MuscleNET: Smart Predictive Analysis for Muscular Activity Using Wearable Sensors

Mücahit Gemici
Dept. of Electrical-Electronics Eng.
Kadir Has University
İstanbul, Turkey
20171706024@stu.khas.edu.tr

Kazım Korkmaz
Dept. of Computer Eng.
Kadir Has University
İstanbul, Turkey
20171701050@stu.khas.edu.tr

Nail Tuğberk Ayhan
Dept. of Electrical-Electronics Eng.
Kadir Has University
İstanbul, Turkey
20171706028@stu.khas.edu.tr

Şule Soylu
Dept. of Electrical-Electronics Eng.
Kadir Has University
İstanbul, Turkey
20171706027@stu.khas.edu.tr

Faruk Güç
Dept. of Computer Eng.
Kadir Has University
İstanbul, Turkey
20171701052@stu.khas.edu.tr

Arif Selçuk Öğrenci
Dept. of Electrical-Electronics Eng.
Kadir Has University
İstanbul, Turkey
ogrenci@khas.edu.tr

2022 Innovations in Intelligent Systems and Applications Conference (ASYU) | 978-1-6654-8894-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ASYU56188.2022.9925553

***Abstract*—Doing weightlifting training at home has become more popular during the pandemic. Unfortunately, exercising without professional help can lead to dangerous injuries such as muscle tearing. It is possible to create a smart system with machine learning to overcome muscle injuries and suggest an appropriate training program. The use of suitable algorithms enables us to develop programs that can perform predictions based on sEMG (Surface Electromyography) signals. In this study, sEMG signals are collected from the skin surface and features are extracted to be used in deep learning networks. A wearable hardware collects sEMG signals and transfers them to our mobile application via Bluetooth. The mobile application transfers data to the cloud to make predictions based on sEMG signals. We developed MuscleNET for training monitoring, injury prediction/detection, and training quality prediction. Initial measurements indicate that MuscleNET can be used effectively for training quality prediction and real time training monitoring.

Keywords—muscle activity, signal processing, feature extraction, machine learning, deep learning, mobile application, training support

I. Introduction

The future goal of this work is to create an intelligent and secure muscle training tracking system to prevent injuries, and to guide the trainee with the most appropriate, personalized training strategy. Many people are engaged with sports in form of fitness training including weight lifting activities. With the pandemic, number of people practicing fitness at home has increased considerably. However, it is not possible to know or learn how individuals should do those activities and how they can develop their muscles without the help of an expert. In case there is no warning by any expert about a proper training program, trainees may face serious health problems such as muscle ruptures as a result of overloading their muscles. Moreover, many gyms have specialists, but not everyone has access to them, or some people may choose to exercise individually instead of going to the gym. With our MuscleNET application, individuals can monitor the development of their muscles without the need for an expert, and muscle ruptures can be prevented. As the health of individuals comes first in life, the aim is to let people do their sports safely, either individually or in a gym using a mobile application processing sEMG (surface ElectroMyoGraphy) signals from a wearable sensor. The sensor used is compatible with Bluetooth technology so that sensor readings will be transferred to the mobile phone. The application runs state of the art machine learning algorithms for training classification which makes it unique among the countless fitness apps available in the market. Using feature extraction methods, sensor data obtained are used to train machine learning algorithms which inform the user via the mobile app whether the dumbbell used for muscle development is sufficient or not. Moreover, if the training level is heavy for the user and will strain their muscle (and may cause any muscle rupture), the user is also notified. The application also stores training data of the trainee and allows the user to monitor the progress in real time. Section-II will summarize related work in the literature. The methodology applied will be explained in Section-III, and the preliminary results obtained will be displayed in Section-IV. Finally, Section-V will conclude the paper.

II. Related Work

Electrodes in form of medical pads are placed on the surface of the muscle to collect the electromyography signals, hence the name sEMG. The muscle sensor collects the electrical activities over the surface of the skin as muscles are contracted. Numerous papers have been published about processing sEMG signals using different methods for various purposes. Morphological filters [1] and linear discriminant analysis [2] have been deployed for muscle activity type detection. A wearable approach for monitoring leg exercises in a gym has been presented in [3] where a wearable textile sensor system is used to monitor muscle activity. The system records the activity including machines used, type of activity, and the quality of the workout is evaluated. This work provided 85% accuracy in quality recognition. Moreover, they showed how to evaluate the quality of an exercise through consistency measures. High-resolution time-frequency methods (e.g. Stockwell transform, standard and extended B-distribution) are applied to sEMG signals to identify muscle fatigue in [4]. Binary classification with feature engineering has achieved 86.9% accuracy in the identification of neuromuscular disorders in [5].

Machine learning methods have been utilized for noise removal and classification in sEMG signals: the work in [6] compares multilayer perceptron and recurrent neural network architectures. Feature extraction methods and selection of appropriate features for signal classification have been examined in [7, 8, 9, 10]. Specifically, KNN and ensemble methods based on bagged trees give the best accuracy (over 98%) for identification of different wrist and static hand gestures [8]. On the other hand, support vector machine has been found to outperform logistic regression and artificial neural networks in shoulder motion detection using sEMG signals [11]. For hand gesture recognition using sEMG obtained from armband sensors, self-organized maps combined with radial basis function neural networks has resulted in higher accuracy in comparison to several other machine learning methods such as k-means, k-nearest neighbor, and multi-layer perceptron [12]. Machine learning methods have also been used to estimate the muscle force measured through sEMG signals [13]. To the best of the authors, there is no research work on the identification of injury risk or training quality by means of a personal experience based classification of sEMG signals. Therefore, this work has focused on the classification of sEMG sensor signals on the upper arm collected during exercise. The knowledge about the use of proper features has been derived by examining the works cited above. The following section will summarize the methodology applied. Our work also targets the development of a mobile application which can be used during and after training sessions for real time warning generation and performance reporting. The details of the application will be highlighted in the following section.

III. METHODOLOGY

The proposed MuscleNET system is composed of hardware and software components. During the design and development of this system, ISO/IEC/IEEE 29148:2011 - Systems and Software Engineering - Life Cycle Processes - Requirements Engineering standard has been adopted. The requirements gathering process has involved stakeholder (192 students where 60% of them are performing sports) opinions in the form of a survey. Functional and nonfunctional requirements have been cross checked for completeness. To ensure a successful product, multiple design choices have been proposed during the process.

For the hardware of the MuscleNET system three alternative designs have been implemented for sensory signal reception and data communication:

- DF Robot Beetle BLE microcontroller board: It has a Bluetooth 4.0 chip (CC2540) *with* a maximum range of 50m, a Micro USB interface, 2 power ports, 4 digital ports, and 4 analog ports. Its major advantages are the very small size of 28.8mm x 33.1mm and weight of 10g.
- In the second design an Arduino Uno Board and an HC-06 Bluetooth module have been used.
- The final (and adopted) design uses an Arduino Nano Board and an HC-06 Bluetooth module. This design is preferred over the others because of its small size.

The design includes the MyoWare electromyography (EMG) sensor for the measurement of the electrical activity of the muscles during movement. It has two output modes, which are EMG Envelope and Raw EMG, LED indicators, connections to external electrode cables, and adjustable gain on it. We wanted to use this sensor because these sensors are designed to be used directly by microcontrollers. Thus, these sensors' primary output is the EMG envelope, which is an amplified, rectified, and integrated version of the raw EMG signal [14]. The hardware "box" contains a buzzer for audio warnings and a battery for mobile operation. The block diagram of the design is given in Fig. 1.

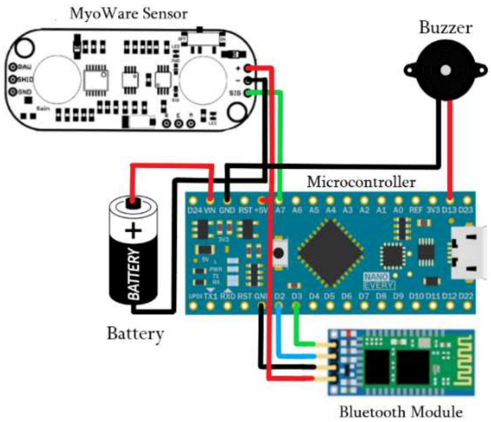

Fig. 1. Hardware components of MuscleNET.

The analysis and design of the software application has been carried out after examining various applications such as "StrongLifts", "Bulk Up," and "Sports Injury Rehabilitation" available in the application markets. The top-down system diagram of the software part of MuscleNET is displayed in Fig. 2, and the use case diagram in Fig. 3 shows the functions that an authenticated user can perform on the mobile application. Based on those analyses, a relational database schema has also been designed for the application. As an example, Fig. 4 shows the training activity diagram. If a user wants to start training, first they set the connection between device and mobile application. After connecting the device to mobile application, the user selects the muscle group for the training. Then the user must enter necessary information about the exercise, and can start the training. After training finishes, the user can review the training, and the system creates reports.

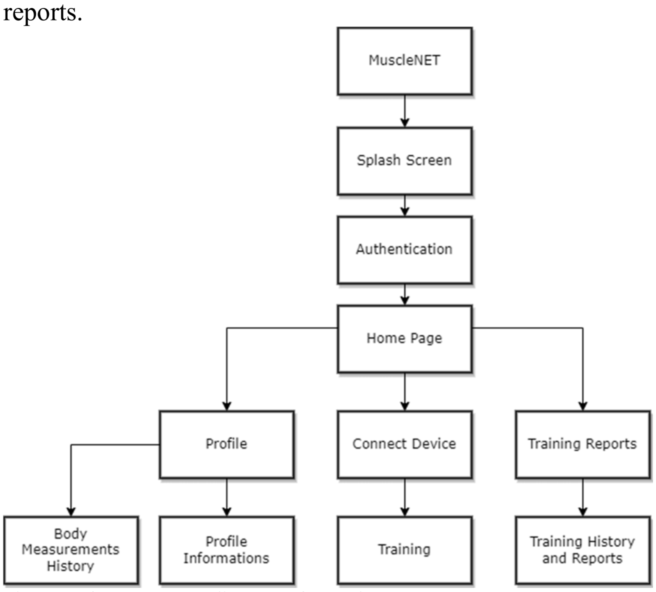

Fig. 2. Software system diagram of MuscleNET

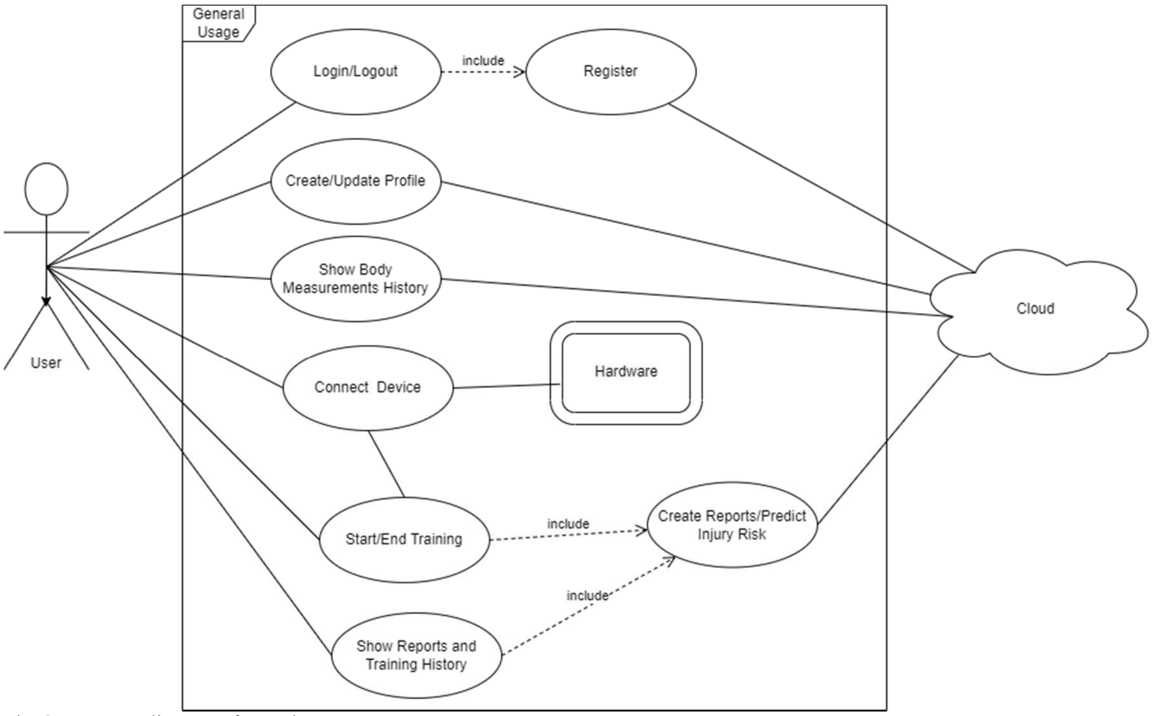

Fig. 3. Use case diagram of MuscleNET

Regarding the smart side of MuscleNET, different machine learning approaches have been adopted for unsupervised clustering of trainings, and supervised injury detection/prediction and training quality classification (Fig.5). Supplying the raw sEMG data to a machine learning system might be useless without using well-developed signal feature extraction. Because of this, features have been selected based on the available literature. We found out that there are two types of features which are either in time-domain or frequency-domain.

There are 16 useful features in time-domain which are Mean Absolute Value (MAV), Mean Absolute Deviation (MAD), Variance (VAR), Average Amplitude Change (AAC), Root Mean Square (RMS), Simple Square Integrals (SSI), Integrated EMG (IEMG), Waveform Length (WL), Willison Amplitude (WAMP), Log Detectors (LOG), Difference Absolute Standard Deviation Value (DASDV), Myopulse Percentage Rate (MYOP), Modified Mean Absolute Value-1 (MMAV1), Modified Mean Absolute Value-2 (MMAV2), Zero Crossing (ZC), Slope Sign Change (SSC). Also, there are 4 other features in frequency-domain which are Autoregressive Coefficients (AR), Modified Median Frequency (MMDF), Modified Mean Frequency (MMNF), Modified Mean and Median Frequency (MDF). Time domain features are preferred to be used for straight-forward application, and for that purpose we selected 13 of the time-domain features (excluded features: MMAV2, ZC, and SSC) which are also used in the literature. Furthermore, the following attributes are also used as features: gender, lifted weight, and power level (Biceps size(cm) * Biceps size(cm) / Body Mass Index).

K-means has been deployed as the unsupervised algorithm to detect if there is a grouping between different signals or not. UMAP (Uniform Manifold Approximation and Projection) is used to decrease the dimensionality of our features and to plot them for analysis. On the other hand, K-means will be used to cluster different signals in 3 classes, results will be compared with our supervised models. Scikit-learn and UMAP libraries will be used for the development of these algorithms. The heart of our work is supervised learning. We aim to prevent injury and measure training quality to give a recommendation for users. Deep neural networks are used to make predictions on a single training signal. Features of a single training signal are used to get predictions from MuscleNET Injury Detection and MuscleNET Training Quality Detection networks.

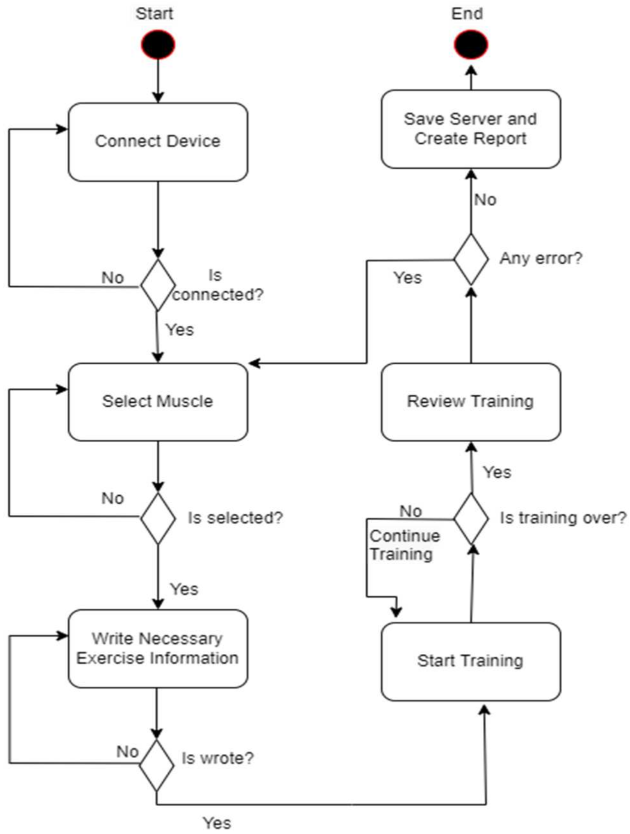

Fig. 4. Training activity diagram of MuscleNET

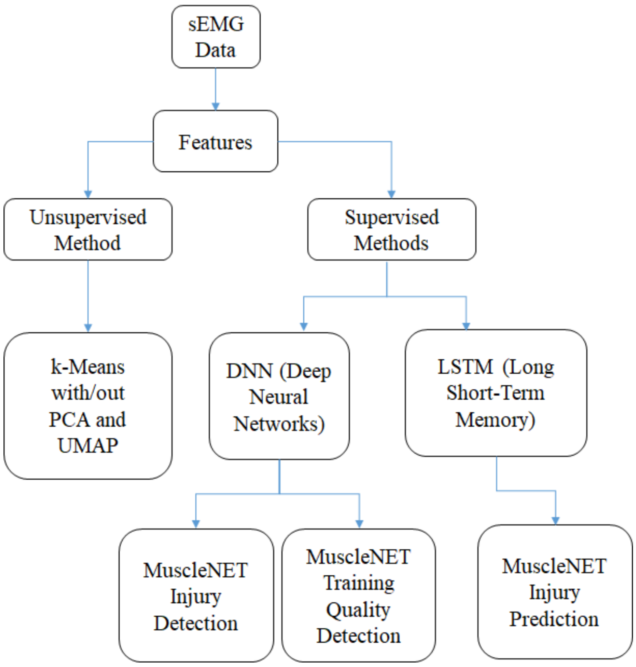

Fig. 5. Machine learning in MuscleNET

The mission of MuscleNET Injury Detection is to analyze features to detect if there is a risk of injury. If your movement was risky, the system will warn the user. MuscleNET Training Quality Detection again analyses features and make predictions about your training, if the user does not perform enough, the network will recommend increasing weight to keep continue for the development of your body. If the user pushes themselves too hard, again the network will warn the user about that movement was too much. As it can be predicted, the system will respond "Well" for good training. On the other hand, our LSTM network MuscleNET Injury Prediction analyzes the last five training signals to make injury prediction to warn the user. The structure of the LSTM networks will allow the network to remind/detect information from previous training signals. Our supervised algorithms are implemented with TensorFlow. The architecture for those neural networks are specified based on numerous simulations. Fig. 6 displays the architecture for the Injury Detection and Training Quality Classification.

Injury Detection network consists of 3 hidden layers with 48 hidden units, 1 hidden layer with 12 hidden units, and a SoftMax layer that outputs a 2-dimensional matrix representing the probabilities of each class: NoInjury and Injury. Training Quality Detection structure is very similar to Injury Detection structure but one extra SoftMax output is added to represent the probabilities of each class: "too low weight, use higher weight for improvement", "keep continue for improvement", and "too high weight, use less weight for improvement."

MuscleNET Injury Prediction deploys LSTM (Long Short-Term Memory) networks to utilize information from previous samples with the packets of 5 timesteps and using multiple samples to output one prediction. After the LSTM layer, we provide a SoftMax layer that outputs a 2-dimensional matrix representing probabilities of 2 classes which are "no injury risk in future, keep continue" and "high injury risk in future, stop training". Normally LSTM networks would provide much greater structures, but we limited time steps to 5 due to the limited time for data collection. The structure of the network is given in Fig. 7.

IV. Results

The final prototype of the product is given in Fig. 8. This prototype has been used to collect training data from 19 users (5 female, 14 male) where a total of 213 training sets are recorded with the corresponding "user based quality assessment." The users have given permission so that their measurement data can be stored and processed. According to GDPR rules, no data has been stored about the actual identity of the users (except the authors). Those training data are visualized after UMAP has been applied for dimensionality reduction (Fig. 9) Red points represent "low weight, increase it for improvement", blue points represent "ideal weight, keep continue for improvement", and yellow points represent "too high weight, decrease it for improvement".

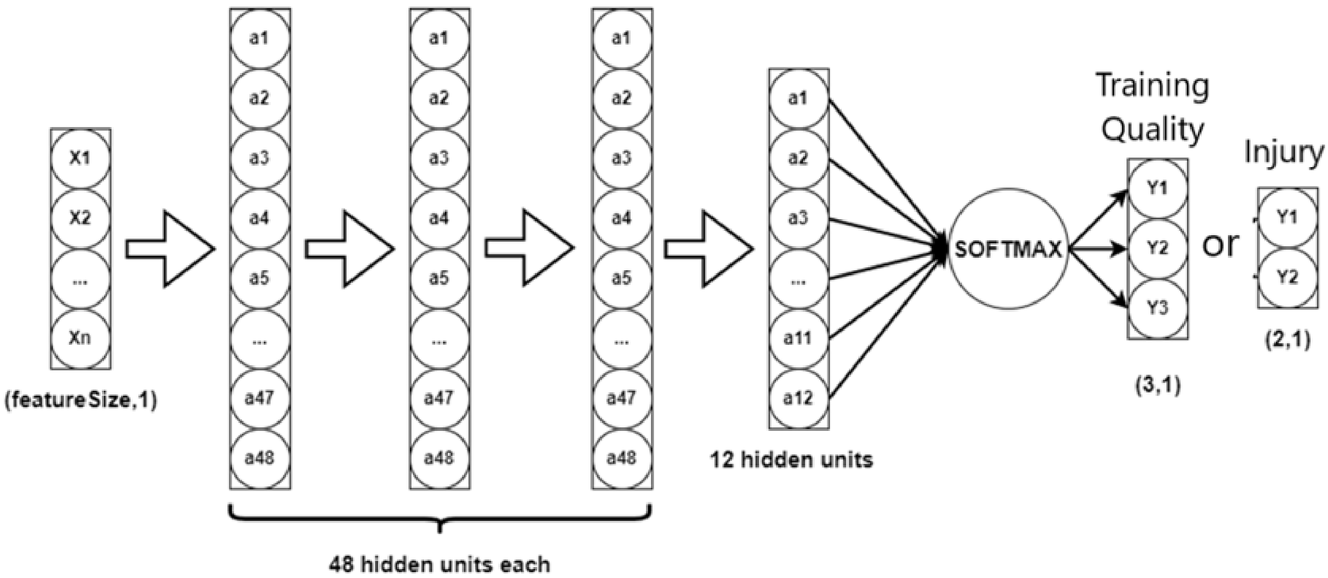

Fig. 6. Deep neural network for injury detection and training quality classification in MuscleNET

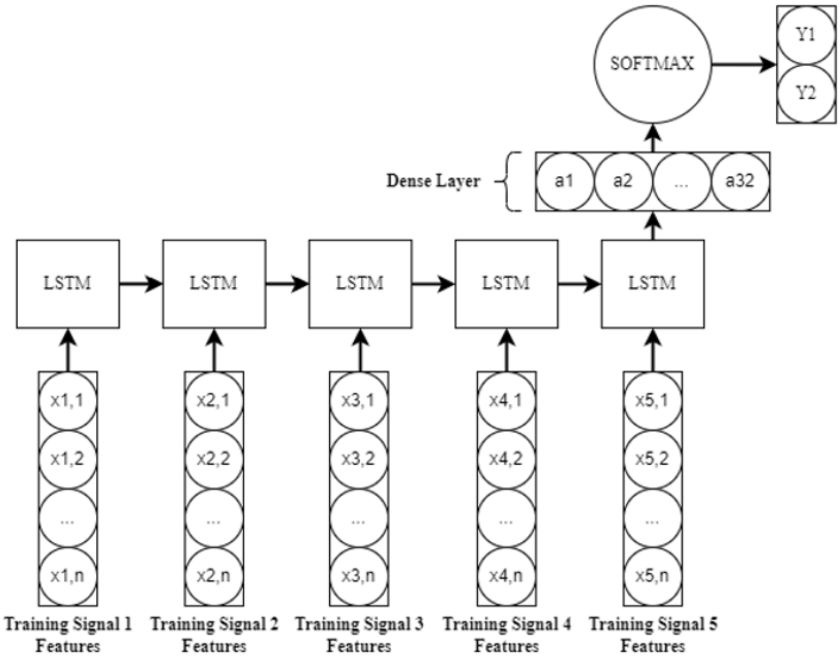

Fig. 7. LSTM network for injury prediction in MuscleNET

There are 89 samples for class 0 (red), 72 samples for class 1 (blue), and 52 samples for class 2 (yellow). We see that red and yellow classes are separated well. However, blue class has a wider distribution. We notice that scaling the data causes a clear separation among classes when compared to the unscaled data. Then the data has been applied SMOTE (Synthetic Minority Oversampling Technique) to have equal number of samples in each class, namely 89 for each class. Then the data are clustered using K-means algorithm into three groups. The groups are checked for their classes and Table I summarizes the accuracy of the groupings for different configurations.

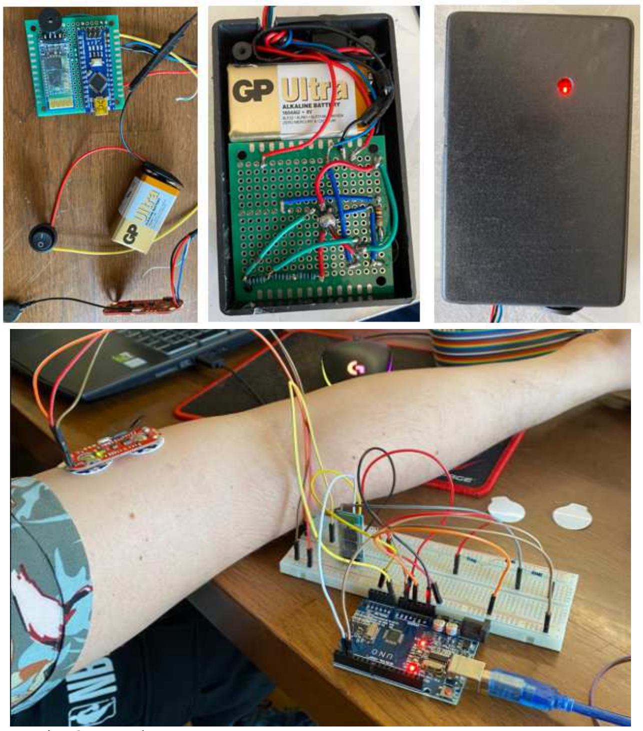
Fig. 8. MuscleNET prototype

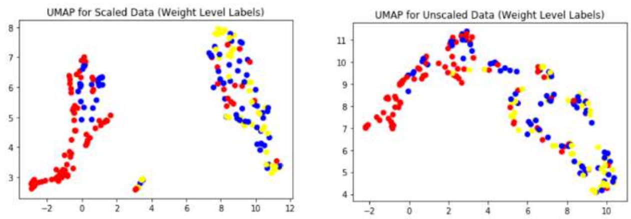

Fig. 9. Training data after UMAP applied

TABLE I. K-MEANS CLUSTERING RESULTS

Configuration	Accuracy (%)
No reduction in feature size	58
Reduce feature size to two using PCA	54
Reduce feature size to two using UMAP	21
Reduce feature size to two using PCA after SMOTE	27

The results indicate that reduction of dimension and SMOTE do not contribute to the performance.

For supervised training of "Training Quality", ReLu activation function is used in our hidden layers, and a SoftMax layer with 3 outputs is used as the last layer of the network. SoftMax outputs probabilities of each class. Adam optimizer which has built-in regularization capabilities, has been used to prevent overfitting. Categorical Cross Entropy is chosen as the loss function because we used a SoftMax output. We have a total of 6147 parameters that are going to learned by the model. With 3000 epochs, we reached 100% training and 81% test accuracy in 91 seconds where 70% of data are used for training and 30% are used for test. We also applied SMOTE to increase the sample space. Table II summarizes the accuracy, precision, recall, and F-score values of the network.

TABLE II. RESULTS OF MUSCLENET TRAINING QUALITY PREDICTION

	MuscleNET Training Quality Detection	
	Original Data	Oversampled Data
Training Accuracy	100%	96,00%
Testing Accuracy	**81%**	**85%**
False Negative	12	12
False Positive	11	12
True Negative	117	150
True Positive	52	69
Precision	82,54%	85,20%
Recall	81,25%	85,20%
F1-Score	**81,89%**	**85,20%**

The results indicate that SMOTE has improved the performance of the network.

The mobile application developed can be used to monitor the training progress and it can also generate reports for past trainings. Sample screens of the application are given in Fig. 10. Furthermore, sample measurements are displayed in Fig. 11 which are obtained from two of the authors.

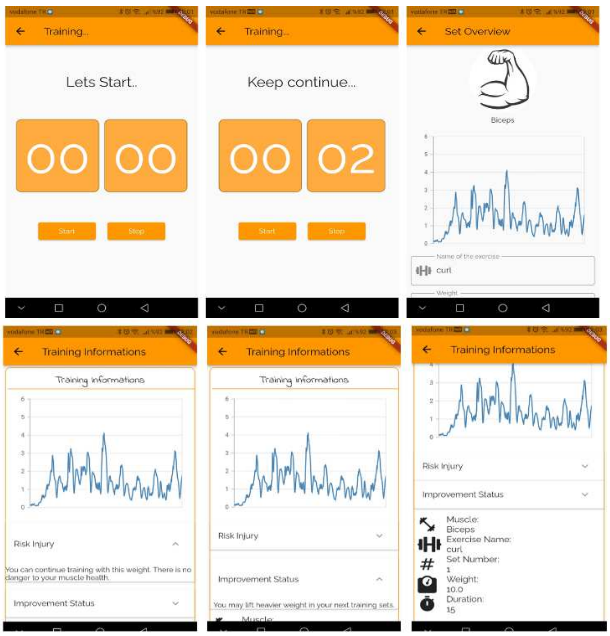

Fig. 10. Training screens in MuscleNET application

V. CONCLUSION

The results obtained for the limited set of data are promising for training quality prediction as the training and test accuracy values are 100% and 81% respectively. There needs to be more data to be collected and labeled for injury detection and injury prediction. The mobile application will be a handy tool for real time monitoring and reporting. A deep learning network of moderate size seems to be efficient for handling the supervised learning of muscle activity monitoring. In general, oversampling techniques improve both clustering and classification performance. SMOTE has improved classification accuracy for test data to 85%.

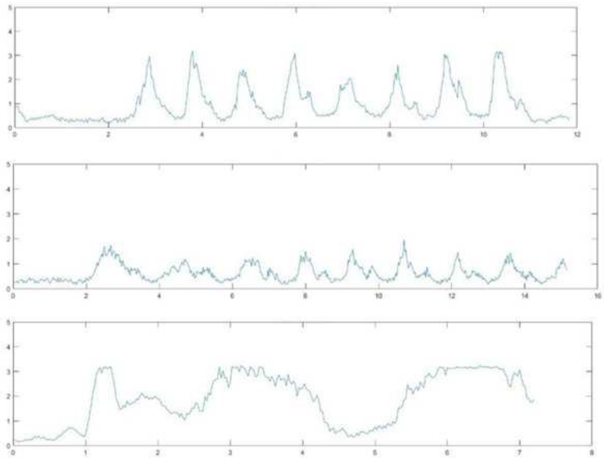

Fig. 11. Sample measurements; top-to-bottom: Tuğberk lifting 3kg; Kazım lifting 3kg; Tuğberk lifting 15kg; the horizontal axis is time in seconds, the vertical axis is the scaled strength value.

REFERENCES

[1] Q. Li and B. Li, “The Muscle Activity Detection from Surface EMG Signal Using the Morphological Filter,” *Applied Mechanics and Materials*, vol. 195–196, pp. 1137–1141, Aug. 2012.

[2] O. Amft, G. Troster, P. Lukowicz, and C. Schuster, “Sensing Muscle Activities with Body-Worn Sensors,” in *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, pp. 138–141, 2006.

[3] B. Zhou, M. Sundholm, J. Cheng, H. Cruz, and P. Lukowicz, “Never skip leg day: A novel wearable approach to monitoring gym leg exercises,” in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–9, March 2016.

[4] P.A. Karthick, D. M. Ghosh, S. Ramakrishnan, “Surface electromyography based muscle fatigue detection using high-resolution time-frequency methods and machine learning algorithms,” *Computer Methods and Programs in Biomedicine*, vol. 154, pp. 45-56, 2018.

[5] J.-W. Lee, M.-J. Shin, M.-H. Jang, W.-B. Jeong, and S.-J. Ahn, “Two-stage binary classifier for neuromuscular disorders using surface electromyography feature extraction and selection,” *Medical Engineering & Physics*, vol. 98, pp. 65-72, 2021.

[6] R. Chowdhury, M. Reaz, M. Ali, A. Bakar, K. Chellappan, and T. Chang, “Surface Electromyography Signal Processing and Classification Techniques,” *Sensors*, vol. 13, no. 9, pp. 12431–12466, Sep. 2013.

[7] B. Karlık, “Machine Learning Algorithms for Characterization of EMG Signals,” *International Journal of Information and Electronics Engineering*, vol. 4, no. 3, 2014.

[8] C. Seyidbayli, F. Salhi, and E. Akdogan, “Comparison of Machine Learning Algorithms for EMG Signal Classification,” *Periodicals of Engineering and Natural Sciences*, vol. 8, no. 2, pp. 1165–1176, Jun. 2020.

[9] C. Tepe and M. Erdim, “Classification of surface electromyography and gyroscopic signals of finger gestures acquired by Myo armband using machine learning methods,” *Biomedical Signal Processing and Control*, vol. 75, 2022.

[10] H. Nodera, Y. Osaki, H. Yamazaki, A. Mori, Y. Izumi, and R. Kaji, “Deep learning for waveform identification of resting needle electromyography signals,” *Clinical Neurophysiology*, vol. 130, no 5, pp. 617-623, May 2019.

[11] Y. Zhou, C.Y. Chen, M. Cheng, Y. Alshahrani, S. Franovic, E. Lau, G.H. Xu, G.X. Ni, J.M. Cavanaugh, S. Muh, and S. Lemos, “Comparison of machine learning methods in sEMG signal processing for shoulder motion recognition,” *Biomedical Signal Processing and Control*, vol. 68, Apr. 2021.

[12] Z.M. Lv, F.Y. Xiao, Z. Wu, Z.S. Liu, and Y. Wang, “Hand gestures recognition from surface electromyogram signal based on self-organizing mapping and radial basis function network,” *Biomedical Signal Processing and Control*, vol. 68, Apr. 2021.

[13] C. Mokri, M. Bamdad, and V. Abolghasemi, “Muscle force estimation from lower limb EMG signals using novel optimised machine learning techniques,” *Medical and Biological Engineering and Computing*, vol. 60, no 3, pp. 683-699, Jan. 2022.

[14] SparkFun, “MyoWare Muscle Sensor” https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MyowareUserManualAT-04-001.pdf