dönüş sürelerinin doğru ve güvenilir tahmini ya da bu süreleri etkileyen etmenlerin veya nedenlerin ortaya çıkarılması ise çözümü zor bir problemdir. Bu çalışmada, özel bir hastanedeki çeşitli birimler arasındaki tıbbi iş süreçlerine ait gerçek istatistiksel işlem süresi verileri kullanılarak iş bitirme sürelerinin sayısal olarak tahmini için özel bir yapay sinir ağı (YSA) modeli tasarlanmış ve kodlanmıştır. YSA modelimizin kestirim performansı, bazı alternatif doğrusal / doğrusal olmayan sayısal kestirim algoritmaları ile karşılaştırmalı olarak analiz edilmiştir. YSA'nın tahmin başarısı ve hata değerleri açısından diğer tüm sayısal kestirim algoritmalarından daha başarılı olduğu ve YSA'nın, tıbbi iş süreçlerinde iş bitirme sürelerini kabul edilebilir hata oranlarında güvenilebilir şekilde tahmin edebildiği ve karar destek sistemlerinde yöneticiler tarafından alternatif bir model olarak kullanılabileceği ortaya konmuştur.

***Anahtar Kelimeler*—** işlem süresi, yapay sinir ağları, sayısal kestirim, tıbbi laboratuvar, çok katmanlı algılayıcı, sağlık hizmetleri

1. INTRODUCTION

Information and communications technology have been effective and efficient in various areas of healthcare and medical science in the recent years. Information technologies have a great impact on the improvement of the quality of the healthcare services, which significantly affect management of information systems in healthcare. Advances in information technologies such as control systems, computer networks, software, and their proper alignment with business processes in healthcare systems enable new approaches and solutions [1-7]. There have

also been some researches that are involved with enhancing the quality of business processes in laboratories, clinics, and other medical organizations, which focus on the turnaround times or duration in related services [8-12].

Turnaround time (TAT) is generally described as the amount of time or duration taken to fulfill a request or process. TAT within medical services and healthcare systems is usually described as the time spent for a particular analysis or during any stage in medical laboratory, other commercial laboratories or a public health laboratory [8, 13]. Processing time for tests is often considered as a significant performance measure in medical services; such as laboratory turnaround time is shown to be a reliable indicator of laboratory effectiveness [8] or the reduction of patient turnaround time in emergency departments is shown to have an impact on the costs, service quality, and patients' satisfaction [9, 11, 12, 14, 15]. There are a few reliable researches or models that focus on reliable estimation or prediction of the durations among such medical services, which could help the managers to make appropriate decisions to decrease such durations or to discover the causes that affect such turnaround times. This has been the primary motivation for us to make a research in this field, which is elaborated in this article. A multilayer perceptron (an artificial neural network architecture) is designed and implemented to predict TAT, which is the duration between the production of test results and delivery of report to the patient in a medical laboratory.

There are several well-known models, and methodologies for numerical estimations or predictions, which are used among different areas of applied sciences. Some of these are also used in machine learning and data mining recently, and some new alternative models and heuristic approaches have been derived. Data mining can be simply defined as the extraction of valuable or meaningful information from large data. It is said to be located at the intersection of data management, statistics, machine learning, pattern recognition, and artificial intelligence [16]. Machine learning can be simply described as a technique that derives technical foundations for data mining. Support vector machines, artificial neural networks, Bayesian networks, and decision trees are some of the algorithms and models that are included in the area of machine learning [17-19].

2. ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANN) are inspired from biological neural networks in brain and they can be defined as systems having nodes with interconnections that make simple or complex computations from the given inputs [20]. They are mostly used for binary or multi-classification, numerical prediction, or clustering tasks that usually have several inputs or features. Artificial neural networks are used to solve diverse tasks or problems such as decision making, time-series prediction, computer aided design, forecasting, pattern and speech recognition, and so on [21-25]. The feed-forward learning models, which are a type of ANN, are usually named as multilayer perceptron (MLP) [26]. In MLP, the differential errors caused by non-linear activations and the corresponding weight updates are usually calculated by backpropagation methodology [20, 27], which is also used in this study.

The linear input function for each of the hidden layers' units and for the output layer is given in equation (1), which is mostly used in MLP [26]. For each unit j in a layer, the net input value I_j is calculated where W_{ij} stands for the connection's weight between the previous layer's unit i and unit j; O_i is the output of unit i; and θ_j stands for the bias value.

$$I_j = \sum_i W_{ij}\ O_i + \theta_j \qquad (1)$$

In ANN, the weights and bias values are usually initialized randomly to values between −0.5 to 0.5 or −1.0 to 1.0. These weights and bias values are updated during the training phase of the neural network model. The network structure in any ANN learns by adjusting the weights in order to make better or more accurate predictions or classifications [28]. Non-linear activations are used to compute the corresponding output value for each unit. Sigmoid function is usually used for the non-linear activation [28]. The output value is calculated by using the sigmoid function that is given in equation (2), where the corresponding input value to unit j is denoted by I_j that is also denoted equation (1), and O_j is the output of unit j.

$$O_j = \frac{1}{1 + e^{-I_j}} \qquad (2)$$

It should be noted that we also used hyperbolic tangent function [29] as well as sigmoid function as non-linear activation functions in our MLP model. The output value derived by using the non-linear hyperbolic tangent function is given in equation (3).

$$O_j = \frac{e^{I_j} - e^{-I_j}}{e^{I_j} + e^{-I_j}} \qquad (3)$$

For numerical prediction tasks, the output layer is composed of one single node and the predicted value of that instance is established by the value derived from the node within the output layer. The difference between the

actual value and the value that is predicted is named as the prediction error. Backpropagation is one of the well-known techniques that can be used to calculate the errors and the new connection weights in ANN [30]. Backpropagation method is usually used with the "gradient descent" optimization algorithm in order to update the new weights in a more feasible way [27]. The backpropagation method for calculating the error in the output layer is given in the equation (4). It should be noted that *j* represents the output node, O_j stands for the predicted outcome and R_j is the original value for that instance.

$$Err_j = O_j(1 - O_j)(R_j - O_j) \quad (4)$$

It should be noted that $O_j\,(1 - O_j)$ is established by the derivative of the non-linear sigmoid output function. The errors of the nodes connected to node *j* are multiplied by its corresponding weights and summed up to calculate the error of a node *j* within any hidden layer in the neural network. The calculation of the error in a hidden layer node *j* is shown in equation (5) where W_{jk} denotes the connection's weight from node *j* to a node *k*, and Err_k stands for the error in node *k*.

$$Err_j = O_j(1 - O_j)\sum_k Err_k W_{jk} \quad (5)$$

Training an ANN is established by the computation of these errors in the forward pass and update of the weights and biases in the backward pass within each iteration. The iteration is also named as "epoch" in ANN terminology, an epoch can be defined as a single pass through the entire training set within ANN model where backpropagation errors are calculated, and nodes' weights are updated [17]. The weight update calculation within each connection is given in the equations (6) and (7) where $\Delta W_{ij(k-1)}$ is the change in W_{ij} in the previous $(k\text{-}1)^{th}$ iteration, W'_{ij} is the new value for the k^{th} iteration, Err_j denotes the error in node *j*, O_i is output from node *i* and λ denotes the learning rate of the ANN where $(0 < \lambda < 1)$.

$$\Delta W_{ij} = \lambda\, Err_j\, O_i \quad (6)$$

$$W'_{ij\,(k)} = W_{ij\,(k)} + \lambda\, Err_j O_i + m\left(\Delta W_{ij\,(k-1)}\right) \quad (7)$$

It should be noticed that there is a constant parameter *m* in equation (7), and this parameter is named as momentum. This momentum value is used to prevent the system from converging to a local optimum.

Biases are also updated in a similar manner by using the equations given in (8) and (9), where $\Delta\,\theta_j$ denotes the change in bias θ_j.

$$\Delta\theta_i = \lambda\, Err_i \quad (8)$$

$$\theta'_{j\,(k)} = \Delta\theta_{j\,(k)} + \lambda\, Err_j + m\left(\Delta\,\theta_{j\,(k-1)}\right) \quad (9)$$

The training process in ANN is established by updating the weights and bias values for each round of iteration or epoch based on these error values. An epoch can be defined as a single pass through the entire training set within ANN model where backpropagation errors are calculated and nodes' weights are updated [17]. It should be noticed that a momentum is also used to prevent the system from converging to a local optimum. In other words, this parameter helps to increase learning rate in a balanced manner while minimizing the instability risk in ANN [27].

Min-max normalization [12, 28] is mostly used as a statistical data transformation method for all of the continuous variables in any ANN model, which is given in equation (10).

$$X_{norm.} = \frac{X - \min(X)}{\max(X) - \min(X)} \quad (10)$$

3. MATERIALS AND METHODS

The data used in this study was provided from the medical laboratory's database in one of the private hospitals in İstanbul, Turkey and this real data had been collected over a three-year period. This study was carried out according to Turkish legislation on information security and privacy, thus no confidential or private data such as hospital name, patient names, social security numbers, and patient ID's were used in this study. There were a total of 149892 records consisting nine different features such as "Patient ID", "Department Name", "Date and time of patient registration", "Date and time of request for a physician service", "Specimen collection date and time", "Laboratory sample receipt date and time", "Date and time of report for test results", "Date and time of the delivery of the report to the patient", and "Date and time of patient's visit to physician". A sample of the data is shown in Fig. 1. Some of this data has been used in another study, however in that study, the features and the attribute to be predicted is different, and an entirely different model, which is known as "adaptive network-based fuzzy inference system", is used in that study [9].

It was known that one of the main objectives of the hospital's senior management was to achieve a reliable

model to predict the durations or TAT between several stages of healthcare services that might be useful for their decision-making processes. Several different combinations of TAT could be derived and analyzed such as; TAT for "Date and time of patient registration" and "Specimen collection date and time", or duration between "Laboratory sample receipt date and time" and "Date and time of report for test results".

These TAT values were calculated from the original data set and two TAT values among these were chosen as feature attributes, which were; the duration between the stages "Laboratory sample receipt date and time" and "Specimen collection date and time"; and the duration between the stages "Date and time of report for test results" and "Laboratory sample receipt date and time", namely.
There were several candidates among different TAT values to be used as the attribute for prediction. TAT for "Date and time of the delivery of the report to the patient" and "Date and time of report for test results" were chosen as the attribute for prediction because the analysis and estimation of the duration between those two stages were crucial for the hospital managers. It had been previously observed that there were unexpected delays or latencies between the stages "Date and time of the delivery of the report to the patient" and "Date and time of report for test results". In addition, the duration between these two stages had significant outlier values and the hospital management was curious whether these had been occurring within some specific departments or due to some unique days during the week or some unique hours during the day. The duration / turnaround time between the stages "Date and time of the delivery of the report to the patient" and "Date and time of report for test results" would be named and abbreviated as "TAT-target" in this article.

Patient ID	Department Name	Patient registration Date / Time	Request for a physician service Date / Time	Specimen collection Date / Time	Sample receipt Date / Time	Report of test results Date / Time	Delivery of report to patient Date / Time	Patient's visit to physician Date / Time
120190	Check-Up	4.09.2015 08:36	4.09.2015 08:37	4.09.2015 08:33	4.09.2015 09:04	4.09.2015 09:31	4.09.2015 09:33	4.09.2015 10:15
120476	Cardiology	4.09.2015 08:48	4.09.2015 09:13	4.09.2015 09:13	4.09.2015 09:19	4.09.2015 10:42	4.09.2015 10:45	4.09.2015 10:46
33781	Urology	4.09.2015 09:58	4.09.2015 10:42	4.09.2015 10:43	4.09.2015 11:08	4.09.2015 11:20	4.09.2015 11:20	4.09.2015 11:26
120174	Nephrology	4.09.2015 10:58	4.09.2015 11:16	4.09.2015 11:13	4.09.2015 11:51	4.09.2015 12:13	4.09.2015 12:13	4.09.2015 12:17
19268	Cardiology	4.09.2015 14:02	4.09.2015 14:03	4.09.2015 14:11	4.09.2015 14:19	4.09.2015 14:40	4.09.2015 14:40	4.09.2015 14:44
79256	Emergency	4.09.2015 14:54	4.09.2015 14:59	4.09.2015 15:01	4.09.2015 15:16	4.09.2015 15:38	4.09.2015 15:38	4.09.2015 15:42
120592	Emergency	4.09.2015 15:11	4.09.2015 15:20	4.09.2015 16:10	4.09.2015 16:24	4.09.2015 16:30	4.09.2015 16:30	4.09.2015 16:33
74869	Emergency	4.09.2015 17:50	4.09.2015 17:51	4.09.2015 18:01	4.09.2015 18:02	4.09.2015 18:34	4.09.2015 18:34	4.09.2015 18:34
120662	Emergency	4.09.2015 21:06	4.09.2015 21:17	4.09.2015 21:19	4.09.2015 21:32	4.09.2015 21:34	4.09.2015 21:34	4.09.2015 21:34
87336	Emergency	5.09.2015 05:15	5.09.2015 05:55	5.09.2015 05:56	5.09.2015 05:56	5.09.2015 06:11	5.09.2015 06:11	5.09.2015 06:11
120594	Neurology	5.09.2015 08:47	5.09.2015 09:56	5.09.2015 09:59	5.09.2015 10:09	5.09.2015 11:03	5.09.2015 11:04	5.09.2015 11:08
120521	Dermatology	5.09.2015 09:42	5.09.2015 10:11	5.09.2015 10:14	5.09.2015 10:26	7.09.2015 10:09	7.09.2015 10:09	7.09.2015 10:38
118464	Neurology	5.09.2015 09:46	5.09.2015 10:37	5.09.2015 10:38	5.09.2015 10:43	5.09.2015 11:54	5.09.2015 11:57	5.09.2015 12:04
120149	Nephrology	5.09.2015 10:17	6.09.2015 09:18	6.09.2015 09:27	6.09.2015 09:37	6.09.2015 11:13	6.09.2015 11:16	6.09.2015 11:17

Figure 1. Sample of the data set retrieved from medical laboratory database

It should be noted that some of the attributes in the data set that was used in this study were also converted to different data types and formats in order to make it feasible for both ANN architecture and prediction of numerical values. "Department Name" attribute was originally a categorical attribute with different nominal values like "Neurology", "Check-up", etc. Since ANN can only use numerical attributes as input data, the categorical values in "Department Name" were converted into binary encoded values by using the method named as one-hot encoding or dummy encoding. [30].

The durations between the stages "Laboratory sample receipt date and time" and "Specimen collection date and time"; "Date and time of report for test results" and "Laboratory sample receipt date and time"; and "Date and time of the delivery of the report to the patient" and "Date and time of report for test results" are all calculated by transforming them into time unit as seconds and then calculating the differences. "TAT-target" is one of those such numerical attributes that should be predicted accurately by our ANN model.

"Specimen collection date and time", "Laboratory sample receipt date and time", "Date and time of report for test results", and "Date and time of the delivery of the report to the patient" were also chosen as feature attributes in the data set and they also had to preprocessed. These attributes were originally encoded in a specific date and time notation, and in order to use a data format appropriate for

ANN, they were transformed into "day of week" (categorical values such as "Sunday", "Friday", etc.) and "hourly time interval" integer values, ranging between one and twenty-four. Thus, a data set with 149892 instances and 12 attributes was established. A sample from this data set is shown in Fig. 2, where it could be seen that the right-most column is the "TAT-target" attribute, which is to be predicted in this study.

Using min-max normalization, all of the numerical attributes were normalized between values 0 and 1. This was done to normalize the numerical variables and to make it feasible ANN. Thus, our data set consisted of min-max normalized numerical attributes and some one-hot encoded attributes that were originally nominal.

It should be noted that instead of the entire data set with 149892 instances (records), two different subsample data sets were derived and used in this study. This was a requirement since the execution time of some of the algorithms were significantly beyond the hospital management's acceptable limits. Hence, statistical random sampling without replacement methodology was used to derive smaller sample data sets [30].

Department Name	Specimen collection (hourly time interval)	Specimen collection (day of week)	Sample receipt (hourly time interval)	Sample receipt (day of week)	Report of test results (hourly time interval)	Report of test results (day of week)	Delivery of report to patient (hourly time interval)	Delivery of report to patient (day of week)	TAT ((sample receipt) - (specimen collection)) time unit: seconds	TAT ((report of test results) - (sample receipt)) time unit: seconds	TAT ((delivery of report to patient) - (report of test results)) time unit: seconds
Emergency	2	Saturday	3	Saturday	3	Saturday	3	Saturday	1440	60	60
Emergency	8	Saturday	8	Saturday	8	Saturday	8	Saturday	240	240	60
Cardiology	9	Saturday	10	Saturday	10	Saturday	10	Saturday	720	1860	720
Cardiology	9	Saturday	10	Saturday	11	Saturday	11	Saturday	720	4500	60
Urology	11	Saturday	12	Saturday	13	Saturday	13	Saturday	3360	2640	60
Urology	11	Saturday	12	Saturday	13	Saturday	13	Saturday	3360	2580	120
Psychiatry	12	Saturday	13	Saturday	14	Saturday	14	Saturday	3840	1800	0
Emergency	22	Sunday	22	Sunday	22	Sunday	22	Sunday	120	120	60
Emergency	8	Monday	8	Monday	9	Monday	9	Monday	720	1260	0
Emergency	20	Friday	20	Friday	20	Friday	20	Friday	240	840	0
Emergency	20	Friday	20	Friday	20	Friday	20	Friday	240	540	300
Emergency	20	Friday	20	Friday	11	Monday	11	Monday	240	226140	0
Cardiology	9	Monday	10	Monday	10	Monday	12	Monday	1200	2460	6120
Nephrology	10	Monday	10	Monday	11	Monday	12	Monday	1740	3780	1980
Urology	11	Monday	12	Monday	12	Tuesday	12	Tuesday	2640	87480	0

Figure 2. Sample from the data set used in numerical prediction with transformed variables and computed turnaround times (TAT)

For the parameters given as; N = population size, μ = distribution ratio, E = margin of error, σ = standard deviation of the population and $Z_{\alpha/2}$ = upper critical value for the standard normal distribution within the chosen confidence level, the sample size n can be calculated as follows:

$$n = \frac{\mu(1-\mu)}{\left(\frac{E}{Z_{\alpha/2}}\right)^2} \quad (11)$$

Since there is a limited population, the formula given in (11) can be evaluated as below by using the correction factor:

$$n = \frac{N\mu(1-\mu)}{\mu(1-\mu) + (N-1)\left(\frac{E}{Z_{\alpha/2}}\right)^2} \quad (12)$$

The new sample size for the first subsample data set was chosen as follows:

- accepted margin of error (confidence interval) = 2%,
- confidence level = 98%

Similarly, the sample size for the second data set was chosen as follows:

- accepted margin of error (confidence interval) = 1%,
- confidence level = 95%

Using these statistical sampling parameters and the equations given in (11) and (12), the sample sizes for the first and second subsample data sets were set to be 3315 and 9100, respectively, which suffices the minimum

sample size limits regarding the original data set size and chosen sampling parameters. Hence, these two subsample data sets were derived with these record sizes by random selection and sampling from the original data set.

One of the performance evaluation measures for numerical prediction that is used in this study is coefficient of determination (R^2), which is used as an evaluation metric [31-34]. If the predicted values among the test / validation instances derived by a numerical prediction algorithm are denoted as $p_1, p_2,.. \ p_n$ and the actual values are denoted as $a_1, a_2,.. \ a_n$, then the coefficient of determination R^2 can be calculated as follows:

$$R^2 = 1 - \frac{SSE}{SST}$$

$$where \quad (13)$$

$$SSE = \sum_{i=1}^{n} (p_i - a_i)^2$$

$$SST = \sum_{i=1}^{n} (a_i - \overline{a})^2$$

It should be noted that, in the equation (13), n denotes the total number of samples, and $\bar{a}$ denotes the arithmetic mean of actual values respectively [35].

Another performance measure for numerical prediction that was used in this study was root mean squared error (RMSE), which is also defined as the "standard error of the estimate" [36]. Given that, the predicted values are $p_1, p_2,.. \ p_n$ and the actual values are $a_1, a_2,.. \ a_n$ within a test / validation data set, RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{(p_1 - a_1)^2 + .. + (p_n - a_n)^2}{n}} \quad (14)$$

It is known that mean squared error and root mean squared error metrics have a tendency to magnify the impact of outliers. On the other hand, it is also known that mean absolute error (MAE) measure does not tend to exaggerate the error values caused by outliers [19]. Since, we had several outliers in the data used in this study, and since it was strictly necessary for us not to discard the records having outlier values, we also used MAE as another prediction performance measurement. MAE calculation is given in (15).

$$MAE = \frac{|p_1 - a_1| + \cdots |p_n - a_n|}{n} \quad (15)$$

The descriptive statistics for the "TAT-target" attribute within both of the data sets are given in Table 1. It could be seen from Table 1 that the minimum "TAT-target" value is observed to be zero for both of the subsample data sets, in other words, the minimum duration between "Date and time of the delivery of the report to the patient" and "Date and time of report for test results" is 0 seconds. This might be considered as a controversial issue but the hospital management claimed that in some ordinary cases, the two stages within "TAT-target" might occur just at the same time. Hence, these records and values were not discarded in the study and accepted as accurate input data. It could also be observed from Table 1 that for the maximum "TAT-target" value for subsample data set 1 is 169800 seconds and maximum "TAT-target" value for subsample data set 2 is 288300 seconds. These values show that the duration between "Date and time of the delivery of the report to the patient" and "Date and time of report for test results" might even go up to 4805 minutes (approximately equivalent to 80 hours) in some extreme cases.

Table 1. Descriptive statistics for "TAT-target" attribute for two data sets

	Data set 1 (3315 instances)	Data set 2 (9100 instances)
Mean	1033.69 sec	1086.24 sec
Standard deviation	7503.95 sec	8329.07 sec
Minimum	0 sec	0 sec
Maximum	169800 sec	288300 sec

4. DESIGN AND IMPLEMENTATION

We designed and implemented a multilayer perceptron model, which was similar to the one described in "Artificial neural networks" section. We developed our multilayer perceptron model using Microsoft Visual C# programming language. The multilayer perceptron's training parameters that were used in both of the data sets in the tool were as follows: momentum: 0.6, learning rate: 0.32, number of iterations (epochs): 1000. The architecture of the multilayer perceptron in this study was designed and implemented with an input layer with 45 nodes, one output layer with one node, two hidden layers where the first hidden layer had 28 nodes and the second hidden layer was composed of two nodes. The input layer of the MLP was composed of 45 nodes because there were 11 feature attributes in the data sets. Among these eleven feature attributes, six of them were composed of numeric attributes that necessitated six input nodes. For the remaining five features, four of them were related with "day of week" that have categorical (nominal values)

where each has seven different values and this made up a total of 28 additional input nodes eventually. The department name was also another nominal attribute that provides with eleven different values, which required the creation of eleven additional input nodes. Hence, 45 nodes were implemented and used in the input layer of the ANN model in this study.

All the nodes were fully connected in the ANN model, which can be described as a multilayer perceptron with feed-forward learning using backpropagation error with gradient descent. We designed and used a different MLP in this study with two hidden layers, in contrast to most ordinary MLP models having only one hidden layer.

It should also be noted that hyperbolic tangent function is used as the non-linear activation function in the first hidden layer, and sigmoid function is used as the non-linear activation function in the second hidden layer in our MLP model. The total number of weights used in the multilayer perceptron model was 1318 which can be calculated as ((45 x 28) + (28 x 2) + (2 x 1)). The initial weights for all of the connections and bias values within all layers were randomly set between minus one and plus one before the initiation of training phase in this specific MLP. A simple representative diagram of our MLP architecture is given in Fig. 3.

Weka version 3.7.12 was used for conducting the experiments with alternative machine learning algorithms that were feasible for this study. Weka [37] is an open source data mining and machine learning software that is developed in Java programming language. The algorithms in Weka were comparatively tested with the ANN model. 10-fold cross-validation methodology was used for all of the tests and experiments. The experiments were carried out with two different data sets, where one of them had 3315 instances and the other one had 9100 instances. Cross-validation is one of the most preferred reliable and accurate statistical techniques when the data set has to be separated into train and test data sets [16]. In "k-fold cross-validation", the initial data set is partitioned into *k* "mutually exclusive subsets" or "folds" [28]. For instance, if *k* is set to three, then it becomes a threefold cross-validation where the initial data is split into three equal partitions; in each round, two folds are used for training and the other is used for testing and this procedure is repeated three times so that every instance in the initial data set will have been used exactly once for testing [19, 28]. It has been proven that ten is accepted to be the right number of folds to get the best error estimate [19]. Thus, ten-fold cross-validation was chosen for the training and testing methodology in this study.

Coefficient of determination (R^2), mean absolute error (MAE), and root mean squared error (RMSE) were used to measure and compare the prediction performances of the algorithms obtained by ten-fold cross-validation results. The algorithms that were provided in Weka and used in this study are described shortly as below.

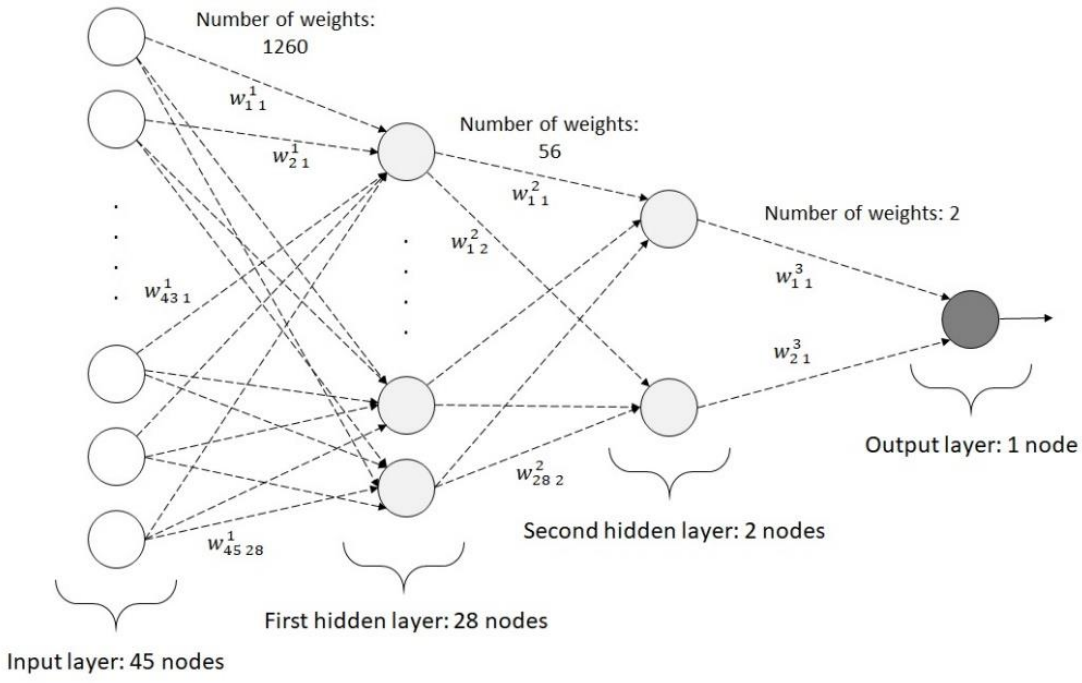

Figure 3. Diagram of our MLP model that is used in this study

- k-nearest neighbors (k-NN): It is a type of instance-based learner algorithm [38].
- KStar: This algorithm uses a similarity function with an entropy-based distance metric to make the comparisons between the records and it is a type of instance-based learning algorithm [39].
- Linear regression: Multiple linear regression with two or more estimator variables and for its model selection, Akaike criterion is used [40].
- Isotonic regression: It chooses the attribute with the lowest squared error and derives its isotonic regression model based on this decision [19].
- Partial least squares (PLS) regression: It is a type of regression model, which calculates derived directions that, as well as having high variance, they are strongly correlated with the class [19].
- Least median of squares (LeastMedSq) regression: It uses multiple linear regression to derive the predictions. Least squared regression functions are produced by using random subsamples of the data set [41].
- M5 Model Tree (M5P): It is a special type of decision tree that can be used for numerical predictions. It is based on "M5Base" algorithm, which generates "M5" model trees and rules [42, 43].
- Decision Stump: It is a simple decision tree algorithm with only one level. It is usually used with a boosting algorithm. For numerical predictions, it uses regression based on mean-squared error values [44].
- Radial Basis Function (RBF) network: It is an artificial neural network model that implements normalized Gaussian radial basis functions [45] for activation processes.
- Support Vector Machine (SVM) for regression with Sequential Minimal Optimization (SMO): This is a type of support vector machine that uses polynomial kernel function. It implements John Platt's sequential minimal optimization algorithm for training [46] and implements regression for numerical prediction [47].
- Single layer perceptron: A simple ANN does not have a hidden layer. It is composed of input nodes and an output node, where the transfer (activation) function is a linear signum function [27].
- Multilayer Perceptron: Feed-forward ANN algorithm that uses backpropagation with the gradient descent approach for error calculation and weight updates. The learning rate, momentum, number of iterations, number of hidden layers, and the number of nodes can be flexibly changed. It can be used for both classification and numerical prediction [19].

The parameter settings for some of these machine learning algorithms (k-NN, M5P, single layer perceptron, and multilayer perceptron) are given in Table 2, where they were set to alternative values according to the observations during the experiments. All the other algorithms' parameters were set to default values in Weka. It should be noted that some of the other linear numerical prediction algorithms such as simple linear regression, pace regression and some nonlinear regression algorithms were not included in this article because the performance results obtained by those algorithms were much lower and more inaccurate in terms of RMSE, MAE and R^2.

Table 2. Parameter settings of some of the algorithms

Algorithm name	**Parameter settings**
k-NN	k=2 distance metric: Minkowski distance (order=4) weighted distance not used
M5 Model Tree (M5P)	minimum number of instances at leaf nodes: 8
Single layer perceptron	transfer function: signum learning rate: 0.9 learning rate with linear decay number of iterations: 1000
Multilayer perceptron	number of hidden layers: 1 number of nodes in the hidden layer: 23 learning rate: 0.3 momentum: 0.2 number of iterations: 500 decay in learning rate: no activation function: sigmoid

5. RESULTS AND DISCUSSION

All of the tests and execution of both the algorithms in Weka and our ANN model were conducted on the same hardware platform. The results were obtained by using ten-fold cross-validation where the performance evaluation measures are given in the tables Table 3 and Table 4. In these tables, twelve machine learning algorithms were executed and tested by the Weka software and the results for the last one were obtained with ANN multilayer perceptron model that was implemented and developed by the authors of this paper. There are several options in Weka for changing the algorithms' parameters and they were tested with different parameters as well as with their default settings. In most of the cases, it was observed that changing the parameters degraded the performance of the algorithms so only the results obtained with their default configurations are included in the tables below.

It was observed that changing the parameters improved the estimation performance of only k-nearest neighbors (k-NN) and M5 Model decision tree (M5P) algorithms so the results obtained by those are included in the tables. The best results for k-NN algorithm were obtained by the following parameters; k was set to two (two nearest neighbors) and the distance metric was set to Minkowski distance with an order of four. The best results for M5P were obtained when the minimum number of instances at leaf nodes were set to eight.

The comparative results show that the ANN model (feed-forward multilayer perceptron neural network with backpropagation and gradient descent) has been observed as the most successful algorithm among both of the data sets. It provided the lowest RMSE and MAE values and highest R^2 values when tested independently with both of the data sets. Since, MAE and RMSE are performance evaluation measure for the algorithms' numerical predictions, ANN always provided the best and the most accurate predictions with the lowest RMSE and MAE values.

It is known that the coefficient of determination, which is also named as "R squared" and denoted as R^2, can be used as a statistical measure of how well observed outcomes are replicated by the model [20, 35]. It is also known that R^2 is accepted to be representing the best performance when it is close to or equal to one. In this study, since R^2 was used for the correlation between the algorithm's predictions and original values for "TAT-target", ANN significantly outperformed all the other algorithms with highly reliable R^2 values. However, when the execution times of the algorithms are compared, it could be seen that SVM gives the worst execution times, and our ANN model has the second worst performance within both data sets.
We conducted another prediction performance analysis with our ANN model, since it was shown to be the most successful algorithm among both of the data sets. The original entire dataset was randomly subsampled into four different test sets, which two of these were composed of 9100 instances, and the other two had 3315 instances. It should also be noted that all of the instances in these new test data sets were

Table 3. Comparative prediction performance results observed by using 10-fold cross-validation for the first subsample data set with 3315 instances

Algorithm name	**R^2**	**RMSE[a]**	**RMSE[b]**	**MAE[c]**	**MAE[d]**	**Execution time (seconds)**
k-NN	0.5811	0.0286	4832.12	0.0046	797.53	0.27
KStar	0.4064	0.0346	8496.58	0.0044	1066.41	34.05
Linear regression	0.0194	0.0454	7705.10	0.0101	1709.91	0.36
Isotonic regression	0.1461	0.0409	6940.28	0.0077	1314.24	0.77
Partial least squares regression	0.3683	0.0354	6047.16	0.0120	2080.91	0.49
LeastMedSq regression	0.0005	0.0445	7562.87	0.0059	1003.09	3.59
M5 Model Tree (M5P)	0.4375	0.0337	5635.62	0.0051	894.10	0.65
Decision Stump	0.1493	0.0408	6927.58	0.0079	1349.18	0.25
Radial Basis Function network	0.0012	0.0442	7496.87	0.0086	1461.58	0.79
SVM with SMO	0.1176	0.0741	12581.29	0.0061	1032.56	2934.10
Single layer perceptron	0.0014	0.9992	7573.43	0.9982	1033.26	8.47
Multilayer perceptron	0.7850	0.0215	3490.87	0.0049	834.54	308.43
ANN (our model)	**0.8832**	**0.0201**	**2362.07**	**0.0032**	**726.71**	512.78

a Root mean squared error of min-max normalized "TAT-target" values.
b Root mean squared error of real "TAT-target" values.
c Mean absolute error of min-max normalized "TAT-target" values.
d Mean absolute error of real "TAT-target" values.

Table 4. Comparative prediction performance results observed by using 10-fold cross-validation for the second subsample data set with 9100 instances

Algorithm name	**R^2**	**RMSE[a]**	**RMSE[b]**	**MAE[c]**	**MAE[d]**	**Execution time (seconds)**
k-NN	0.5226	0.0200	5266.50	0.0024	635.44	0.89
KStar	0.4177	0.0221	8959.62	0.0028	1017.94	198.06
Linear regression	0.0620	0.0281	8096.63	0.0053	1531.55	1.35
Isotonic regression	0.1445	0.0268	7716.90	0.0049	1404.35	12.94
Partial least squares regression	0.1313	0.0276	7876.51	0.0063	1807.57	3.32
LeastMedSq regression	0.0033	0.0291	8389.94	0.0037	1052.80	54.63
M5 Model Tree (M5P)	0.4651	0.0222	6090.42	0.0028	793.48	9.74
Decision Stump	0.1614	0.0265	7628.02	0.0050	1427.99	1.56
Radial Basis Function network	0.0023	0.0289	8326.69	0.0054	1571.38	2.56
SVM with SMO	0.0054	0.0410	11828.90	0.0031	883.89	34178.21
Single layer perceptron	0.0004	0.9998	8399.10	0.9994	1085.81	23.79
Multilayer perceptron	0.8462	0.0124	3292.08	0.0023	635.80	768.50
ANN (our model)	**0.9012**	**0.0097**	**3004.57**	**0.0019**	**600.77**	1854.95

a Root mean squared error of min-max normalized "TAT-target" values.
b Root mean squared error of real "TAT-target" values.
c Mean absolute error of min-max normalized "TAT-target" values.
d Mean absolute error of real "TAT-target" values.

especially chosen to be completely different from the previous two data sets that were used in 10-fold cross-validation experiments. The prediction performance analysis was carried out as follows: we trained our ANN model using the same data set that was mentioned in Table 4, and then we used two new test sets having 9100 instances and obtained the performance values. Similarly, we also trained our ANN model using the same data set that was mentioned in Table 3, and then we used two new test sets having 3315 instances and obtained the performance values. It can be seen that our ANN model achieved successful results in all of these four tests, which are given in Table 5.

Table 5. Comparative prediction performance results of our MLP model among four different test sets

Data sets	**R^2**	**RMSE[a]**	**RMSE[b]**	**MAE[c]**	**MAE[d]**
Test set – 1 (9100 instances)	0.8944	0.0106	3002.78	0.0022	612.55
Test set – 2 (9100 instances)	0.9168	0.0091	2995.42	0.0014	589.47
Test set – 3 (3315 instances)	0.8652	0.0284	2457.35	0.0038	790.18
Test set – 4 (3315 instances)	0.8758	0.0249	2511.46	0.0035	758.32

a Root mean squared error of min-max normalized "TAT-target" values.
b Root mean squared error of real "TAT-target" values.
c Mean absolute error of min-max normalized "TAT-target" values.
d Mean absolute error of real "TAT-target" values.

Table 6. Pearson correlations between "TAT-target" and independent variables

Independent variable name	**Pearson correlation**
Patient registration	-0.015
Request for a physician service	-0.022
Specimen collection	-0.024
Sample receipt	-0.021
Report of test results	0.049
Delivery of report to patient	-0.046
Patient's visit to physician	-0.038
Duration between "Laboratory sample receipt" and "Specimen collection"	0.016
Duration between "Report for test results" and "Delivery of report to patient	0.142

We also examined the statistical independence between the "TAT-target" variable and all of the other feature variables. The correlations between the dependent variable "TAT-target" and the other independent variables are analyzed by means of Pearson correlations [35]. It was observed that no significant correlations between the target and other attributes exist, in other words, Pearson correlation coefficient values were close to zero, which is also given in Table 6.

6. CONCLUSIONS

The results show that a feedforward artificial neural network with a multilayer perceptron architecture using backpropagation with gradient descent could be a reliable and promising alternative for numerical prediction of TAT amongst different medical services in hospital and other medical institutions that might be used by the hospital management or by the decision makers in healthcare systems. The results show that not only the artificial neural network model surpasses all of the other numerical prediction algorithms within different data sets in terms of RMSE, MAE and R^2, but also provides a high correlation between the original and estimated TAT values regarding the fact that R^2 was observed to be greater than 0.9 or very close to 0.9. In addition, our ANN model also achieved successful and reliable prediction performance results among four different test data sets that were particularly not used in the previous validation and training data sets.

On the other hand, it has been observed that the execution times of our ANN model gave the second worst results among the algorithms used in this study, which might be concluded as the only drawback of our ANN model. This can be related to two basic reasons. One of these reasons is that most of the ANN algorithms having a multilayer perceptron architecture are generally known to run slower than other linear or non-linear algorithms, especially within the training phase. The weight updates and error calculations during the training phase in MLP are shown to be more time-consuming than other alternative algorithms. The second reason is related with the specific MLP model used in this study. The MLP model had two hidden layers, where the first hidden layer was constructed with twenty-eight nodes and the second hidden layer was constructed with two nodes. Hence, this approach caused the MLP model to be implemented in a more complex architecture than ordinary multilayer perceptron models with a single hidden layer and much less number of nodes and connections. However, this architectural complexity, which also brought forth a high computational complexity, was a necessary requirement. The prediction accuracy was observed to degrade significantly when the ANN architecture was tested with less number of nodes and a single hidden layer. It was observed that the best results were only achieved by the unique complex MLP model that was finally accomplished in this study. It is planned to design and develop a new version of this MLP model as a future study, which could overcome the computational complexity and eventually decrease the execution time.

REFERENCES

[1] R. Bernardi, P. Constantinides, J. Nandhakumar, "Challenging Dominant Frames in Policies for IS Innovation in Healthcare Through Rhetorical Strategies", *Journal of the Association for Information Systems*, 18(2), 81-112, 2017.

[2] S. Dilek, S. Özdemir, "Wireless Sensor Networks in Healthcare", *International Journal of Informatics Technologies*, 7(2), 7-19, 2014.

[3] C. He, X. Fan, Y. Li, "Toward Ubiquitous Healthcare Services with a Novel Efficient Cloud Platform", *IEEE Transactions on Biomedical Engineering*, 60(1), 230-234, 2013.

[4] A. R. Lyon, J. K. Wasse, K. Ludwig, M. Zachry, E. J. Bruns, J. Unutzer, E., McCauley, "The Contextualized Technology Adaptation Process (CTAP): Optimizing Health Information Technology to Improve Mental Health Systems. Administration and Policy", *Mental Health and Mental Health Services Research*, 43(3), 394-409, 2016.

[5] X. W. Ng, W. Y. Chung, "VLC-Based Medical Healthcare Information System", *Biomedical Engineering: Applications, Basis and Communications*, 24(2), 155-163, 2012.

[6] H. M. Söderholm, D. H. Sonnenwald, "Visioning Future Emergency Healthcare Collaboration: Perspectives from Large and Small Medical Centers", *Journal of the American Society for Information Science and Technology*, 61(9), 1808–1823, 2010.

[7] A. Demir, E. İ. Tatlı, "Security Analysis of Medical Devices within Wireless Body Area Networks and Mobile Health Applications", *International Journal of Informatics Technologies*, 11(1), 1-8, 2018.

[8] B. Goswami, B. Singh, R. Chawla, V. K. Gupta, V. Mallika, "Turnaround Time (TAT) as a Benchmark of Laboratory Performance", *Indian Journal of Clinical Biochemistry*, 25(4), 376-379, 2010.

[9] İ. H. Köksal, B. Türkoğlu, M. Eminağaoğlu, "An Adaptive Network-Based Fuzzy Inference System for Estimating the Duration of Medical Services: A Case Study", **IEEE 10th International Conference on Application of Information and Communication Technologies (AICT)**, Baku, Azerbaijan, 801-806, 2016.

[10] M. Scagliarini, M. Apreda, U. Wienand, G. Valpiani, "Monitoring Operating Room Turnaround Time: A Retrospective Analysis", *International Journal of Health Care Quality Assurance*, 29(3), 351-359, 2016.

[11] D. Sinreich, Y. Marmor, "Ways to Reduce Patient Turnaround Time and Improve Service Quality in Emergency Departments", *Journal of Health Organization and Management*, 19(2), 88-105, 2005.

[12] K. A. Willoughby, B. T. B. Chan, M. Strenger, "Achieving Wait Time Reduction in the Emergency Department", *Leadership in Health Services*, 23(4), 304-319, 2010.

[13] B. Breil, F. Fritz, V. Thiemann, M. Dugas, "Mapping Turnaround Times (TAT) to A Generic Timeline: A Systematic Review of TAT Definitions in Clinical Domains", *BMC Medical Informatics and Decision Making*, 11(34), 1-12, 2011.

[14] M. Fieri, N. F. Ranney, E. B. Schroeder, E. M. Van Aken, A. H. Stone, "Analysis and Improvement of Patient Turnaround Time in an Emergency Department", **IEEE Systems and Information Engineering Design Symposium**, University of Virginia, Charlottesville, USA, 239-244, 2010.

[15] A. B. Storrow, C. Zhou, G. Gaddis, J. H. Han, K. Miller, D. Klubert, A. Laidig, D. Aronsky, "Decreasing Lab Turnaround Time Improves Emergency Department Throughput and Decreases Emergency Medical Services Diversion: A Simulation Model", *Academic Emergency Medicine*, 15(11), 1130-1135, 2008.

[16] C. Hand, H. Mannila, P. Smyth, **Principles of Data Mining**, MIT Press, London, 2001.

[17] T. Hastie, R. Tibshirani, J. Friedman, **The Elements of Statistical Learning, Data Mining, Inference and Prediction**, 2nd ed., Springer, New York, 2009.

[18] N. Nedjah, M. Luiza, J. Kacprzyk, **Innovative Applications in Data Mining,** Springer-Verlag, Berlin, 2009.

[19] I. H. Witten, E. Frank, M. A. Hall, **Data Mining: Practical Machine Learning Tools and Techniques**, 3rd ed., The Morgan Kaufmann Series in Data Management Systems, 2011.

[20] E. Alpaydın, **Introduction to Machine Learning**, 2nd ed., MIT Press, 2010.

[21] S. Özden, A. Öztürk, "Electricity Energy Demand Forecasting for an Industrial Region (Ivedik) by using Artificial Neural Network and Time Series", *Bilişim Teknolojileri Dergisi*, 11(3), 255-261, 2018.

[22] E. S. Olivas, J. D. M. Guerrero, M. M. Sober, J. R. M. Benedito, A. J. S. Lopez, **Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques,** IGI Global, 2009.

[23] J. Reyes, A. Morales-Esteban, F. Martinez-Alvarez, "Neural Networks to Predict Earthquakes in Chile", *Applied Soft Computing*, 13(2), 1314-1328, 2013.

[24] E. Çelik, O. Çavuşoğlu, H. Gürün, N. Öztürk, "Estimation of the Clearance Effect in the Blanking Process of CuZn30 Sheet Metal Using Neural Network−A Comparative Study", *Bilişim Teknolojileri Dergisi*, 11(2), 187-193, 2018.

[25] Ö. Tanidir, O.B. Tör, "Accuracy of ANN Based Day-Ahead Load Forecasting in Turkish Power System: Degrading and Improving Factors", *Neural Network World*, 25(4), 443–456, 2015.

[26] S. Senan, "A Neural Net-Based Approach for CPU Utilization", *Bilişim Teknolojileri Dergisi,* 10(3), 263-272, 2017.

[27] S. Haykin, **Neural Networks and Learning Machines**, 3rd ed., Pearson Education, Inc., New Jersey, 2009.

[28] J. Han, M. Kamber, **Data Mining: Concepts and Techniques**, 2nd ed., Morgan Kaufmann Publishers, San Francisco, 2006.

[29] A. Graves, **Supervised Sequence Labelling with Recurrent Neural Networks,** Springer-Verlag, Berlin, 2012.

[30] T. Dasu, T. Johnson, **Exploratory Data Mining and Data Cleaning,** John Wiley & Sons Inc., New Jersey, 2003.

[31] H. Özkişi, M. Topaloğlu, "The Estimation of the Photovoltaic Cell Productivity with the Use of Artificial Neural Network", *Bilişim Teknolojileri Dergisi*, 10(3), 247-253, 2017.

[32] C. D. Ravinesh, M. Şahin, "Application of the Extreme Learning Machine Algorithm for the Prediction of Monthly Effective Drought Index in Eastern Australia", *Applied Soft Computing*, 15(3), 512-525, 2015.

[33] I-C. Yeh, C. Lien, "The Comparisons of Data Mining Techniques for the Predictive Accuracy of Probability of Default of Credit Card Clients", *Expert Systems with Applications*, 36, 2473-2480, 2009.

[34] M. H., Calp, An Estimation of Personnel Food Demand Quantity for Businesses by Using Artificial Neural Networks, *Journal of Polytechnic*, DOI: 10.2339/politeknik.444380, 2019. (In Press)

[35] D. T. Larose, **Data Mining Methods and Models,** John Wiley & Sons Inc., New Jersey, 2006.

[36] D. T. Larose, **Discovering Knowledge in Data - An Introduction to Data Mining**, John Wiley & Sons Inc., New Jersey, 2005.

[37] Internet: Machine Learning Group at the University of Waikato, http://www.ss.waikato.ac.nz/ml/Weka/, 03.05.2018.

[38] D. W. Aha, D. Kibler, M. K. Albert, "Instance-Based Learning Algorithms", *Machine Learning*, 6(1), 37-66, 1991.

[39] J. G. Cleary, L. E. Trigg, "K*: An Instance-Based Learner Using an Entropic Distance Measure", **12th International Conference on Machine Learning**, Tahoe City, California, USA, 108-114, 1995.

[40] H. Akaike, "Likelihood of a Model and Information Criteria", *Journal of Econometrics*, 16(1), 3-14, 1981.

[41] P. J. Rousseeuw, A. M. Leroy, **Robust Regression and Outlier Detectio**n, John Wiley & Sons Inc., 1987.

[42] R. J. Quinlan, "Learning with Continuous Classes", **5th Australian Joint Conference on Artificial Intelligence**, Singapore, 343-348, 1992.

[43] Y. Wang, I. H., Witten, "Induction of Model Trees for Predicting Continuous Classes", **9th European Conference on Machine Learning**, Prague, Czech Republic, 1997.

[44] W. Iba, P. Langley, "Induction of One-Level Decision Trees", **9th International Conference on Machine Learning**, Aberdeen, Scotland, 233–240, 1992.

[45] M. D. Buhmann, **Radial Basis Functions: Theory and Implementations**, Cambridge University Press, UK, 2003.

[46] J. Platt, "Fast Training of Support Vector Machines Using Sequential Minimal Optimization", **Advances in Kernel Methods - Support Vector Learning**, editors: B. Schoelkopf, C. Burges, A. Smola, MIT Press, 1998.

[47] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, K.R.K. Murthy, "Improvements to the SMO Algorithm for SVM Regression", *IEEE Transactions on Neural Networks*, 11(5), 1188-1193, 2000.