VOL.4NO.2September 2015 - ISSN 2233 – 1859

UOIuBIH
ORSinBIH
Operations Research Society in Bosnia and Herzegovina

Southeast Europe Journal of Soft Computing

Available online:http://scjournal.ius.edu.ba/

Application Of Machine Learning In Healthcare: Analysis On MHEALTH Dataset

Muhammed Ali Kutlay
Sadina Gagula-Palalic

International University of Sarajevo, Faculty of Engineering and Natural Sciences, HrasnickaCesta 15, Ilidža71210 Sarajevo, Bosnia and Herzegovina

Article Info

Article history:
Received 17 Jul.2015
Received in revised form 17 Aug. 2015

*Keywords:*healthcare, machine learning techniques, movements, vital signals, diagnosis, mobile health

Abstract

The healthcare services in developed and developing countries are critically important. The use of machine learning techniques in healthcare industry has a vital importance and increases rapidly. The corporations in healthcare sector need to take advantage of the machine learning techniques to obtain valuable data that could later be used to diagnose diseases at much earlier stages. In this study, a research is conducted with the purpose of discovering further use of the machine learning techniques in healthcare sector. Research was conducted by analyzing a well-established dataset called MHEALTH, comprising body motion and vital signs recordings for ten volunteers of diverse profile while performing 12 physical activities. Dataset was analyzed using certain classification algorithms such as Multilayer Perceptron and Support Vector Machine, then results from these algorithms were compared to determine the most utile algorithm for analyzing such dataset. Study aims to determine irregularities using data from body motion and vital signs of volunteers, then these findings can be used either to diagnose particular diseases before they occur and avoid them. Results can also be used to monitor movements of ill or elderly people and observe whether they are doing any prohibited movements that would lead them to injuries or further illnesses.

1. INTRODUCTION

The health care is one of the very important assets in the society. However, due to rapid grow, societies' needs for healthcare exceeds the supplies of affordable and accessible care. As need for healthcare grows, providing a sufficient healthcare to the society is the number one priority of the foundations in healthcare sector. The structure of the health sector varies depending on the country's population, cultural development, natural resources, political and economic system. Increase of importance given to healthcare and the quality level of healthcare, increases competition among health organizations and provides an important contribution to the development of the sector.

Beyond any doubt, one of the most important factors that affects healthcare sector is technology. Despite of quick growth in societies and in societies' need for healthcare, todays' advancing technology may be one of the most important factors that may respond to the need of health care services in societies. Luckily, today we have an advanced technology in healthcare systems that can help making decisions based on gathered information. This functionality of the technology in healthcare systems is already being used to gather information about any symptom that a patient has, to diagnose particular diseases before they occur on the patient, and to prevent any of these diseases by taking precautions.

With the help of these technologies, many patients have already been rescued from various deathly diseases.

Machine learning is a process of training machines to give them ability to make decisions based on certain algorithms. There are variety of already defined machine learning algorithms. Each algorithm processes data in a different way with the purpose of training machine and to give them ability of making decisions.

In this study, we analyzed a well-established dataset called MHEALTH using two different machine learning algorithms called Multilayer Perceptron and Support Vector Machine with the purpose of classifying body movements of a volunteer. We aimed to reveal differences and similarities of these two algorithms, but the main aim was to determine which algorithm is more precise, and can be used to classify body movements of a person based on vital signs recordings. The study can be used to diagnose particular diseases and to monitor movements of ill or elderly people and observe whether they are doing any prohibited movements that would lead them to injuries or further illnesses.

1.MULTI-LAYER PERCEPTRON

A multi-layer perceptron is a feed-forward neural network, consisting of a number of units (neurons) which are connected by weighted links [1]. Multi-layer Perceptron is architecturally nonlinear and has a structure that consists of many neurons which have activation function are hierarchically connected to each other. Multi-layer perceptron uses a learning system that is called back-propagation.

The perceptron consist of weights (including bias), the summation processor, and an activation function.

A perceptron takes a weighted sum of inputs and outputs:

- 1 if the sum is > some adjustable threshold value (theta)
- 0 Otherwise

$W1.X1 + W2.X2+ \ldots\ldots + Wn.Xn > 0 \to 1$ (output)

$W1.X1 + W2.X2+ \ldots\ldots + Wn.Xn < 0 \to 0$ (output)

The inputs and connection weights are typically real values
How it works?

The inputs values are presented to the perceptron, and if the predicted output is the same as the desired output then the performance is considered satisfactory and no changes to the weights are made.

However if the output does not match the desired output then the weights need be changed to reduce the error
Perceptron weight adjustment;

$\Delta w = \eta \times d \times x$
d = predicted output
η – Learning rate, usually less than 1
x – Input data

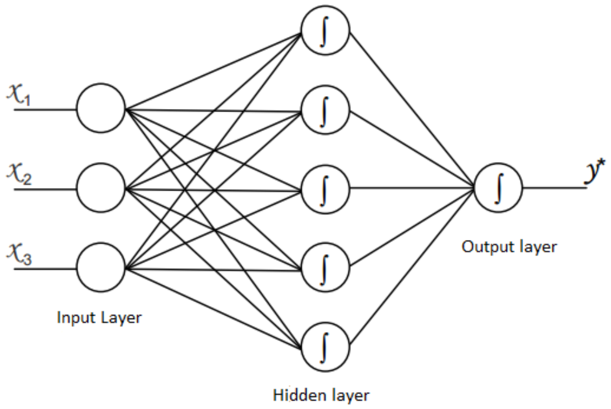


Fig. 1 General architecture of MLP [2].

Input layer: Accepts the data vector or pattern.
Hidden layers: One or more layers. They accept the output from the previous layer, weight them, and pass through a, normally, non-linear activation function.
Output layer: Takes the outputs from the final hidden layer, weights them, and possibly pass through an output nonlinearity, to produce the target values [3].

1.1. BACKPROPAGATION

The popularity of on-line learning for the supervised training of multilayer perceptron has been further enhanced by the development of the Backpropagation algorithm [4]. In back-propagation algorithm, outputs are compared according to input values whether output weights are close to desired weights and differences between those weights are evaluated as mistake. Also, correction is made to a particular point and by the end of this iteration network's learning will be finished successfully. The algorithm is named as Back propagationbecause, it tries to minimize faults backward, from output to input. Backpropagation learning theorem is used to re-calculate the weights in each layer according to existing outputs [5].

1.2. CROSS-VALIDATION

In machine learning, to test success of the applied methods, dataset is divided into two groups called as training set and test set. In K-fold cross-validation technique, a K value must be determined first. As also Weka suggests, the most preferred K value is 10.

For K=10, we divide our dataset into 10 equal folds. Let's assume we have 1000 data in our dataset. In this case, we divide our dataset equally into 10 folds that each one holds 100 data in it. After fragmentation process is done, k-fold cross-validation system starts working. Firstly, some of the folds are selected for testing and rest of them are used for training. Usually, 33% is used for testing and 66% is used for training. Shortly, system works in a logic that learning will be provided using all possible combinations from cross-validation. Performance of the algorithm is calculated with the formula:

$$E = \frac{\sum_{i=1}^{k} n_i}{m}$$

Let $E_1, \dots, Et$ be the accuracy estimates obtained in t runs.

$$e = \frac{\sum_{j=1}^{t} E_j}{t}$$

$$\sigma^2 = V = \frac{\sum_{j=1}^{t} (E_j - e)^2}{t-1}$$

Then, the estimate for the algorithm performance is an error of e with standard-deviation of σ.

2. SUPPORT VECTOR MACHINE

Support Vector Machines are statistical classifiers. It uses Lagrange multipliers to represent the support vectors. Only the data with non-zero Lagrange multipliers will be the support vectors. SVMs uses quadratic programming to find the optimal solution of Lagrange multipliers. SVMs also restrict the choice of kernel that the quadratic programming is a convex problem. Therefore, it guarantees global optimization with the corresponding kernel [6]. Basically, the support vector machine is a binary learning machine with some highly elegant properties. SVM runs according to the principle of structural risk minimization and based on convex optimization [5]. Aforementioned algorithm does not require any distribution function information associated with the data. This is the reason it is a learning algorithm independent from distribution. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on [7].

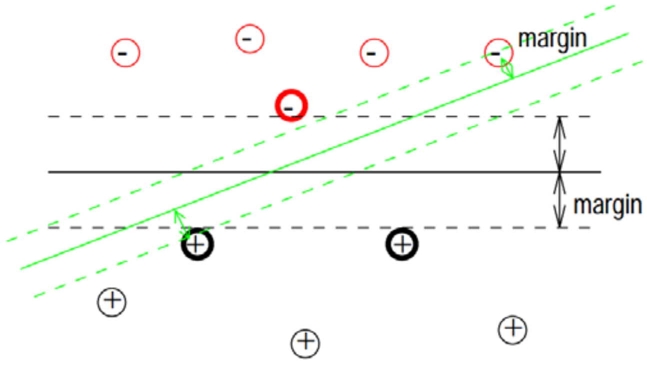


Fig. 2 A linear SVM. The circled data points are the support vectors – the examples that are closest to the decision boundary. They determine the margin with which the two classes are separated[7].

3. MHEALTH DATASET

MHEALTH dataset is a well-established dataset, comprising body motion and vital signs recordings for ten volunteers of diverse profile while performing 12 physical activities. Dataset contains 161.280 lines of data representing physical body motion of a subject. Researchers used sensors on chest, right wrist, and left ankle in subject’s body to measure the body motion, namely purpose is to calculate determine acceleration, rate of turn, and magnetic field orientation that is experienced in different part of the body. Also, the sensors that are placed on the chest provide 2-lead ECG measurement that can be used for heart monitoring.

3.1. EXPERIMENTAL SETUP

The collected dataset contains body motion and vital signs recordings for ten volunteers of diverse profile while performing 12 physical activities. Shimmer2 [4] wearable sensors were used for the recordings. The sensors were respectively placed on the subject's chest, right wrist and left ankle and attached by using elastic straps. The use of multiple sensors permits us to measure the motion experienced by diverse body parts, namely, the acceleration, the rate of turn and the magnetic field orientation, thus better capturing the body dynamics. The sensor positioned on the chest also provides 2-lead ECG measurements which are not used for the development of the recognition model but rather collected for future work purposes. This information can be used, for example, for basic heart monitoring, checking for various arrhythmias or looking at the effects of exercise on the ECG. All sensing modalities are recorded at a sampling rate of 50 Hz, which is considered sufficient for capturing human activity. Each session was recorded using a video camera. This dataset is found to generalize to common activities of the daily living, given the diversity of body parts involved in each one (e.g., frontal elevation of arms vs. knees bending), the intensity of the actions (e.g., cycling vs. sitting and relaxing) and their execution speed or dynamicity (e.g., running vs. standing still). The activities were collected in an out-of-lab environment with no constraints on the way these must be executed, with the exception that the subject should try their best when executing them.

3.2. ACTIVITY SET

The activity set is listed in the following:

L1: Standing still (1 min)

L2: Sitting and relaxing (1 min)

L3: Lying down (1 min)

L4: Walking (1 min)

L5: Climbing stairs (1 min)

L6: Waist bends forward (20x)

L7: Frontal elevation of arms (20x)

L8: Knees bending (crouching) (20x)

L9: Cycling (1 min)

L10: Jogging (1 min)

L11: Running (1 min)

L12: Jump front & back (20x)

NOTE: In brackets are the number of repetitions (Nx) or the duration of the exercises (min).

3.3. ATTRIBUTE INFORMATION

The data collected for each subject is stored in a different log file: 'mHealth_subject.log'. Each file contains the samples (by rows) recorded for all sensors (by columns). The labels used to identify the activities are similar to the abovementioned (e.g., the label for walking is '4').

The meaning of each column is detailed next:

Column 1: acceleration from the chest sensor (X axis)
Column 2: acceleration from the chest sensor (Y axis)
Column 3: acceleration from the chest sensor (Z axis)
Column 4: electrocardiogram signal (lead 1)
Column 5: electrocardiogram signal (lead 2)
Column 6: acceleration from the left-ankle sensor (X axis)
Column 7: acceleration from the left-ankle sensor (Y axis)
Column 8: acceleration from the left-ankle sensor (Z axis)
Column 9: gyro from the left-ankle sensor (X axis)
Column 10: gyro from the left-ankle sensor (Y axis)
Column 11: gyro from the left-ankle sensor (Z axis)
Column 13: magnetometer from the left-ankle sensor (X axis)
Column 13: magnetometer from the left-ankle sensor (Y axis)
Column 14: magnetometer from the left-ankle sensor (Z axis)
Column 15: acceleration from the right-lower-arm sensor (X axis)
Column 16: acceleration from the right-lower-arm sensor (Y axis)
Column 17: acceleration from the right-lower-arm sensor (Z axis)
Column 18: gyro from the right-lower-arm sensor (X axis)
Column 19: gyro from the right-lower-arm sensor (Y axis)
Column 20: gyro from the right-lower-arm sensor (Z axis)
Column 21: magnetometer from the right-lower-arm sensor (X axis)
Column 22: magnetometer from the right-lower-arm sensor (Y axis)
Column 23: magnetometer from the right-lower-arm sensor (Z axis)
Column 24: Label (0 for the null class)

*Units: Acceleration (m/s^2), gyroscope (deg/s), magnetic field (local), ecg (mV)

4. RESULTS

As mentioned before, Multilayer Perceptron and Support Vector Machines algorithms were applied to analyze the dataset. The results are analyzed in terms of time performance and accuracy.

4.1. MLP RESULTS

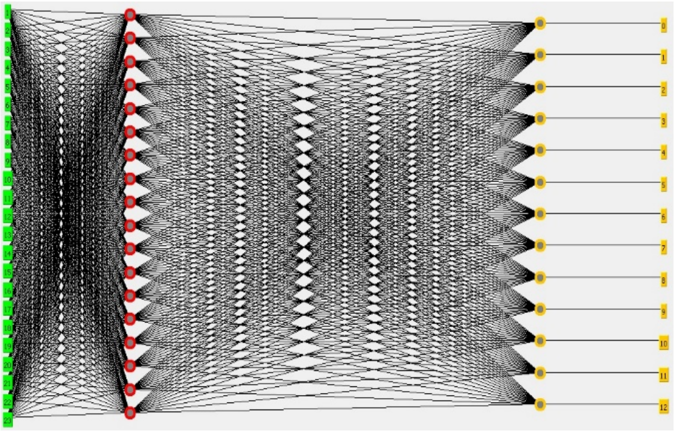
Multilayer Perceptron does quite well for real classification problems, but it may process data very slowly. Considering speed of the algorithms, we can say that MLP is 10 to 2000 times slower than the other algorithms. This is an important disadvantage for MLP. Multilayer Perceptron implement arbitrary decision boundaries;

- Given two or more hidden layers that are large enough and are trained properly
- Training by back propagation;
- Iterative algorithm based on gradient descent
- Quite good performance but extremely slow

Since there are 161.280 lines of data for a subject's body motion record, applying an extensive algorithm such as Multilayer Perceptron takes quite time to give a result. Considering that Weka uses K-fold cross-validation for classification problems, we see that with the cross-validation process, it is quite normal that it takes hours for analyzing and applying MLP on a dataset with 161.280 lines of data.

There are 161280 cases (instances) each with 24 input variables. Neural network in MLP is using the given values of the 24 input variables to predict our class variable, which as we know, consists of a value that represents body motion of a subject. So the training is to adjust the internal weights to get as close as possible to the known class values.

There are 13 output units and 18 hidden nodes labeled Sigmoid Node 0….30. The weights are given for each variable that feeds into each Sigmoid Node, plus the threshold weight, which is used to give some input to the output neuron in case of some problem with the other weights. It is like having an extra number in the denominator to prevent division by zero.

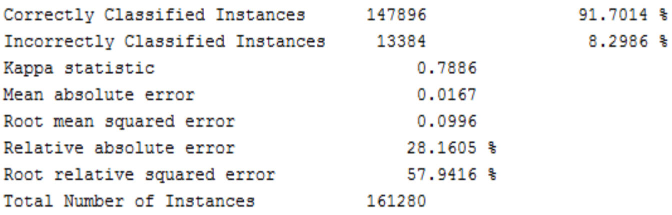


Neural network structure of MLP applied on MHEALTH dataset

The 18 hidden nodes, pass an output value to the output node itself called the linear node 0 which has a feed, a weight, from each of the 18 hidden neurons. One way to think of the weights is like a slope value. A high + weight means a high correlation between the variable and the outcome. A high - weight means a high negative correlation. A near zero weight means that variable has little or no effect on the outcome.

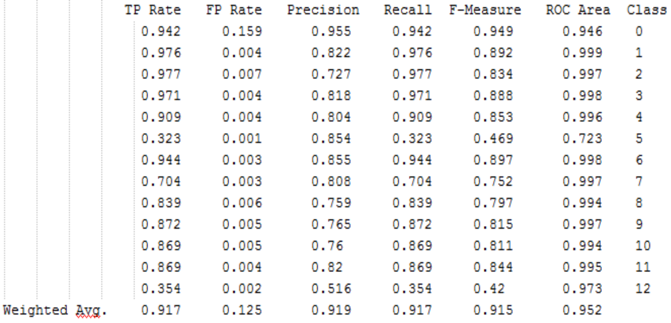
In all the variables, the threshold weight is either the highest or very high which is not good since the network is putting a lot of emphasis on that number which is not part of our data set. We need another variable of a type that covers the data space in a way that the other variables do not. We want to get the threshold neurons to be close to zero and randomly + or - in sign, in order to insure that the data we are using really does predict the outcome of our class. How to use this existing trained network is to save the network and then call it with the testing set and see how it does. Since we got more than 80% (91%), we can use this network with our unknown data. We can try to collect another variable to try to improve the train outcome.

=== Stratified cross-validation ===
=== Summary ===



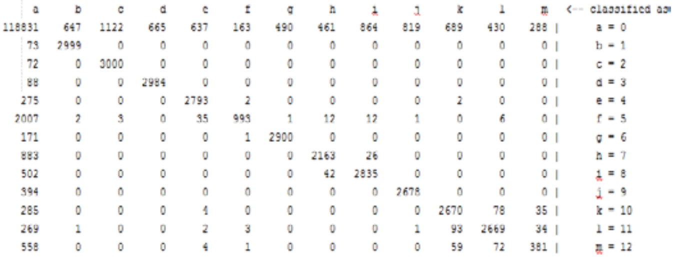
```
Correctly Classified Instances      147896               91.7014 %
Incorrectly Classified Instances     13384                8.2986 %
Kappa statistic                          0.7886
Mean absolute error                      0.0167
Root mean squared error                  0.0996
Relative absolute error                 28.1605 %
Root relative squared error             57.9416 %
Total Number of Instances           161280
```

=== Detailed Accuracy by Class ===



```
                 TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                   0.942     0.159     0.955      0.942     0.949      0.946    0
                   0.976     0.004     0.822      0.976     0.892      0.999    1
                   0.977     0.007     0.727      0.977     0.834      0.997    2
                   0.971     0.004     0.818      0.971     0.888      0.998    3
                   0.909     0.004     0.804      0.909     0.853      0.996    4
                   0.323     0.001     0.854      0.323     0.469      0.723    5
                   0.944     0.003     0.855      0.944     0.897      0.998    6
                   0.704     0.003     0.808      0.704     0.752      0.997    7
                   0.839     0.006     0.759      0.839     0.797      0.994    8
                   0.872     0.005     0.765      0.872     0.815      0.997    9
                   0.869     0.005     0.76       0.869     0.811      0.994    10
                   0.869     0.004     0.82       0.869     0.844      0.995    11
                   0.354     0.002     0.516      0.354     0.42       0.973    12
Weighted Avg.      0.917     0.125     0.919      0.917     0.915      0.952
```

=== Confusion Matrix ===



```
     a      b      c      d      e      f      g      h      i      j      k      l      m   <-- classified as
118831    647   1122    665    637    163    490    461    864    819    689    430    288 |      a = 0
    73   2999      0      0      0      0      0      0      0      0      0      0      0 |      b = 1
    72      0   3000      0      0      0      0      0      0      0      0      0      0 |      c = 2
    88      0      0   2984      0      0      0      0      0      0      0      0      0 |      d = 3
   275      0      0      0   2793      2      0      0      0      0      2      0      0 |      e = 4
  2007      2      3      0     35    993      1     12     12      1      0      6      0 |      f = 5
   171      0      0      0      0      1   2900      0      0      0      0      0      0 |      g = 6
   883      0      0      0      0      0      0   2163     26      0      0      0      0 |      h = 7
   502      0      0      0      0      0      0     42   2835      0      0      0      0 |      i = 8
   394      0      0      0      0      0      0      0      0   2678      0      0      0 |      j = 9
   285      0      0      0      4      0      0      0      0      0   2670     78     35 |      k = 10
   269      1      0      0      2      3      0      0      0      1     93   2669     34 |      l = 11
   558      0      0      0      4      1      0      0      0      0     59     72    381 |      m = 12
```

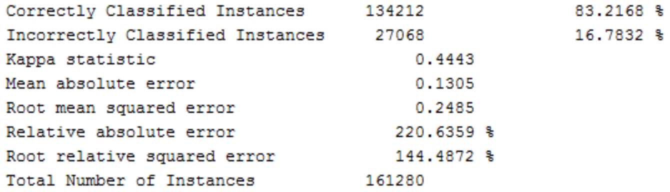
4.2. SVM RESULTS

In this part, we build Support Vector Machine (SVM) models for predicting the body motions of a subject. The goal is to learn the possibilities offered by the Weka software for that. The Weka software implements John Platt’s Sequential Minimal Optimization (SMO) algorithm for training a support vector classifier, and this explains abbreviation SMO used in Weka for this methods. Also, we used polynomial kernel which is defined in Sequential Minimal Optimization (SMO) algorithm to train our support vector classifier.

Considering the number of instances (161280) in our dataset, we have obtained very good model with a small number of misclassification errors (27068) and rather high value of the Kappa statistic 0.4443. The only thing that became worse in comparison with previous models is ROC Area. This can however be easily explained by the fact that original SVM method is not probabilistic, and only a single optimal value of threshold (which is in the case of the standard SVM approach the distance between the separating hyper plane and the coordinate origin) is provided. Without such freely moving threshold, it would not be possible to perform virtual screening based on ranking chemical compounds and adjusting threshold for selection.

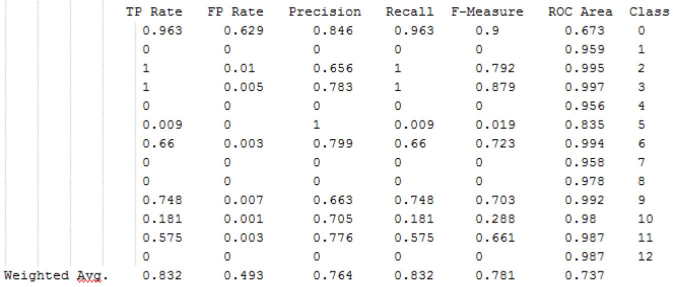
This results in the relatively bad value of ROC Area. Nonetheless this can be improved by using a special modification of the original SVM approach, which assigns probability value to each prediction. Since the algorithm for assigning probability values to SVM predictions is based on the use of logistic functions, such models are called in Weka Logistic Models.

=== Stratified cross-validation ===
=== Summary ===



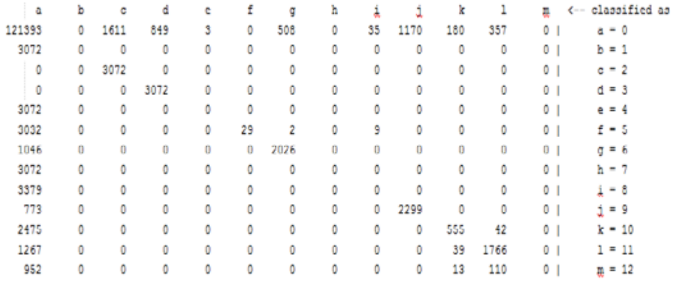
```
Correctly Classified Instances      134212              83.2168 %
Incorrectly Classified Instances     27068              16.7832 %
Kappa statistic                          0.4443
Mean absolute error                      0.1305
Root mean squared error                  0.2485
Relative absolute error                220.6359 %
Root relative squared error            144.4872 %
Total Number of Instances           161280
```

=== Detailed Accuracy by Class ===



```
                 TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                   0.963     0.629      0.846     0.963     0.9        0.673    0
                   0         0          0         0         0          0.959    1
                   1         0.01       0.656     1         0.792      0.995    2
                   1         0.005      0.783     1         0.879      0.997    3
                   0         0          0         0         0          0.956    4
                   0.009     0          1         0.009     0.019      0.835    5
                   0.66      0.003      0.799     0.66      0.723      0.994    6
                   0         0          0         0         0          0.958    7
                   0         0          0         0         0          0.978    8
                   0.748     0.007      0.663     0.748     0.703      0.992    9
                   0.181     0.001      0.705     0.181     0.288      0.98     10
                   0.575     0.003      0.776     0.575     0.661      0.987    11
                   0         0          0         0         0          0.987    12
Weighted Avg.      0.832     0.493      0.764     0.832     0.781      0.737
```

=== Confusion Matrix ===



```
     a     b     c     d     e     f     g     h     i     j     k     l     m   <-- classified as
121393     0  1611   849     3     0   508     0    35  1170   180   357     0 |     a = 0
  3072     0     0     0     0     0     0     0     0     0     0     0     0 |     b = 1
     0     0  3072     0     0     0     0     0     0     0     0     0     0 |     c = 2
     0     0     0  3072     0     0     0     0     0     0     0     0     0 |     d = 3
  3072     0     0     0     0     0     0     0     0     0     0     0     0 |     e = 4
  3032     0     0     0     0    29     2     0     9     0     0     0     0 |     f = 5
  1046     0     0     0     0     0  2026     0     0     0     0     0     0 |     g = 6
  3072     0     0     0     0     0     0     0     0     0     0     0     0 |     h = 7
  3379     0     0     0     0     0     0     0     0     0     0     0     0 |     i = 8
   773     0     0     0     0     0     0     0     0  2299     0     0     0 |     j = 9
  2475     0     0     0     0     0     0     0     0     0   555    42     0 |     k = 10
  1267     0     0     0     0     0     0     0     0     0    39  1766     0 |     l = 11
   952     0     0     0     0     0     0     0     0     0    13   110     0 |     m = 12
```

6. CONCLUSION

Machine learning algorithms can figure out how to perform important tasks by generalizing from examples. This is often feasible and cost-effective when manual programming is not. Applications of machine learning in healthcare sector are increasing with each passing day. In healthcare sector, in order to get better results from diagnosis and treatment done by doctors, to prevent human-induced mistakes, and to help doctors' decision, machine learning based decision support systems are used. In this study, we applied two different supervised machine learning algorithms called Multilayer Perceptron and Support Vector Machine to analyze a well-established dataset called MHEALTH with the purpose of classifying body movements of a volunteer. This study can be further used for monitoring patients' daily activities, and if necessary restricting them from doing any movements that would be harmful for their health.

According to experimental and statistical studies, Multi-layer Perceptron algorithm gives much accurate results but very slow in terms of performance. However, Support VectorMachine algorithm is much faster yet gives less accurate results than Multi-layer Perceptron.

This study is a starting point for the future studies. In the future studies, improvement of machine learning algorithms used in this study and hybrid use of them are under consideration. Hereby, it is expected to increase the classification performance and to reduce the operating time of the algorithms.

REFERENCES

[1] Hosch WL. (2014) "Machine learning" In Encyclopedia Britannica. Retrieved from http://www.britannica.com/technology/machine-learning.

[2] M. Riedmiller (1994) Advanced Supervised Learning in Multi-layer Perceptrons fromBackpropagationto Adaptive Learning Algorithms.University of Karlsruhe.

[3] Gales M. (2013) Statistical PatternProcessing. University of Cambridge

[4] Zurada JM. (1992) Introduction to Artificial Neural Systems. West Publishing Company.

[5] Çetin M. Uğur A. Bayzan Ş. (2006) İleriBeslemeliYapaySinirAğlarındaBackpropagationAlgoritmasınınSezgiselYaklaşımı. IV. BilgitekveAkademikBilişimSempozyumu.

[6] Chang C-C. Lin C-J. (2001) A Library for Support Vector Machines.National Taiwan University, Taipei, Taiwan.

[7] MacLeod C. (2001) anIntroduction to Practical Neural Networks and Genetic Algorithms for Engineers and Scientists.