Biomedical Statistics and Informatics
2017; 2(3): 95-102
http://www.sciencepublishinggroup.com/j/bsi
doi: 10.11648/j.bsi.20170203.11

Diagnosing Knee Osteoarthritis Using Artificial Neural Networks and Deep Learning

Jean de Dieu Uwisengeyimana[1, 2, *], Turgay Ibrikci[1]

[1]Department of Electrical and Electronics Engineering, Cukurova University, Adana, Turkey

[2]Department of Electrical and Electronics Engineering, University of Rwanda, Kigali, Rwanda

Email address:

uwisenjeado@gmail.com (J. de D. Uwisengeyimana)

[*]Corresponding author

To cite this article:

Jean de Dieu Uwisengeyimana, Turgay Ibrikci. Diagnosing Knee Osteoarthritis Using Artificial Neural Networks and Deep Learning. *Biomedical Statistics and Informatics.* Vol. 2, No. 3, 2017, pp. 95-102. doi: 10.11648/j.bsi.20170203.11

Received: January 24, 2017; **Accepted**: February 18, 2017; **Published**: March 29, 2017

Abstract: Among various medical diagnostic tests performed to identify osteoarthritis in the knee, most of them are invasive and expensive. Therefore, in this study, another methodology for diagnosing osteoarthritis in the knee in a more qwick, non-invasive and cheap manner was proposed. For that purpose, surface electromyography signals recorded from the four muscles surrounding the knee, the recording of the flexion degree in the knee and pattern recognition algorithms were used. The datasets of this experiment comprised 22 subjects among whom 11 subjects had normal knee and other 11 Subjects had an osteoarthritis-affected knee. The total sample size was 1, 048, 576 samples and were processed using segments of overlapping-windows of 5000 samples. Time-series features were then extracted from each segment and were used to train, test and validate 7 different learning classifiers and 7 variants of deep learning networks. In this study, the best performance measure of 99.5% was achieved by multilayer perceptron. Quadratic support vector machine and complex tree performed as well with accuracy of 99.4% and 98.3% respectively. In contrast, the use of deep learning networks which were investigated over a wide range of hidden size of the sparse autoencoders, showed accuracy of 86.6% with final softmax layer and accuracy of 91.3% by replacing the final softmax layer with k-nearest neighbour. By comparison, artificial neural networks outperformed deep learning networks and it is therefore concluded that the knee pathology can be diagnosed more effeciently and automatically using surface electromyography signals and artificial neural network algorithms.

Keywords: Diagnosis, Electromyography, Deep Learning, Feature Extraction, Artificial Neural Network

1. Introduction

Osteoarthritis is one of joint pathologies that mostly affects cartilage and the majority of cases of cartilage damage involve the knee joint [1, 2]. Critical knee injuries such as those that damage the pairs of cruciate ligaments in the knee have been of the most interest in researches [3]. These knee damages, when not diagnosed and treated the soonest, may result in damage and deterioration resulting in disappearing of meniscus and weak alignment [4]. It is therefore very important to early diagnose pathology of the knee so as to treat it the soonest.

Several studies have studied the efficient way for diagnosing osteoarthritis. From [5], one method was by analysing the knee images recorded by X-Ray and to automate the detection of osteoarthritis using image classification techniques. Actually X-Ray technique is an efficient way to diagnose any default in bones and joints, but according to [6], this way is expensive and is not always preferred the earliest possible. Another technique is by using Magnetic Resonance Imaging (MRI), which is used to take many images of the knee cartilage and its ligaments. Studies [7] and [8], discussed the recent start of using MRI and quantitative image analysis technology to give information on the state of the cartilage, bone and degenerative changes in osteoarthritis. However, MRI tests are also expensive and most of the hospitals in developing countries do not have these equipments.

Hence, in this study we propose an automated process to diagnose any abnormal knee by using muscle signals recorded by the Surface Electromyography (SEMG) along with signal processing techniques and classifying algorithms.

In the literatures, the main use cases of Electromyography (EMG) signals are met in neurology, rehabilitation [9], ergonomics [10], and sports [11]. These signals are also used as diagnostic tool for neuro-mascular and motor control problems [12, 13]. According to [14], the state of quadriceps muscles can suggest whether the ligaments in the knee are unstable, or that there are problems with the knee.

To realize the purposes of this study, which is about building a prototyping system to automatically diagnose knee pathology to support related medical diagnosis, the combination of SEMG signals recorded from the four muscles of interest (rectus femoris, vastus medialis, biceps femoris and semitendinosus), goniometry signals and machine learning techniques wereused. Also a comparison between Artificial Neural Network (ANN) and deep learning has been done.

2. Materials and Methods

2.1. Lower Limb EMG Data Sets

In this study, we have used the newly uploaded datasets which are publicly available at UCI [15]. In these datasets, twenty two people were hired to record the data. From those people, eleven subjects had knee pathologies confirmed by the physiotherapist. Three positions which are *marching, sitting,* and *standing* were undergone by the subjects while analyzing the behavior associated with the knee muscles. The muscles of concern were the vastus medialis, semitendinosus, biceps femoris and rectus femoris. The data sets were acquired from a subject to a computer's storage through bluetooth. Also, 14-bit resolution and fs=1000 Hz were used. Finally, the total sample size acquired was 1, 048, 576 samples.

2.2. EMG Feature Extraction

Time series features can be extracted based on either time domain, frequency domain or a combination of the two signal domains. Each of these different types of features are used in a specific application. From many previous studies especially those concerned with EMG signals, the frequently used features were based on time-domain statistical features. [16-19]. The time-domain features were widely used in many literatures due to their relatively easy construction and high efficiency.

In this study, before feature extraction, EMG data acquired from all the four electrodeswere segmented resulting in a chain of analysis windows. As shown in figure 1, time overlapping windows method was used. The window overlapping was 90% and the window length was 5000 samples.

After signal windowing, extraction of time domain features was made from each of the analysis windows. It should be noted that the processes of signal windowing and feature extraction acted as dimensionality reduction which in the end resulted in 2642 samples and 30 features.

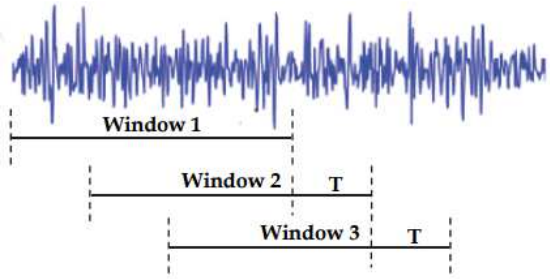

Figure 1. *Segmentation of analysis windows of EMG data [20].*

Below are formulas for the representative techniques in time-domain used for extracting signal features from each signal window:

- Mean power of the raw data:

$$P = \frac{1}{N}\sum_{n=1}^{N} X^2(n) \tag{1}$$

- Peak Value (PV) of the raw signal:

$$PV = MAX\left(\sqrt{X^2(n) + \hat{X}^2(n)}\right) \tag{2}$$

Where, $\hat{X}^2(n)$ is Hilbert transform of $X(n)$, $X_+(n)$ is the pre-envelope of $X(n)$ and $\sqrt{X^2(n) + \hat{X}^2\,(n)}$ is the envelope of $X(n)$

- Mean of the raw data (μ_x):

$$\mu_x = \frac{1}{N}\sum_{n=1}^{N} X(n) \tag{3}$$

- Standard deviation of raw data (σ_x):

$$\sigma_x = \sqrt{\left\{\frac{1}{N}\sum_{n=1}^{N} (X(n) - \mu_x)^{\,2}\right\}} \tag{4}$$

- Variance of the raw data (σ_x^2):

$$\sigma_x^2 = \left\{\frac{1}{N}\sum_{n=1}^{N} (X(n) - \mu_x)^2\right\} \tag{5}$$

Where, $X(n)$ is the vector of the data points and μ_x is the mean of the data points.

2.3. Classification Algorithms

After processing the SEMG signals, several classifiers were applied using Matlab toolbox [21].

2.3.1. Multilayer Perceptron (MLP)

Multilayer Perceptron was implemented to tackle the limitation of single layer Perceptron. To train the multilayer perceptron network, the error back-propagation learning algorithm is adopted [22]. The basis of this learning algorithm is on the error-correction learning rule, described by the below equation.

$$w(n+1) = w(n) + \eta[d(n) - y(n)]x(n) \tag{6}$$

Where in the equation above, $x(n)$is the examples' input vector, $w(n)$ is the weight vector, $b\,(n)$ is the bias, $y(n)$ is the system response, $d\,(n)$ is the target response, η is the learning rate constant, which is a constant less than a unit and

n is the time step, $n = 0, 1, 2, ...$

2.3.2. Support Vector Machine

i. Linear Support Vector Machines (SVMs)

SVMs were previously employed in classification of two categories by setting a maximum margin hyperplane between the two classes. Assuming a linear classifier with a function

$$\mathrm{f(w, b)} = signum(w^T x + b) \quad (7)$$

Then, the distance from the point x to the hyperplane saparating the classes is given by

$$\frac{w^T x_i + b}{\|w\|} \quad (8)$$

Thus, SVMs consist of the following constrained optimization:

$$\min_{w, \xi_i} \frac{1}{2} w^T w + C \sum_{n=1}^{N} \max\ (0, 1 - y_i\ (w. x_i + b)) \quad (9)$$

Where, in the equations (7)- (9), x_i are support vectors, w is the normal vector to the plane, y_i is the desired class response, b indicates the bias, C is the normalization constant and, ξ_i are slack variables that are inserted into the equation to give the classifier the ability to handle some data that could not be well separated, like those data containing noise. [23]. The of equation (9) is referred to as the primal form of norm L1-SVM

ii. Quadratic Support Vector Machine

The fact that the equation (9) is not differentiable leads to another most used variation known as the dual formor as the norm L2-SVM which minimizes the squared hinge loss:

$$\min_{w, \xi_i} \frac{1}{2} w^T w + C \sum_{n=1}^{N} {\xi_i}^2 \quad (10)$$

Hence, since norm L2-SVM is differentiable, this is what shows a difference between quadratic and linear SVM. For Kernalized SVMs, like quadratic SVM, optimization must be done in dual rather than primal form [24, 25].

2.3.3. Logistic Regression

Assuming that we have a function g: $X \rightarrow C$, where C is the class label, and $X = (X_1, X_2, ..., X_N)$ is examples' input vector; this method considers a distribution in the form of $P(C|X)$ and immediately approximates its parameters from the examples [26, 27]. This shows that logistic regression is a parametric learning model and the parameters that it surmises in the case where class Chas boolean labels (0 or 1) is given by:

$$P(Y = 1|X) = \frac{1}{1 + e^{\,(w_o + \sum_{i=1}^{n} w_i X_i)}} \quad (11)$$

and,

$$P(Y = 0|X) = \frac{e^{\,(w_o + \sum_{i=1}^{n} w_i X_i)}}{1 + e^{\,(w_o + \sum_{i=1}^{n} w_i X_i)}} \quad (12)$$

With logistic regression, the distribution $P(C|X)$ is thought of as to keep track of the shape of logistic function. So considering the case of linear classification, the model sets category C= 0 if X satisfies the equation:

$$0 < w_o + \sum_{i=1}^{n} w_i\, X_i \quad (13)$$

and sets C= 1 otherwise.

2.3.4. Linear Discriminant Analysis

The concept of Linear Discriminant Analysis (LDA) is to separate the samples from two different classes by making the inter-class distance large while setting small intra-class variances. For long time, LDA has been an important way for either classification or dimension reduction tasks. For classification tasks, LDA is based on mahalanobis distance (D_M) [28]:

$$D_M = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu})^T C^{-1}\ (\boldsymbol{x} - \boldsymbol{\mu})} \quad (14)$$

Where, $\boldsymbol{x} = (x_1, x_2, ..., x_n)^T$ is the predictors' vector, $\mu = (\mu_1, \mu_2, ..., \mu_n)^T$ is the class centroids' vector and C is the pooled within-class covariance of the predictors.

2.3.5. Complex and Medium Decision Trees

Decision Trees (DTs) are tree-like decision techniques employed to construct classification or regression systems [29]. In each iteration of decision tree learning algorithms, a dataset is fed and a variable is sorted out and is used to split up the dataset into subsets; where every subset is considered as the provided data set for the next iteration. Now the concept of decision tree algorithm is based on using information gain to decide which best variable to select to test each node. To introduce information gain, we first introduce the entropy, which measures the amount of information and noise present in a signal [30].

Given a binary classification problem with a set D of posive (p_+) and negative (p_-) examples, the entropy of a set D relative to this binary categorization is given by:

$$Entropy\ (D) = -p_+ \log_2\ (p_+) - p_- \log_2\ (p_-) \quad (15)$$

Then, the information gain which measures the expected reduction in entropy, or uncertainty caused by partitioning the examples according to the selected attributes, is given by

$$Gain\ (D, A) = Entropy(D) - \sum_{v \in Values\ (A)} \frac{D_v}{D}\ Entropy\ (D_v) \quad (16)$$

Where, $Values\ (A)$ is a set of all possible values for attribute A and D_v is the subset of D for which attribute A has a value v.

The main difference between complex tree and medium tree is determined by the number of leaves; where for medium tree, this number does not go beyond 20 splits while for complex tree it can come up to 100 splits.

2.3.6. k-Nearest Neighbours

k-Nearest Neighbours (k-NN) is a lazy learning classifier, which means that it hardly learns anything from the training examples [31]. To classify any new data sample, the k-NN algorithm will have to first estimate the k closest neighbors from the training examples to the new sample. Then, the system class label for the new sample, will be the same as the class label of the k closest neighboring points. If k = 1, the

new sample will be automatically set to the class of its nearest neighbor.

To find k-neighbors closest to the new data samplek-NN, uses a metric for measuring the distance between the new point and cases from the examples. The most popular distance functions to measure this distance are defined below:

$$Euclidean = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2} \quad (17)$$

$$Manhattan = \sum_{i=1}^{N}|x_i - y_i| \quad (18)$$

$$Minkowski = (\sum_{i=1}^{N} (|x_i - y_i|)^q)^{1/q} \quad (19)$$

Where, x and y are the new point and a case from the examples, respectively.

2.3.7. Deep Learning Networks and Algorithm

A deep learning model is basically an artificial neural network that has more than one hidden layers [32-34]. The first breakthrough results in deep learning appeared since early 2000 and used Deep Belief Networks (DBNs) to pre-train deep networks [35-37]. In many past studies, autoencoders have regained the prominence in the deep learning approach designed for the pre-training task [38, 39].

Algorithm for autoencoders is influenced by the concept of a good depiction of the data. Suppose the input to an autoencoder is a vector $x \epsilon \mathbb{R}^{D_x}$, then the encoder maps the vector x back and forth to another vector $z \epsilon \mathbb{R}^{D^{(1)}}$ as follows:

$$z^{(i)} = W_1^{(i)} x + b_1^{(i)} \quad (20)$$

$$\tilde{x}^{(i)} = W_2^{(i)} x + b_2^{(i)} \quad (21)$$

Where, the superscript (i) indicates the layer, $W_1^{(i)} \epsilon \mathbb{R}^{D^{(i)} * D_x}$ and $W_2^{(i)} \epsilon \mathbb{R}^{D^{(i)} * D_x}$ are weight matrices for encoder and decoder respectively and $b_1^{(i)} \epsilon \mathbb{R}^{D^{(i)}}$ and $b_2^{(i)}$ are a bias vectors for encoder and decoder respectively.

We can set up the following objective function, which is the sum of squared differences between $\tilde{x}^{(i)}$ and $x^{(i)}$:

$$J(W_1, b_1, W_2, b_2) = \sum_{i=1}^{m} \left(\tilde{x}^{(i)} - x^{(i)}\right)^2 \quad (22)$$

$$= \sum_{i=1}^{m} \left(W_2 z^{(i)} + b_2 - x^{(i)}\right)^2 \quad (23)$$

$$= \sum_{i=1}^{m} \left(W_2 \left(W_1 x^{(i)} + b_1\right) + b_2 - x^{(i)}\right)^2 \quad (24)$$

Where, in the formulas (22)- (24) again, $x^{(i)}$ are the input vectors to the autoencoder, $z^{(i)}$ are the coded input vector, $\tilde{x}^{(i)}$ are the decoded input vector, W_1 and b_1 are respectively weight and bias parameters for the encoder and W_2 and b_2 are also respective weight and bias parameters for the decoder.

3. Results and Discussions

In this study, the main goal wasto build a diagnosing sytem which would be able to automatically distinguish between the patients having abnormal knee and those having normal knee with least error rate. To do this, EMG signals acquired from four muscles surrounding the knee were first filtered using a second order Chebyshev filter which removed signals that were outside the bandwidth of the electromyography signals which roughly ranges from 20Hz to 500Hz [40]. For this filter design, the attenuation ripple was 3 dB and the attenuation of unwanted signals was 60 dB.

From the filtered signals, thirty time-domain features were extracted and were used to train, test and validate 7 classifiers and 7 deep learning networks. The results from the first seven classifiersare presented in table 1, where different classifier evaluation metrics such as classification accuracy, precision (specificity), recall (sensitivity), F-measure and the error rate were calculated from the confusion matrix. For this purpose, the formulas below were used.

Classification accuracy: The number of correct predictions from all the predictions made.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (25)$$

Recall: The proportion of actual positive which are predicted positive.

$$Recall\ (Sensitivity) = \frac{TP}{TP+FN} \quad (26)$$

Precision: The proportion of predicted positive which are actual positive.

$$Precision = \frac{TP}{TP+FP} \quad (27)$$

F-measure: It is harmonic mean between Precision and Recall and also known as F-score.

$$F = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (28)$$

Error Rate: The number of mistakes made by the classifier.

$$Error\ Rate = \frac{FP+FN}{TP+TN+FP+FN} \quad (29)$$

Where in the formulas above, TP is True positive, TN is True Negative, and FP is False Positive, and FN is False Negative.

Table 1. *The statistical performance averages for the 7. different classifiers.*

Classifier Group	Classifier	Accuracy%	AUC	Recall	Precision	F-measure	Error Rate
Decision Trees	*Complex Tree (CT)*	98.3	0.99	0.98	0.99	0.98	0.017
	Medium Tree (MT)	97	0.98	0.93	0.99	0.96	0.03
Discriminant Analysis	*Linear Discriminant Analysis (LDA)*	65.9	0.72	0.73	0.59	0.65	0.341

Classifier Group	Classifier	Accuracy%	AUC	Recall	Precision	F-measure	Error Rate
Logistic regression	*Logistic regression (LR)*	78.6	0.88	0.71	0.86	0.78	0.214
Feed Forward Networks	*MultiLayer Percetron (MLP)*	99.5	1	0.99	0.99	0.99	0.005
Support Vector Machines	*Linear SVM (LSVM)*	81.2	0.85	0.71	0.93	0.8	0.188
	Quadratic SVM (QSVM)	99.4	1	0.99	0.99	0.99	0.006

In addition to the performance metrics described above, a Receiver Operating Characteristic (ROC) curve was another important measure used toselect the best classifier [41-43]. To compare classifiers ROC performance was expressed as a single scalar valuestanding in place of expected performance by calculating the area under the ROC curve (AUC). Noting that the value of AUC is always between 0 and 1, then the best classifier is the one with a relatively high value of AUC. combining the results of all these performance metrics, we can see that the most three competitive algorithms for this application with their respective accuracies were shown up to be *multilayer percetron (99.5%), Quadratic SVM (99.4%)* and *Complex Tree (98.3%)*.

From the column bar chart of figure 2. which compares the performance metrics among the seven classification learners, it is seen that MLP is outstanding due to its highest accuracy, recall, precision, F-measure and its lowest error rate. Furthermore, MLP systems actually have one more other advantage over kernelized SVM that might make it more applicable: Regarding the training time, MLP is quicker than SVM and the reason for the relative slowness of SVM is that its training requires calculating the solutions of the associated Lagrangian dual problem rather than primalproblem [44].

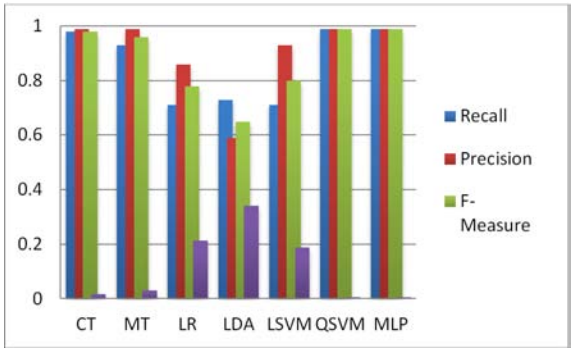

Figure 2. *Comparing performance metrics among the trained classification learners.*

Apart from shallow classification learners, 7 deep learning models were also investigated. To make the first basis of deep learning system, we made an artificial neural network that had two hidden layers and train the hidden layers separately in unsupervised manner using two sparse autoencoders. Then, by use of the output of the last autoencoder, we trained a final softmax layer, and matched the layers together [45]. Then, different performances of this deep learning model were achieved by varying the size of the hidden layer for the two autoencoders. Where, the highest accuracy of 86.6% was achieved with 900 nodes in sparse autoencoder 1. and 200 nodes in sparse autoencoder 2. as shown in figure 3.

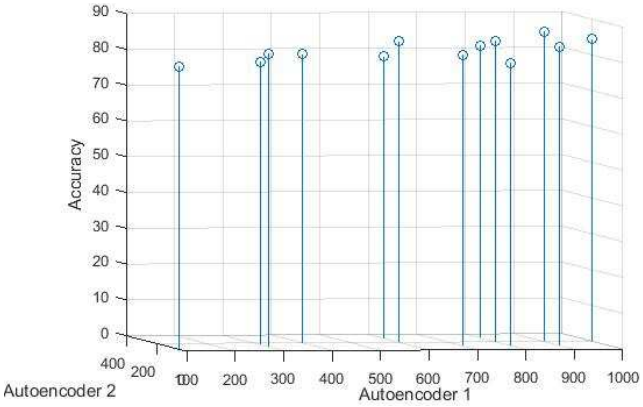

Figure 3. *3-D plot of deep net performance with variation of autoencoder's hidden size.*

In addition to the size of the hidden layers, other parameters of the deep learning network such as maximum epoch, *sparsity regularizer* and the *sparsity proportion* were varied along intervals from 100 to 1500 for maximum epoch, from 1 to 10 for the sparsity regularizer and from 0 to 1 for the sparsity proportion. The result of this parametric variation was the gain of different accuracies of the network. In figure 4, we plotted the resulting accuracy of 86.6% by adjusting maximum epoch, whereas the sparsity regularizer and the sparsity proportion were fixed at 7 and 0.35 respectively. The maximum accuracy was achieved at the maximum epoch of 1000.

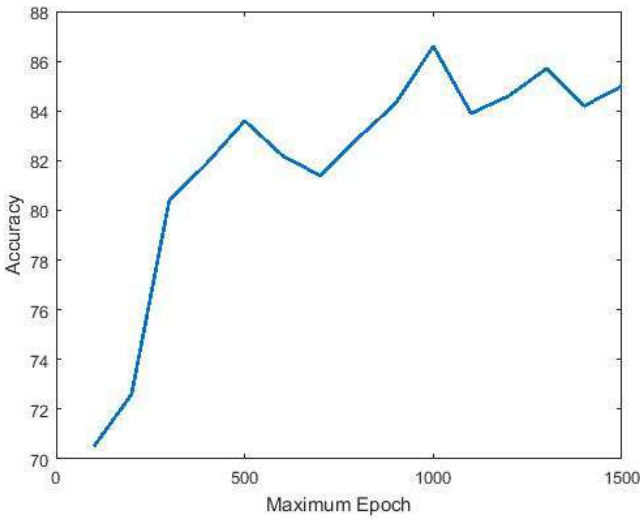

Figure 4. *Plot of the resulting accuracy by varying maximum epoch parameter.*

Several other means have also been investigated in order to boost the accuracy of the earlier-discussed deep learning network. Hence, new variants of the earlier deep learning

architecture were made by replacing the final softmax layer with other standard learning algorithms. Hence, instead of feeding the output of the second autoencoder to the final softmax layer, it was rather fed to complex tree, medium tree, logistic regression, k-NN, quadratic SVM and linear discriminant. By this way, we achieved an accuracy of 91.3% with a deep learning network made by combination of sparse autoencoder's output with k-NN. Detailed results are presented in table 2. and the performance comparison has been shown in figure 5.

Table 2. *Results of new variants of deep learning by different final layer classification algorithms.*

Base algorithm	Final layer algorithm	Accuracy%	AUC	Recall	Precision	F-Measure	Error Rate
Sparse Autoencoders+	*Softmax*	86.6	0.91	0.90	0.89	0.89	0.134
	Complex Tree (CT)	86.5	0.89	0.86	0.9	0.88	0.135
	Medium Tree (MT)	81.9	0.85	0.82	0.88	0.85	0.181
	Logistic regression (LR)	88.9	0.88	0.86	0.9	0.88	0.111
	k-NN	91.3	0.91	0.88	0.93	0.90	0.087
	Quadratic SVM (QSVM)	89.8	0.94	0.84	0.93	0.88	0.102
	Linear Discriminant Analysis (LDA)	64.6	0.54	0.55	0.81	0.65	0.354

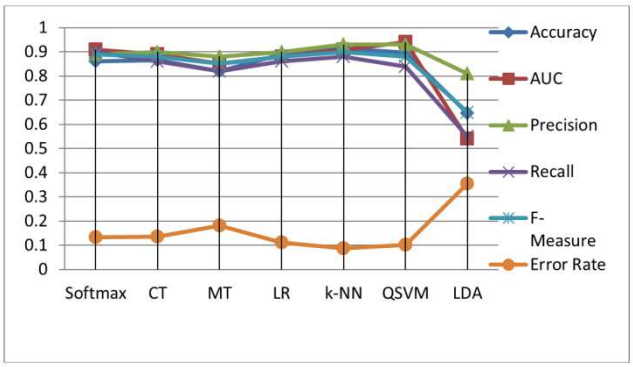

Figure 5. *Comparing results obtained by combining sparse autoencoders with different learning algorithms.*

4. Conclusion

Automatic diagnosis is very important in medical field. In this study, the main purpose was to distinguish healthy people from people with knee abnormality. For this purpose, signals from four lower limb muscles recorded using a four-channel Surface Electromyography (SEMG) was used along with goniometer signals which measures the flexion at the knee. From EMG and goniometer data obtained, statistical features were extracted and applied to train, test and validate 7. learning classifiers and 7 variants of deep learning systems. In this study, the best performance measure of 99.5% was achieved by multilayer perceptron. Quadratic support vector machine and complex tree also achieved a good accuracy of 99.4% and 98.3% respectively. Whereas, the combination of deep learning network with k-NN showed a relatively lower accuracy (91.3%) in comparison to artificial neural networks. Therefore, it is concluded that the knee pathology can be diagnosed using surface Electromyography signals and Artificial Neural Network (ANN) algorithms.

Acknowledgments

The authors would firstly like to appreciate the research funds provided by Çukurova University with the Project Number: FYL-2017-7864. We also thank the Scientific and Technological Research Council of Turkey (TÜBİTAK)-BIDEB2235 and the University of Rwanda (UR) for their help, which made this work possible.

References

[1] McAlindon, T. E., Bannuru, R. R., Sullivan, M. C., Arden, N. K., Berenbaum, F., Zeinstra, S. M. B., Hawker, G. A., Henrotin, Y., Hunter, D. J., Kawaguchi, H., Kwoh, K., Lohmander, S., Rannou, F., Roos, E. M., Underwood, M. (2014). OARSI Guidelines for the non-surgical management of knee osteoarthritis. Osteoarthritis and Cartilage. Volume, 22: Pages, 363-388. DOI: http://dx. doi. org/ 10.1016/j. joca.2014.01.003.

[2] Teitel, A. D. and Zieve, D. (2010). Osteoarthritis. MedlinePlud Medical Encycolpedia. National Institutes of Health. Last updated: Sept 26, 2011.

[3] Centeno, C. J., Pitts, J., Sayegh, H. A., Freeman, M. D. (2015). Anterior cruciate ligament tears treated with percutaneous injection of autologous bone marrow nucleated cells: a case series. Journal of Pain Research. Volume, 8: Pages, 437–447. Doi: 10.2147/JPR. S86244.

[4] Robert, E. M., Stephen, J. M., Marsha, E. M., Angelo, J. C., Matthew, D. (2008). Management of the Patient with an ACL/MCL Injured Knee. N Am J Sports Phys Ther. Volume, 3 (4): Pages, 204–211.

[5] Shamir, L., Ling, S. M., Scott, W. W., Bos, A., Orlov, N., MacUra, T. J., Eckley, D. M., Ferrucci, L., Goldberg, I. G. (2009). Knee X-ray image analysis method for automated detection of Osteoarthritis. IEEE transactions on bio-medical engineering. 2009; 56 (2): 407-415. doi: 10.1109/TBME.2008.2006025.

[6] Bedson, J., Jordan, K., Croft, P. (2003). How do GPs use x-rays to manage chronic knee pain in the elderly? A case study. Ann Rheum Dis. Volume, 62: Page, 450–454. Available at www.annrheumdis.com.

[7] Eckstein, F., Burstein, D., Link, T. M. (2006). Quantitative MRI of cartilage and bone: degenerative changes in osteoarthritis. NMR Biomed. Volume, 19 (7): 822-54. DOI: 10.1002/nbm.1063

[8] Peterfy, C. and Kothari, M. (2006). Imaging osteoarthritis: magnetic resonance imaging versus x-ray. Curr Rheumatol Rep. Volume 8. (1): Pages, 16-21.

[9] Ferris, D. P., Gordon, K. E., Sawicki, G. S., Peethambaran, A. (2006). An improved powered ankle–foot orthosis using proportional myoelectric control. Gait & Posture, Volume 23, Issue 4, 2006, Pages 425–428. http://dx. doi. org/10.1016/ j. gaitpost.

[10] Gazzoni, M. (2010), Multichannel surface electromyography in ergonomics: Potentialities and limits. Hum. Factors Man. Volume, 20: Pages, 255–271. Doi: 10.1002/hfm.20219.

[11] Jefferson, F. L., Debora, C., Fabia, M., Marcelo, L. T., Claudia, T. C. (2012). Evaluating the Electromyographical Signal During Symmetrical Load Lifting. Applications of EMG in Clinical and Sports Medicine. DOI: 10.5772/25732.

[12] Goen, A. (2014). Classification of EMG Signals for Assessment of Neuromuscular Disorders. International Journal of Electronics and Electrical Engineering Vol. 2, No. 3.

[13] Reaz, B. I., Hussain, M. S., Yasin, M. F. (2006) Techniques of EMG signal analysis: Detection, processing, classification and applications, Biol. Proced. Volume, 8 (1): Pages, 11-35.

[14] Gonzalez, M. H., Hernandez, G. M., Sotelo, J. R., Sanchez, O. A. (2015). Knee functional state classification using surface electromyographic and goniometric signals by means artificial neural networks. Ing. Univ. Volume, 19 (1): Pages, 51-66 http://dx. doi. org/ 10.11144/Javeriana. iyu19-1. kfsc.

[15] Sanchez, O. F. A., Sotelo, J. L. R., Gonzales, M. H., Hernandez, G. A. M. (2014). EMG dataset in Lower Limb. UCI Machine Learning repository, avaiable: 2014-02-05.

[16] Phinyomark, A., Phukpattaranont, P., Limsakul, C., Phothisonothai, M. (2011). Electromyography (Emg) Signal Classification Based On Detrended Fluctuation Analysis. Fluctuation and Noise Letters. Volume, 10 (3): Pages, 281-301. DOI: 10.1142/S0219477511000570.

[17] Krishnan, N. C., Juillard, C., Colbry, D., Panchanathan, S. (2009). Recognition of hand movements using wearable accelerometers. JAISE, Volume, 1 (2): Pages, 143-155.

[18] Tkach, D., Huang, H., Kuiken, T. A. (2010). Study of stability of time-domain features for electromyographic pattern recognition. Journal of NeuroEngineering and Rehabilitation. Volume, 7 (21). doi: 10.1186/1743-0003-7-21.

[19] Hamedi, M., Salleh, S. H., Noor, A. M., Swee, T. T., Afizam, I. K. (2012). comparison of Different Time-domain Feature Extraction Methods on Facial Gestures' EMGs. Progress In Electromagnetics Research Symposium Proceedings, KL, MALAYSIA, 1897-1900.

[20] Li, G. (2011). Electromyography Pattern-Recognition-Based Control of Powered Multifunctional Upper-Limb Prostheses, Advances in Applied Electromyography, Prof. Joseph Mizrahi (Ed.), InTech, DOI: 10.5772/22876.

[21] MATLAB, Statistics and Machine Learning Toolbox Release 2015b, The MathWorks, Inc., Natick, Massachusetts, United States.

[22] Gales, M. (2015). Statistical Pattern Processing. University of Cambridge Engineering Part IIB. http://mi. eng. cam. ac. uk/~mjfg/local/4F10/lect6. pdf

[23] Koo, J. Y., Lee, Y., Kim, Y., Park, C. (2008). Representation of the Linear Support Vector Machine. Journal of Machine Learning Research. Volume, 9 (2008): Pages, 1343-1368.

[24] Yeh, C. (2015). Support Vector Machines for classification. Methods and Theory. Available at http://efavdb.com/svm-classification/

[25] Chamasemani, F. F. and Singh, Y. P. (2011). Multi-class Support Vector Machine (SVM) Classifiers- An Application in Hypothyroid Detection and Classification. Bio-Inspired Computing: Theories and Applications (BIC-TA).

[26] Jurafsky, D. and Martin, J. H. (2016). Logistic Regression. Speech and Language Processing. Draft of November 7, 2016.

[27] Korkmaz, M., Güney, S., Yiğîter, Ş. Y. (2011). The Importance Of Logistic Regressıon Implementations In The Turkish Livestock Sector and Logistic Regression Implementations/Fields. J. Agric. Fac. Hr. U. Volume 16 (2): Pages, 25-36.

[28] Mandelkow, H., Zwart, J. A. D., Duyn, J. H. (2016). Linear Discriminant Analysis Achieves High Classification Accuracy for the bold fMRI Response to Naturalistic Movie Stimuli. Front Hum Neurosci. Volume: 10: Page 128. doi: 10.3389/fnhum.2016.00128.

[29] Sewaiwar, P. and Verma, K. K. (2015). Comparative Study of Various Decision Tree Classification Algorithm Using WEKA. International Journal of Emerging Research in Management &Technology. ISSN: 2278-9359, Volume-4, Issue-10.

[30] Patel, N. and Singh, D. (2015). An Algorithm to Construct Decision Tree for Machine Learning based on Similarity Factor. International Journal of Computer Applications, Volume 111 – No 10.

[31] Imandoust, S. B. and Bolandraftar, M. (2013). Application of k-Nearest Neighbor (k-NN) Approach for Predicting Economic Events: Theoretical Background. Int. Journal of Engineering Research and Applications. Volume, 3 (5): Pages, 605-610.

[32] Baldi, P. (2012). Autoencoders, Unsupervised Learning, and Deep Architectures. JMLR: Workshop and Conference Proceedings. Volume, 27: Pages, 37–50.

[33] Deng, L. and Yu, D. (2014). Deep Learning: Methods and Applications. Foundations and Trends in Signal Processing. Volume, 7 (3-4): Pages, 1–199. doi: 10.1561/2000000039.

[34] Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. Neural Networks. Volume, 61: Pages, 85–117. Available at doi: 10.1016/j.neunet.2014.09.003.

[35] Hinton, G. E., Osindero, S., Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural computation. Volume, 18 (7): Pages, 1527-1554.

[36] Bengio, Y. (2009). Learning Deep Architectures for AI. Foundations and Trends in Machine Learning. Volume, 2 (1): Pages, 1-127. http://dx. doi. org/ 10.1561/2200000006.

[37] Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H. (2007). Greedy layer-wise training of deep networks. In Advances in Neural Information Processing Systems.

[38] Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986). Learning Representations By Back-Propagating Errors. Nature. Volume 323: Pages, 533 – 536. Doi: 10.1038/323533a0.

[39] Erhan, D., Bengio, Y., Courville, A., Manzagol, P. A., Vincent, P. (2010). Why Does Unsupervised Pre-training Help Deep Learning?. Journal of Machine Learning Research. Volume, 11 (2010): Pages, 625-660.

[40] De Luca, C. J., Gilmore, L. D, Kuznetsov, M., Roy, S. H. (2010). Filtering the surface EMG signal: Movement artifact and baseline noise contamination. Journal of Biomechanics. Volume, 43 (2010): Pages, 1573–1579. Doi: 10.1016/j.jbiomech.2010.01.027.

[41] Matthias, K. (2012). Performance Measures in Binary Classification. International Journal of Statistics in Medical Research. Volume, 1: Pages, 79-81.

[42] Rotello, C. M. and Chen, T. (2016). ROC curve analyses of eyewitness identification decisions: An analysis of the recent debate. Cognitive Research: Principles and Implications 2016 1: 10. DOI: 10.1186/s41235-016-0006-7.

[43] Majnik, M. and Bosnic´, Z. (2013). ROC analysis of classifiers in machine learning: A survey. Intelligent Data Analysis 17, 531–558. DOI 10.3233/IDA-130592.

[44] Wright, S. (2009). Optimization Algorithms in Support Vector Machines. computational Learning Workshop, University of Wisconsin-Madison, Chicago, June 2009.

[45] LeCun, Y., Bengio, Y., Hinton, G. (2015). Review on deep learning. Nature, Volume, 521: Pages, 436-444. doi: 10.1038/nature14539.