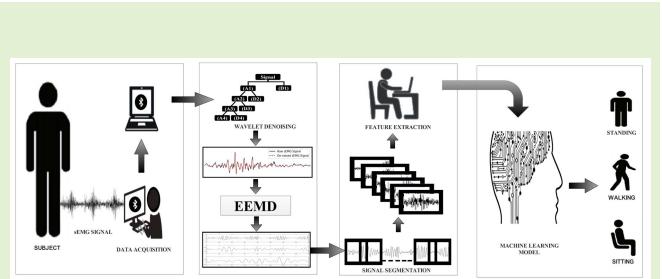


A Hybrid WD-EEMD sEMG Feature Extraction Technique for Lower Limb Activity Recognition

Ankit Vijayvargiya^{ID}, Student Member, IEEE, Vishu Gupta^{ID}, Member, IEEE,
Rajesh Kumar^{ID}, Senior Member, IEEE, Nilanjan Dey^{ID}, Senior Member, IEEE,
and João Manuel R. S. Tavares^{ID}, Member, IEEE

Abstract—Classification and analysis of surface EMG (sEMG) signals have been of particular interest due to their numerous applications in the biomedical field. They can be used for the diagnosis of neuromuscular diseases, kinesiological studies, and human-machine interaction. However, these signals are difficult to process due to their noisy nature. To overcome this problem, a hybrid of wavelet with ensemble empirical mode decomposition pre-processing technique called WD-EEMD is proposed for classifying lower limb activities based on sEMG signals in healthy and knee abnormal subjects. First, Wavelet De-noising is used for filtering out white Gaussian Noise (WGN) and unwanted signals (contribution of other muscle signals). Next, an Ensemble Empirical Mode Decomposition is used for filtering out power line interference (PLI) and baseline wandering (BW) noises, followed by extraction of a total of nine time-domain features. Finally, the performance parameters of the Linear Discriminant Analysis (LDA) classifier are calculated with a 3-fold cross-validation technique. This study involves 11 healthy and 11 individuals with a knee abnormality for three different activities: walking, flexion of the leg up (standing), and leg extension from sitting position (sitting). Different pre-processing techniques similar to that of WD-EEMD were compared. It was observed that the proposed method achieves an average classification accuracy of 90.69% and 97.45% for healthy subjects and knee abnormal subjects, respectively.

Index Terms—Biomedical signal analysis, EMG classification, WD-EEMD, ensemble empirical mode decomposition, wavelet denoising, linear discriminant analysis, gait activities.



I. INTRODUCTION

KNEE problems are defined as a sensation of discomfort in the knee that are caused by lack of proper warm-up, poor form during physical activities, or osteoarthritis. According to [1], one out of every four individuals have joint symptoms or arthritis because of an underlying condition

Manuscript received May 13, 2021; revised July 3, 2021; accepted July 3, 2021. Date of publication July 8, 2021; date of current version September 15, 2021. This work was supported by the Visvesvaraya Ph.D. Scheme, MeitY, Government of India, under Grant MEITY-PHD-2942. The associate editor coordinating the review of this article and approving it for publication was Dr. Ravibabu Mulaveesala. (*Corresponding author: Ankit Vijayvargiya*.)

Ankit Vijayvargiya and Rajesh Kumar are with the Department of Electrical Engineering, Malaviya National Institute of Technology Jaipur, Jaipur 302017, India (e-mail: ankitvijayvargiya29@gmail.com; rkumar.ee@mnit.ac.in).

Vishu Gupta is with the TensaX Innovation Lab, Jaipur 302017, India (e-mail: vishu.gupta0607@gmail.com).

Nilanjan Dey is with the Department of Computer Science, JIS University, Kolkata, West Bengal 700109, India (e-mail: neelanjan.dey@gmail.com).

João Manuel R. S. Tavares is with the Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial, Departamento de Engenharia Mecânica, Faculdade de Engenharia, Universidade do Porto, 4200-465 Porto, Portugal (e-mail: tavares@fe.up.pt).

Digital Object Identifier 10.1109/JSEN.2021.3095594

such as degenerative arthritis of the knee. The knee joint is a synovial joint that is formed with several surrounding structures, including ligaments, bones, cartilage and tendons, to perform its functions [2]. Any external harm to any of these can result in knee abnormality [3]. Knee osteoarthritis, cerebral palsy are some knee abnormalities that cause knee pain and reduce the quality of daily life of a person [4], [5].

Assistive devices can be used to enhance the quality of the daily life of an unhealthy person. These devices are categorized into: orthosis and prosthesis. The prosthesis is an artificial limb for a missing body part while orthosis is used to improve the functionality of moving body parts for weak person. These devices are also classified based on power: active and passive. An active assistive device uses a power source to activate the actuators while a passive device has no power source. So, automatic control is possible with active devices while passive devices cannot be similarly controlled. As usual, the active devices could be body-powered, or electric-powered. Electric-powered lower limb assistive devices have been widely used, and may be operated by a pressure resistor, strain gauge, micro-switch, electroencephalogram signals (EEG), electromyogram signals (EMG), etc. In recent years, EMG signals have widely been

used for controlling assistive devices because it allows the recognition of movement in advance [6] and provides faster detection of the signal variation [7]. Invasive or non-invasive techniques are used to acquire EMG signals from muscles in which non-invasive techniques are better than invasive techniques as no medical supervision is required and infection is also negligible during placement of non-invasive (sEMG) electrodes [8]. In the non-invasive technique, electrical activity produced by skeletal muscle is collected through a sEMG sensor.

The design of a neuro-fuzzy controller has been proposed by Kiguchi *et al.* for the upper limb robotic exoskeleton, which is an upper limb assistive device [9]. The purpose of this exoskeleton is to enhance the quality of life of the injured, disabled, elderly, and physically challenged people. A sEMG based low-cost elbow joint-powered exoskeleton was developed for bicep brachii strength augmentation by Krasin *et al.* [10]. The user who worn this exoskeleton can freely move in normal condition; however, when the biceps muscle is in an underloaded condition, then this muscle produces a different EMG signal which causes the exoskeleton to automatically switch to the assistive lifting movement and then returns to its normal condition once the muscle is relaxed. A low-cost sEMG controlled upper limb prosthetic arm has been developed by Sharmila *et al.* [11]. For automatic control of the prosthetic arm, the sEMG signal is acquired for different hand movements from the users and features are extracted after preprocessing the data. Based on these features, the signal is classified for various hand movements. After recognizing the hand movements by the classifier, a control signal is generated and given to the motor of the prosthetic arm to perform the intended movements.

Over the past several years, applications of the upper limb using sEMG signal have been focused on by researchers compared to the applications of the lower limb, because acquiring the sEMG signals of lower limb muscles is more complex due to the contribution of multiple motor units at a time and their dependency on neuromuscular activity, physiological and anatomical properties of the involved muscles. Recognition of various gait activities based on the sEMG signal of the lower limb has an important role in controlling the exoskeleton for the knee abnormal person or in the prosthesis control for the lower limb amputee. Neural network based myopathy and neuropathy classification using sEMG signal was proposed in [12]. Kugler *et al.* have recognized Parkinson's disease using sEMG signal [13]. The classification of six different movements of lower limb using machine learning classifiers was studied by Shukla *et al.* [14]. Vijayavargiya *et al.* [15] worked on identifying knee abnormality in the subject, by collecting imbalanced surface EMG signals data as a result of different sized signal lengths, of healthy and unhealthy individuals. To realize multi-step classification, various schemes were adopted and employed a computational classifier for conclusive recognition. In other research [16], they used various machine learning algorithms to provide a comparative analysis between them. In this, walking, standing, and sitting three lower limb activities are observed and the movements are recorded in terms of sEMG signals for the purpose of classification of subjects with knee abnormality. Various different steps are performed

on sEMG data to achieve classification. And five machine learning algorithms including Decision tree, Extra tree, KNN (k-nearest neighbor), SVM (support vector machine), and Random forest to provide performance comparison in terms of accuracy, sensitivity, specificity, and F1-score to identify knee abnormality in unhealthy subjects. A. Gautam *et al.* [17] introduced a novel classification approach to incorporate lower limb activities accompanying prognostication of the knee joint angle. Convolutional Neural Network (CNN) and LSTM combinedly make architecture to classify lower limb activities where CNN is used for extracting features from sEMG signal data and LSTM is used for joint angle prediction and to interpret the features follow up dense layer is connected for classification. Combining these three blocks they have proposed MyoNet model to predict lower limb activities (out of walking, standing, and sitting) simultaneously with joint angle prediction.

Despite several successful applications, sEMG based knee activity recognition remains a challenging problem due to their noisy nature. Due to the mixing of different noise signals or artifacts such as inherent noise, ambient noise and motion artifacts, the identity of an actual sEMG signal originating in the muscle is lost. For this purpose, different methods have been proposed for sEMG noise elimination by the researchers. The frequency range of the sEMG signal is 10 to 500 Hz and an amplitude range of 0 to 10 mV [18]. The conventional filtering methods such as low-pass, high-pass, and band-pass filters can be used to remove the noises that are not in the range of the sEMG signals. However, they are unable to remove random noises such as white Gaussian noise that is in the range of active sEMG signal spectrum band. The frequency ranges of the motor unit in an sEMG signal can be represented by Wigner-Ville distribution (WVD) [19]. It exhibits excellent localization properties, but has a cross-term effect and thus, cannot deal with multi-component signals. Wavelets overcome the limitations caused by WVD. It does not have a cross-term and thus, has the capability of handling multi-resolution problems. Various adaptive filtering [20] techniques, like Wiener filtering [21], based on the Fourier approach have also been proposed for the removal of noise in surface myoelectric signals. As mentioned in many of earlier works [22], [23], the Wavelet Transform (WT) has been used in processing sEMG signals, as it is an extremely flexible approach to signal decomposition with a lot of choices in wavelet functions. The properties of the wavelet function and the characteristic of the signal to be analyzed need to be more carefully matched, before the classification process [22]. WT is also used for de-noising these signals by selecting an optimal wavelet function for them [24]. The use of wavelets has also gained widespread acceptance when extracting features or analyzing signals, especially for sEMG signals [25], [26]. They have advantages over classical techniques like Fourier transform or autoregressive models in analyzing physical situations where the signal contains discontinuities and sharp spikes [27], [28].

Studies on the decomposition of sEMG signals have been done since the 1960s [29], [30], where efforts have been made to segregate individual contributing motor potentials. A powerful technique called Empirical Mode Decomposition (EMD) was introduced [31] in 1998, which decomposed sEMG signals

into Intrinsic Mode Functions (IMFs). EMD proved to be very effective and useful yet there were a few limitations associated with it. Improvements have been made in this algorithm, one of which is the Ensemble EMD (EEMD) [32].

Motivated by the need for neural control of the lower limb exoskeleton or prosthesis, investigated different types of mobility tasks that could be correctly identified using sEMG signals obtained from the leg muscles. The results of this study will help in the future development of neural-controlled artificial exoskeleton or lower limbs prosthesis with versatile activities for injured or disabled persons. Due to mixing of various noises in the sEMG signal, sEMG based knee activity recognition is a challenging problem. Therefore, in this article, a hybrid pre-processing technique WD-EEMD is proposed for the analysis of sEMG signal for the recognition of lower limb activities.

The major contributions of this research are:

- 1) Identification of the lower limb activity using the sEMG signals obtained from the leg muscles in individuals with and without knee abnormality.
- 2) A hybrid pre-processing technique WD-EEMD (Wavelet Denoising - Ensemble Empirical Mode Decomposition) which is proposed for the analysis of sEMG signals for lower limb activities recognition.
- 3) A total of nine time-domain features are extracted from the sEMG signals of the four lower limb muscles using the overlapping windowing technique.
- 4) Performance parameters of LDA classifier are calculated for the recognition of gait activities in healthy and knee abnormal subjects.

The structure of the article is as follows: a survey of data collection is given in section II. Section III presents the proposed methodology. Section IV comprises the results and discussion. Conclusion and future scope are given in Section V.

II. DATA ACQUISITION

In this study, we have considered publicly available datasets from UCI for the classification of lower limb activity [33]. It consists of 22 volunteer participants above the age of 18 years, among which 11 participants appear to be fit and the rest exhibit knee abnormalities. No preceding case history was found regarding pain or injury in the knee of healthy participants. Among participants with knee abnormalities, four were affected with a meniscus injury, six suffered from anterior cruciate ligament (ACL), and one was encountered with sciatic nerve injury. The left leg for healthy subjects and affected limb for the knee abnormal subjects were chosen for acquiring the EMG signal. All the subjects undergo three different exercises: walking, flexion of the leg up (standing), and leg extension from sitting position (sitting) to analyze the behavior of the knee muscle. The data was recorded by four surface electrodes around the muscles: biceps femoris, rectus femoris, semitendinosus and vastus medialis, and the goniometer was attached to the external side of the knee joint. The used data collection equipment for obtaining the sEMG data was a MWX8 by Biometrics Ltd. that has 4 analog channels and 8 digital channels, out of which 4 for sEMG and 1 for

goniometry were used. All the data was stored directly from MWX8 storage to the computer and transmitted to the Datalog software by the bluetooth adapter in real-time. The sampling frequency was of 1000 Hz and a 14-bit resolution was used. This study only focuses on the effect of EMG signals of the lower limb muscles during the lower limb activities, so only the sEMG signals are considered.

III. METHODOLOGY

This section provides a description of the methods used in building the proposed methodology.

A. Wavelet De-Noising (WD)

To remove noise from the sEMG signal, a technique called the thresholding method is used based on wavelets. Wavelets can be visualized as small waves or ripples that have a very short and finite period. In the wavelet analysis method, first is selected a wavelet function that is called mother wavelet. Then, the low frequency version of wavelet is used to perform frequency analysis while the high frequency version of wavelet is used for temporal analysis [27]. Different mother wavelets are generated from a single basic wavelet $\psi(t)$, by scaling and translation [34]:

$$\psi_{s,\tau} = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right). \quad (1)$$

There are various kinds of mother wavelets and each has its own characteristics. Some of the most popular wavelet families are: Haar, Daubechies, Coiflet and Symlet.

When performing Discrete Wavelet Transform (DWT), wavelet coefficients can be generated by passing the signal through high-pass (detail coefficients) and low-pass (approximate coefficients) filters. The number of detail coefficients generated depends on the adopted level of decomposition.

After the wavelet decomposition of a signal, approximate and detailed coefficients are obtained. To eliminate the noise, small signal details can be excluded without any loss of important information. This, thresholding sets all coefficients to zero that are less than a particular threshold [14]. There are various modes for it such as soft and hard thresholding; however, both present their challenges. Garotte threshold function was proposed to overcome the shortcomings in the soft and hard thresholding methods. The universal threshold, that is used in this study, is defined as:

$$\lambda = \sigma \sqrt{2 \ln(N)}, \quad (2)$$

where $\sigma = (\text{MAD})/0.6745$, with MAD referring to the Median Absolute Deviation of the wavelet coefficient and N is the length of signal.

As studied in [26], db7 from the Daubechies family is used till the fourth decomposition level in this work, and the decomposition is selected upto four level where one level of approximate coefficients and four levels of details coefficients are obtained. Garotte thresholding is applied on the second detail coefficient level (D2).

TABLE I
EXTRACTED SEMG FEATURES AND THEIR MATHEMATICAL FORMULATION

	Extracted Feature	Mathematical formulation
1	Mean Absolute Value (MAV)	$\frac{1}{N} \sum_{i=1}^N x_i $ where x_i is a sample of the sEMG signal
2	Root Mean Square (RMS)	$\sqrt{\frac{1}{N} \sum_{i=1}^N x_i ^2}$
3	Zero Crossing (ZC)	where $f(x_i) = \begin{cases} 1 & \text{if, } (x_i > 0 \text{ and } x_{i+1} < 0) \\ & \text{or } (x_i < 0 \text{ and } x_{i+1} > 0) \\ 0 & \text{otherwise} \end{cases}$
4	Slope Sign Change (SSC)	where $f(x_i) = \begin{cases} 1 & \text{if, if, } (x_i > x_{i-1} \text{ and } x_i > x_{i+1}) \\ & \text{or } (x_i < x_{i-1} \text{ and } x_i < x_{i+1}) \\ 0 & \text{otherwise} \end{cases}$
5	Variance (VAR)	$\frac{1}{N-1} \sum_{i=1}^N x_i^2$
6	Difference Absolute Standard Deviation Value (DASDV)	$\sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} (x_{i+1} - x_i)^2}$
7	Average Amplitude Change (AAC)	$\frac{1}{N} \sum_{i=1}^{N-1} x_{i+1} - x_i $
8	Skewness (Skew)	$\frac{E[(x-\mu)^3]}{\sigma^3}$
9	Kurtosis (Kurt)	where σ is the Standard deviation of the signal dataset, μ = Mean of the dataset and E is the Expected value estimator of the dataset.

B. Ensemble Empirical Mode Decomposition

Due to the non-linearity and non-stationarity of sEMG signals, decomposition techniques that assume a process to be linear and stationary may yield deceptive results [35]. Empirical Mode Decomposition (EEMD) is a powerful tool for decomposing non-stationary and non-linear signals with complicated spatial and temporal structures into complete or almost orthogonal components, called Intrinsic Mode Functions [36]. IMF is a mono-component function or an oscillatory mode with one instantaneous frequency [37].

Using the EMD algorithm, a given signal $x(t)$ can be decomposed into a number of IMFs iteratively through a shifting algorithm. The procedure is as follows [31]:

- 1) An upper $u(t)$ and a lower $l(t)$ envelopes are created through interpolation (here, cubic) of all local maxima and minima of $x(t)$.
- 2) A running mean envelope $m(t)$ is then calculated using $m(t) = \frac{u(t)+l(t)}{2}$.
- 3) The mean envelope is then subtracted from the signal, which gives $k(t) = x(t) - m(t)$.
- 4) It is verified that whether $k(t)$ satisfies the following conditions of being an IMF:
 - The number of local extrema and zero crossings in the entire length of $k(t)$ must either be equal or at most differ by one;
 - At any point in the series, the mean value of $k(t)$ should be zero.
- 5) If $k(t)$ does not satisfy the conditions of being an IMF, $x(t)$ is replaced by $k(t)$ and sifting is continued, i.e., steps 1-4 are reiterated until the signal obtained satisfy the conditions. The sifting process can also be stopped if $k(t)$ is a monotonic function.

The original signal $x(t)$ may be obtained by summation of IMFs and the residual term:

$$x(t) = \sum_{m=1}^{M-1} IMF_m(t) + r_M(t) \quad (3)$$

where r_M is the residual term after extracting $M - 1$ IMFs.

However, the EMD algorithm has a problem of frequent appearance mode mixing due to its sensitivity to noise [38]. To alleviate this problem, the noise-assisted data analysis method EEMD was proposed, which describes the IMFs as the average of an ensemble of trials [36].

The procedure of Ensemble EMD is as follows [36]:

- 1) An equally distributed and independent white noise series having the same standard deviation is added to the targeted data [38].

$$x_m(t) = x(t) + n_m(t) \quad (4)$$

where $n_m(t)$ is the m_{th} white noise series added to $x(t)$, and $x_m(t)$ is the noise-added signal.

- 2) The signal $x_m(t)$ is then decomposed using the EMD algorithm into P IMFs $h_{p,m}$, where P denotes the number of IMFs and $h_{p,m}$ is the p_{th} IMF of the m_{th} trial.
- 3) The above two steps are repeated M times, but with different white noises, where M indicates the number of ensembles.
- 4) The final IMFs are calculated by averaging each of the P IMFs over M trials.

$$IMF_P = \frac{1}{M} \sum_1^M h_{p,m}, \quad p = 1, 2, \dots, P \\ m = 1, 2, \dots, M \quad (5)$$

The added noise cancels each other during averaging of corresponding IMFs in the EEMD process. The final IMFs remain

inside the natural dyadic filter windows, hence decreasing the chances of mode-mixing [32].

Wavelet De-noising is used for filtering out white Gaussian Noise (WGN) and unwanted signals, e.g. contributions of other muscle signals. It also helps in preserving critical features. Ensemble Empirical Mode Decomposition is used for filtering out power line interference (PLI) and baseline wandering (BW) noises. It is also used here for decomposing the signal to extract relevant features from it. In this article, the pre-processing technique using Wavelet De-noising (WD) and Ensemble Empirical Mode Decomposition (EEMD) is referred to as WD-EEMD, and together, they form an excellent hybrid approach.

C. Feature Extraction

After the WD-EEMD analysis, a sliding windowing technique is used to extract the features rather than considering the entire signal at once, due to its stochastic nature [39]. A sliding window procedure, of adjacent or overlapped nature, is used for segmentation [40]. In [41], the results exemplify that the overlapped windowing approach outperforms the disjoint or adjacent windowing scheme on the basis of classification accuracy. Segmenting the data into short windows clinches constant local mean, which assures stationarity of the data during the feature extraction process [42]. The overlapped windowing approach was implemented in this study to divide each of the temporal series into optimal segments or sub-frames of 256ms time windows and a leap of 64ms [43].

After preprocessing and signal segmentation, a feature extraction stage is used to emphasize the relevant structures of sEMG signals. The features must be selected in such a way that condenses the suitable information and maximally separate the output classes. Time-domain features are normally employed for muscle activity detection, muscle contraction, and onset detection whereas, frequency domain features are used to detect neural abnormalities and muscle fatigue.

In this research, nine time-domain features: Mean Absolute Value, Root Mean Square, Zero Crossing, Slope Sign Changes, Variance, Difference Absolute Standard Deviation Value, Skewness, and Kurtosis, are extracted from the three IMFs generated using the EEMD algorithm. As these features do not require any transformation and also due to their computational simplicity, they are generally quick and easily implemented for sEMG pattern recognition. A brief description of the features used is given in [Table I](#).

D. Feature Classification

As per the literature survey, many different machine learning models, like Support Vector Machine classifier and Decision Tree, have been used for sEMG classification; however, Linear Discriminant Classifier has been shown to be effective on applications to lower limb sEMG signal [43]. In this classifier, the discriminant property of LDA is enhanced during classification by maximizing the ratio of the between-class variance to the within-class variance. The within the class matrix (S_w)

and between-class matrix (S_b) can formally be defined as:

$$S_w = \sum_{k=1}^C \sum_{i=1}^{N_k} (f_i^k - \mu_k)(f_i^k - \mu_k)^T, \quad (6)$$

$$S_b = \sum_{k=1}^C (\mu_k - \mu)(\mu_k - \mu)^T, \quad (7)$$

where f_i^k is the i^{th} sample of class k , μ_k is the mean of class k , C is the number of classes, N_k is the number of samples in class k , and μ is the mean of all classes.

E. Performance Evaluation Metrics

To analyze the performance of machine learning models true positive rate, false positive rate, true negative rate, false negative rate are estimated, and their outcome helps in the building of confusion metrics. A confusion matrix facilitates the visualization of the performance of the model on the test dataset. In this study, we have three different classes: walking (W), sitting (S) and standing (T). The confusion matrix is formed as:

$$C = \begin{bmatrix} C_{WW} & C_{WS} & C_{WT} \\ C_{SW} & C_{SS} & C_{ST} \\ C_{TW} & C_{TS} & C_{TT} \end{bmatrix} \quad (8)$$

where C_{WW} is the number of cases in walking class predicted as walking, C_{WS} is the number of cases in walking class predicted as Sitting, C_{WT} is the number of cases in walking class predicted as standing and others can be defined similarly.

$$\text{Total number of data points} = C_{WW} + C_{WS} + C_{WT} + C_{SW} + C_{SS} + C_{ST} + C_{TW} + C_{TS} + C_{TT}$$

$$\text{Total number of cases as predicted walking } (P_W) = C_{WW} + C_{SW} + C_{TW}$$

$$\text{Total number of cases as actual walking } (A_W) = C_{WW} + C_{WS} + C_{WT}$$

$$\text{Total number of cases as predicted sitting } (P_S) = C_{WS} + C_{SS} + C_{TS}$$

$$\text{Total number of cases as actual sitting } (A_S) = C_{SW} + C_{SS} + C_{ST}$$

$$\text{Total number of cases as predicted standing } (P_T) = C_{WT} + C_{ST} + C_{TT}$$

$$\text{Total number of cases as actual standing } (A_T) = C_{TW} + C_{TS} + C_{TT}$$

The performance parameters for three class dataset are:

Accuracy - defined as the ratio of all correct predictions to the total number of instances in the dataset:

$$\text{Acc} = \frac{C_{WW} + C_{SS} + C_{TT}}{\text{Total number of dataset}}. \quad (9)$$

Specificity - the ratio of correct negative prediction to the total number of actual negative instances in the dataset.

Specificity for walking class:

$$SP_W = \frac{C_{SS} + C_{ST} + C_{TS} + C_{TT}}{A_S + A_T}. \quad (10)$$

Specificity for sitting class:

$$SP_S = \frac{C_{WW} + C_{WT} + C_{TW} + C_{TT}}{A_W + A_T}. \quad (11)$$

TABLE II

SUBJECT-WISE MEAN ACCURACY OF THE LDA CLASSIFIER IN PERCENTAGE WITH THE 3-FOLD CROSS VALIDATION TECHNIQUE FOR HEALTHY INDIVIDUALS DURING WALKING, SITTING AND STANDING, WITH DIFFERENT PRE-PROCESSING TECHNIQUES (BEST VALUES FOUND ARE IN BOLD)

Subject	Wavelet			EMD			WD-EEMD		
	Walking	Sitting	Standing	Walking	Sitting	Standing	Walking	Sitting	Standing
S1	96.43	87.84	91.03	82.14	52.70	61.54	85.71	97.30	97.44
S2	91.89	94.59	93.59	64.86	62.16	70.51	89.19	91.89	100.00
S3	71.43	68.75	91.86	45.71	43.75	46.51	68.57	37.50	72.09
S4	76.32	78.57	75.00	52.63	60.00	62.50	68.42	92.86	93.75
S5	93.94	94.87	92.75	66.67	71.79	81.16	93.94	82.05	92.75
S6	67.57	70.89	92.38	81.08	64.56	85.71	100.00	96.20	100.00
S7	97.56	92.68	94.78	95.12	90.24	97.39	95.12	100.00	95.65
S8	68.09	88.30	92.74	82.98	90.43	90.32	97.87	100.00	95.97
S9	65.38	100.00	99.08	69.23	100.00	78.41	61.54	98.31	95.41
S10	86.21	97.38	96.69	89.66	97.91	96.03	100.00	100.00	94.70
S11	93.10	88.89	96.51	62.07	53.70	79.07	75.86	79.63	90.70
Mean	82.54	87.52	92.40	72.01	71.57	77.20	85.11	88.70	93.50

TABLE III

SUBJECT-WISE MEAN ACCURACY OF THE LDA CLASSIFIER IN PERCENTAGE WITH THE 3-FOLD CROSS VALIDATION TECHNIQUE FOR KNEE ABNORMAL INDIVIDUALS DURING WALKING, SITTING AND STANDING, WITH DIFFERENT PRE-PROCESSING TECHNIQUES (BEST VALUES FOUND ARE IN BOLD)

Subject	Wavelet			EMD			WD-EEMD		
	Walking	Sitting	Standing	Walking	Sitting	Standing	Walking	Sitting	Standing
S1	84.84	96.55	96.51	60.60	75.86	83.72	98.48	87.93	98.83
S2	94.64	89.23	91.42	82.14	78.46	60.00	98.21	100	82.85
S3	96.90	84.21	98.01	96.90	90.35	99.00	100	91.22	100
S4	96.58	95.37	97.42	95.90	97.10	99.35	98.63	100	96.67
S5	95.79	93.06	100	96.63	98.02	99.23	99.16	99.00	100
S6	87.22	75.67	88.61	84.59	85.84	91.39	99.62	100	98.61
S7	97.5	84.81	91.67	86.21	88.61	90.28	100	91.14	95.83
S8	73.08	84.70	85.52	48.08	71.76	64.47	96.15	94.12	100
S9	86.14	67.86	78.13	85.54	76.78	87.5	100	99.10	97.91
S10	90.68	94.19	84.16	92.37	96.51	90.10	97.22	97.67	96.03
S11	94.37	86.00	98.46	95.07	75.00	89.23	100	100	97.69
Mean	90.70	86.51	91.81	84.00	84.91	86.75	98.86	96.38	96.77

Specificity for standing class:

$$SP_T = \frac{C_{WW} + C_{WS} + C_{SW} + C_{SS}}{A_W + A_S}. \quad (12)$$

Precision - gives the ratio of correct positive prediction to the total number of predicted positive instances in the dataset:

$$PR_i = \frac{C_{ii}}{P_i}. \quad (13)$$

Sensitivity (Recall) - the ratio of correct positive prediction to the total number of actual positive instances in the dataset:

$$RC_i = \frac{C_{ii}}{A_i}. \quad (14)$$

F-Score - measures the balance between precision and recall and is equal to harmonic mean of precision and recall:

$$F_i = \frac{2 * RC_i * PR_i}{RC_i + PR_i}, \quad (15)$$

where $i \in \{W, S, T\}$.

IV. RESULT AND DISCUSSION

The current study is based on the application of WD-EEMD based classification of lower limb movement using sEMG signals. A total of 11 healthy and 11 subjects suffering from knee abnormality were considered and the performance parameters for the classification of three different movements: walking, sitting and standing, were computed. The activity signal of the subjects was pre-processed using WD-EEMD and then divided into training and testing sets according to the 3-fold cross-validation technique. An overlapping window with a window size of 256 msec and 25% overlapping was chosen for the segmentation of the signal. Then, the LDA classifier was trained with the training dataset and the performance parameters of the model with the testing dataset calculated.

The K-fold cross-validation methodology is a resampling strategy that utilizes a compelled information test to assess the performance of AI models. In this technique, the samples are divided into groups of k equal size. After that, the training of the model is performed with $k-1$ groups of samples and the

TABLE IV

SUBJECT-WISE PERFORMANCE INDICES OF THE LDA CLASSIFIER IN PERCENTAGE WITH THE 3-FOLD CROSS VALIDATION TECHNIQUE FOR HEALTHY INDIVIDUALS DURING WALKING, SITTING AND STANDING, WITH DIFFERENT PRE-PROCESSING TECHNIQUES (BEST VALUES FOUND ARE IN BOLD)

Subject	Wavelet				EMD				WD-EEMD			
	Accuracy	Specificity	Sensitivity	F-Score	Accuracy	Specificity	Sensitivity	F-Score	Accuracy	Specificity	Sensitivity	F-Score
S1	92.86	96.19	92.64	92.93	74.29	86.64	73.40	73.71	95.71	97.75	95.08	95.41
S2	91.62	95.66	91.77	92.02	72.77	86.17	73.53	73.38	93.19	96.56	93.69	93.54
S3	94.64	97.36	93.04	93.34	96.30	98.14	95.42	95.66	98.15	99.02	97.08	97.66
S4	96.46	98.21	96.46	96.30	97.10	98.53	97.46	97.20	98.55	99.34	98.47	98.33
S5	96.38	98.21	96.29	95.92	97.65	98.92	97.96	97.43	99.36	99.71	99.39	99.31
S6	84.74	92.26	83.83	83.95	87.79	93.70	87.27	87.47	99.30	99.68	99.41	99.27
S7	91.34	95.74	91.33	91.24	88.52	93.98	88.36	88.97	95.67	97.87	95.66	95.58
S8	82.16	90.52	81.10	82.14	63.38	80.93	61.44	62.04	96.71	98.28	96.76	96.88
S9	78.61	88.36	77.38	78.34	83.42	91.32	83.28	83.63	99.20	99.52	99.01	99.20
S10	89.51	94.55	89.67	89.82	92.79	96.37	92.99	92.84	96.95	98.49	96.98	96.92
S11	93.55	96.74	92.94	93.13	87.63	93.77	86.43	86.67	99.19	99.57	99.23	99.26
Mean	90.17	94.89	89.68	89.92	85.60	92.59	85.23	85.36	97.45	98.71	97.34	97.40

TABLE V

SUBJECT-WISE PERFORMANCE INDICES OF THE LDA CLASSIFIER IN PERCENTAGE WITH THE 3-FOLD CROSS VALIDATION TECHNIQUE FOR KNEE ABNORMAL INDIVIDUALS DURING WALKING, SITTING AND STANDING, WITH DIFFERENT PRE-PROCESSING TECHNIQUES(BEST VALUES FOUND ARE IN BOLD)

Subject	Wavelet				EMD				WD-EEMD			
	Accuracy	Specificity	Sensitivity	F-Score	Accuracy	Specificity	Sensitivity	F-Score	Accuracy	Specificity	Sensitivity	F-Score
S1	90.56	94.97	91.76	90.72	61.11	79.30	65.46	62.18	95.56	97.83	93.48	93.50
S2	93.65	96.48	93.36	94.02	66.14	82.20	65.85	65.70	94.71	97.19	93.69	94.12
S3	81.07	89.45	77.35	78.42	45.56	72.22	45.33	44.56	61.54	79.64	59.39	57.72
S4	76.60	87.63	76.63	76.87	59.57	78.60	58.38	58.70	88.30	94.24	85.01	85.40
S5	93.89	96.93	93.85	93.21	74.44	86.94	73.21	72.46	88.33	94.74	89.58	86.94
S6	80.54	90.29	76.94	76.15	77.38	89.19	77.12	74.28	98.64	99.35	98.73	98.32
S7	94.54	96.82	95.01	95.27	94.54	96.76	94.25	94.91	97.06	98.64	96.92	96.46
S8	86.79	92.68	83.04	84.47	89.06	94.48	87.91	87.39	97.74	98.96	97.95	97.40
S9	96.05	97.77	88.16	90.95	88.36	94.36	82.55	81.25	93.28	96.51	85.09	86.23
S10	96.23	97.64	93.43	95.17	96.50	97.85	94.53	95.45	97.84	99.04	98.23	95.79
S11	93.49	95.97	92.83	93.61	68.05	81.74	64.95	65.82	84.62	91.61	82.06	82.62
Mean	89.40	94.24	87.49	88.08	74.61	86.69	73.59	72.97	90.69	95.25	89.10	88.59

testing is performed with k^{th} group of samples. This process is repeated for all the groups obtained from the input data.

Table II presents the subject-wise mean of the correct classification percentage for the three movements under study obtained from the sEMG data acquired from healthy subjects, whereas the performance concerning the knee abnormal subjects is given in **Table III**. The similar pre-processing techniques can be compared as to with and without knee abnormal subjects from the data in these tables, which confirms that WD-EEMD performed significantly better than the Wavelet Transform or EMD when applied individually.

Tables IV and **V** allow the comparison in terms of performance indices between the different pre-processing techniques under studied when applied to subjects with and without knee abnormalities. This comparison allows to conclude that the WD-EEMD pre-processing technique obtained the highest performance indices relatively to other techniques.

Many different methods have been proposed for lower limb activity recognition. Herrera-Gonzalez *et al.* have developed a classifier for the classification of three different exercises using MP-ANN with an accuracy of 88%, 94% and 92% for walking, sitting and standing tasks, respectively **Table VI** [44]. On the other hand, Zhang *et al.* have classified different lower limb movements of healthy subjects by using the Empirical Mode Decomposition based approach obtaining the results given in **Table VI** [45].

Naik *et al.* developed a classifier to classify the walking, sitting and standing activities with an accuracy of 96.14 and 86.17% for healthy subjects and subjects suffering from knee

TABLE VI
COMPARISON OF THE PERFORMANCE OBTAINED BY THE PROPOSED METHODOLOGY AGAINST THE ONES OBTAINED BY LITERATURE STUDIES WITH SAME DATASET

Approach	Subject	Walking	Sitting	Standing
EMD [18]	Healthy	64	67	69
MEMD [18]	Healthy	73	79	82
NA-EMD [18]	Healthy	79	83	83
MP-ANN [19]	Knee Abnormal	88	94	92
Transfer Learning based LRCN [20]	Healthy	98.2	97.7	98.4
	Knee Abnormal	92.8	92.3	92.2
ICA-EBM [5]	Healthy	96.0	96.2	96.2
	Knee Abnormal	86.6	86.4	85.5
Proposed Method	Healthy	85.11	88.70	93.50
	Knee Abnormal	98.86	96.38	96.77

abnormalities, respectively [43]. Gautam and collaborators introduced the transfer learning-based LRCN model to classify the walking, sitting and standing activities obtaining an accuracy of 98.2, 97.7 and 98.4% for healthy subjects and 92.8, 92.3 and 92.2% for individual suffering from knee abnormalities, respectively [17]. The sEMG data of lower limb muscles that we have considered here is the same as the other contributors. **Table VI** allows the comparative performance analysis between the proposed model against literature studies, which allows to conclude that the proposed WD-EMD based pre-processing technique gave high performance for lower limb activity recognition in knee abnormal subjects while in healthy subjects other techniques gave better results than the WD-EMD. The controlling of lower limb assistive devices, for example, is required for individuals with knee abnormality, so the proposed technique seems to be better than other literature methods.

V. CONCLUSION

In this article, a hybrid pre-processing technique called Wavelet Denoising - Ensemble Empirical Mode Decomposition (WD-EEMD) was proposed for the analysis of sEMG signals for recognition of lower limb activity in subjects with and without knee abnormality. Both WD and EEMD filter different types of noises commonly associated with the EMG signal, and hence provide an integrated approach to de-noising and decomposition the input signal. Another advantage of using EEMD is to decompose the signal into several IMFs to assist with the signal segmentation and feature extraction phases using the overlapping windowing technique. The results were compared with the ones obtained by similar pre-processing techniques with the hybrid approach proving to be superior than them. For performance evaluation, 3-fold cross-validation was implemented on the used dataset, and the proposed method achieved an average classification accuracy of 90.69 and 97.45% for healthy subjects and knee abnormal subjects, respectively.

There are still some extensions in the future for the proposed work. First, the used dataset includes data acquired with a relative low number of subjects. Hence, the proposed approach should be validated with a large number of the subject, which will reduce the biasing issue due to the use of a small dataset. Second, the proposed methodology was validated using an offline dataset, and so, further research can aim its validation using a real-time dataset for its clinical validation. Additionally, the other advanced machine learning algorithm can be implemented and one may also try to reduce the extracted features space by using feature reduction techniques.

COMPLIANCE WITH ETHICAL STANDARDS

Ethical approval: This article does not contain any study with human participants performed by any of the authors. Declaration of Competing Interest: The authors report no declarations of interest.

REFERENCES

- [1] A. Foundation. (2019). *Arthritis By The Numbers*. [Online]. Available: <https://www.arthritis.org/getmedia/e1256607-fa87-4593-aa8a-8db4f291072a/2019-abtn-final-march-2019.pdf>
- [2] Y. Sun, E. Teo, and Q. Zhang, "Discussions of knee joint segmentation," in *Proc. Int. Conf. Biomed. Pharmaceutical Eng.*, Dec. 2006, pp. 1–6.
- [3] P. Richeb  , X. Capdevila, and C. Rivat, "Persistent postsurgical pain: Pathophysiology and preventative pharmacologic considerations," *Anesthesiol., J. Amer. Soc. Anesthesiol.*, vol. 129, no. 3, pp. 590–607, 2018.
- [4] A. Vijayvargiya, P. L. Singh, S. M. Verma, R. Kumar, and S. Bansal, "Performance comparison analysis of different classifier for early detection of knee osteoarthritis," in *Sensors for Health Monitoring*. Amsterdam, The Netherlands: Elsevier, 2019.
- [5] M. Oskoui, F. Coutinho, J. Dykeman, N. Jette, and T. Pringsheim, "An update on the prevalence of cerebral palsy: A systematic review and meta-analysis," *Develop. Med. Child Neurol.*, vol. 55, no. 6, pp. 509–519, 2013.
- [6] B.-S. Yang and S.-T. Liao, "Fall detecting using inertial and electromyographic sensors," in *Proc. 36th Annu. Meeting Amer. Soc. Biomech.*, Gainesville, FL, USA, 2012, pp. 15–18.
- [7] J. Cheng, X. Chen, and M. Shen, "A framework for daily activity monitoring and fall detection based on surface electromyography and accelerometer signals," *IEEE J. Biomed. Health Informat.*, vol. 17, no. 1, pp. 38–45, Dec. 2012.
- [8] D. Farina and F. Negro, "Accessing the neural drive to muscle and translation to neurorehabilitation technologies," *IEEE Rev. Biomed. Eng.*, vol. 5, pp. 3–14, 2012.
- [9] K. Kiguchi, T. Tanaka, and T. Fukuda, "Neuro-fuzzy control of a robotic exoskeleton with EMG signals," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 481–490, Aug. 2004.
- [10] V. Krasin, V. Gandhi, Z. Yang, and M. Karamanoglu, "EMG based elbow joint powered exoskeleton for biceps brachii strength augmentation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2015, pp. 1–6.
- [11] K. Sharmila, T. Sarath, and K. Ramachandran, "EMG controlled low cost prosthetic arm," in *Proc. IEEE Distrib. Comput., VLSI, Electr. Circuits Robot.*, Aug. 2016, pp. 169–172.
- [12] R. Swaroop, M. Kaur, P. Suresh, and P. K. Sadhu, "Classification of myopathy and neuropathy EMG signals using neural network," in *Proc. Int. Conf. Circuit, Power Comput. Technol. (ICCPCT)*, 2017, pp. 1–5.
- [13] P. Kugler, C. Jaremenko, J. Schlauchetzki, J. Winkler, J. Klucken, and B. Eskofier, "Automatic recognition of Parkinson's disease using surface electromyography during standardized gait tests," in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2013, pp. 5781–5784.
- [14] P. K. Shukla *et al.*, "Human activity recognition using accelerometer and gyroscope data from smartphones," in *Proc. Int. Conf. Emerg. Trends Commun., Control Comput. (ICONC)*, Feb. 2020, pp. 1–6.
- [15] A. Vijayvargiya, C. Prakash, R. Kumar, S. Bansal, and J. M. R. Tavares, "Human knee abnormality detection from imbalanced sEMG data," *Biomed. Signal Process. Control*, vol. 66, Apr. 2021, Art. no. 102406.
- [16] A. Vijayvargiya, R. Kumar, N. Dey, and J. M. R. Tavares, "Comparative analysis of machine learning techniques for the classification of knee abnormality," in *Proc. IEEE 5th Int. Conf. Comput. Commun. Autom. (ICCCA)*, Oct. 2020, pp. 1–6.
- [17] A. Gautam, M. Panwar, D. Biswas, and A. Acharyya, "MyoNet: A transfer-learning-based LRCN for lower limb movement recognition and knee joint angle prediction for remote monitoring of rehabilitation progress from sEMG," *IEEE J. Translational Eng. Health Med.*, vol. 8, pp. 1–10, 2020.
- [18] N. Nazmi *et al.*, "A review of classification techniques of EMG signals during isotonic and isometric contractions," *Sensors*, vol. 16, no. 8, p. 1304, 2016.
- [19] A. L. Ricamato, R. G. Absher, M. T. Moffroid, and J. P. Tranowski, "A time-frequency approach to evaluate electromyographic recordings," in *Proc. 5th Annu. IEEE Symp. Comput.-Based Med. Syst.*, Jun. 1992, pp. 520–527.
- [20] T. W. Beck, J. M. DeFreitas, J. T. Cramer, and J. R. Stout, "A comparison of adaptive and notch filtering for removing electromagnetic noise from monopolar surface electromyographic signals," *Physiol. Meas.*, vol. 30, no. 4, p. 353, 2009.
- [21] G. Ascherio and P. Giszulich, "Denoising of surface EMG with a modified Wiener filtering approach," *J. Electromyogr. Kinesiol.*, vol. 20, no. 2, pp. 366–373, 2010.
- [22] D. K. Kumar, N. D. Pah, and A. Bradley, "Wavelet analysis of surface electromyography," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 11, no. 4, pp. 400–406, Dec. 2003.
- [23] R. Constable and R. Thornhill, "Using the discrete wavelet transform for time-frequency analysis of the surface EMG signal," *Biomed. Sci. Instrum.*, vol. 29, pp. 121–127, Jan. 1993.
- [24] A. Phinyomark, C. Limsakul, and P. Phukpattaranont, "An optimal wavelet function based on wavelet denoising for multifunction myoelectric control," in *Proc. 6th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol.*, vol. 2, 2009, pp. 1098–1101.
- [25] K. Englehart, B. Hudgins, P. A. Parker, and M. Stevenson, "Classification of the myoelectric signal using time-frequency based representations," *Med. Eng. Phys.*, vol. 21, nos. 6–7, pp. 431–438, 1999.
- [26] A. Phinyomark, C. Limsakul, and P. Phukpattaranont, "Application of wavelet analysis in EMG feature extraction for pattern classification," *Meas. Sci. Rev.*, vol. 11, no. 2, p. 45, Jan. 2011.
- [27] A. Graps, "An introduction to wavelets," *IEEE Comput. Sci. Eng.*, vol. 2, no. 2, pp. 50–61, Summer 1995.
- [28] M.-F. Lucas, A. Gaufriau, S. Pascual, C. Doncarli, and D. Farina, "Multi-channel surface EMG classification using support vector machines and signal-based wavelet optimization," *Biomed. Signal Process. Control*, vol. 3, no. 2, pp. 169–174, Apr. 2008.
- [29] G. L. Gerstein and W. A. Clark, "Simultaneous studies of firing patterns in several neurons," *Science*, vol. 143, no. 3612, pp. 1325–1327, Mar. 1964.
- [30] D. G. Keehn, "An iterative spike separation technique," *IEEE Trans. Biomed. Eng.*, vol. BME-13, no. 1, pp. 19–28, Jan. 1966.
- [31] N. E. Huang *et al.*, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. Roy. Soc. London A, Math., Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, Mar. 1998.

- [32] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: A noise-assisted data analysis method," *Adv. Adapt. Data Anal.*, vol. 1, no. 1, pp. 1–41, Jan. 2009.
- [33] M. Lichman, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, Tech. Rep., 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [34] V. Clemens. (1999). *A Really Friendly Guide to Wavelets*. [Online]. Available: <http://agl.cs.unm.edu/~williams/cs530/arfgtw.pdf>
- [35] C. Sapsanis, G. Georgoulas, and A. Tzes, "EMG based classification of basic hand movements based on time-frequency features," in *Proc. 21st Medit. Conf. Control Autom.*, 2013, pp. 716–722.
- [36] Y. Lei, Z. He, and Y. Zi, "EEMD method and WNN for fault diagnosis of locomotive roller bearings," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7334–7341, 2011.
- [37] S. Gaci, "A new ensemble empirical mode decomposition (EEMD) denoising method for seismic signals," *Energy Procedia*, vol. 97, pp. 84–91, Nov. 2016.
- [38] G. R. Naik, S. E. Selvan, and H. T. Nguyen, "Single-channel EMG classification with ensemble-empirical-mode-decomposition-based ICA for diagnosing neuromuscular disorders," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 7, pp. 734–743, Jul. 2015.
- [39] F. D. Farfán, J. C. Politti, and C. J. Felice, "Evaluation of EMG processing techniques using information theory," *Biomed. Eng. Online*, vol. 9, no. 1, pp. 1–18, 2010.
- [40] R. N. Khushaba, S. Kodagoda, M. Takruri, and G. Dissanayake, "Toward improved control of prosthetic fingers using surface electromyogram (EMG) signals," *Expert Syst. Appl.*, vol. 39, no. 12, pp. 10731–10738, 2012.
- [41] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE Trans. Biomed. Eng.*, vol. 50, no. 7, pp. 848–854, Jun. 2003.
- [42] S. A. Fattah, O. Iqbal, S. Zahin, C. Shahnaz, and G. Rosul, "Basic hand action classification based on surface EMG using autoregressive reflection coefficient," in *Proc. IEEE Region Conf. (TENCON)*, Nov. 2017, pp. 1369–1374.
- [43] G. R. Naik *et al.*, "An ICA-EBM-based sEMG classifier for recognizing lower limb movements in individuals with and without knee pathology," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 3, pp. 675–686, Jan. 2018.
- [44] M. Herrera-González, G. A. Martínez-Hernández, J. L. Rodríguez-Sotelo, and O. F. Avilés-Sánchez, "Knee functional state classification using surface electromyographic and goniometric signals by means of artificial neural networks," *Ingeniería y Universidad*, vol. 19, no. 1, pp. 51–66, 2015.
- [45] Y. Zhang *et al.*, "Noise-assisted multivariate empirical mode decomposition for multichannel EMG signals," *Biomed. Eng. Online*, vol. 16, no. 1, pp. 1–17, 2017.



Ankit Vijayvargiya (Student Member, IEEE) received the B.Tech. (Hons.) degree in electrical engineering from Rajasthan Technical University, Kota, India, in 2010, the M.Tech. degree in instrumentation from Devi Ahilya University, Indore, India, in 2014. He is currently pursuing the Ph.D. degree in electrical engineering from the Malaviya National Institute of Technology, Jaipur, India. He is also an Associate Professor with the Department of Electrical Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur. His research interests include biomedical signal processing, pattern recognition, and machine learning.



Vishu Gupta (Member, IEEE) received the B.Sc. degree in electrical and computer engineering and the M.Sc. degree in electrical engineering from the University of Idaho, Moscow, ID, USA, in 2009 and 2012, respectively, and the Ph.D. degree in electrical engineering from the Malaviya National Institute of Technology (MNIT), Jaipur, Rajasthan, India, in 2020, while focusing on electric vehicles in realistic scenarios and charging management. She has been working as the Head of Research at TensaX Innovation Lab, since 2021. Her research interests include development of intelligent systems and artificial intelligence with specific applications to electric transportation and smart cities.



Rajesh Kumar (Senior Member, IEEE) received the B.Tech. (Hons.) degree in electrical engineering from the National Institute of Technology (NIT), Kurukshetra, India, in 1994, the M.E. (Hons.) degree in power systems from the Malaviya National Institute of Technology (MNIT), Jaipur, Rajasthan, India, in 1997, and the Ph.D. degree in intelligent systems from the University of Rajasthan, Rajasthan, India, in 2005. Since 1995, he has been a Faculty Member of the Department of Electrical Engineering, MNIT, where he is currently associated as a Professor. He was a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore (NUS), Singapore, from 2009 to 2011. His research interests include theory and practice of intelligent systems, computational intelligence and analytics, electric transportation, smart grid, and robotics.



Nilanjan Dey (Senior Member, IEEE) received the Ph.D. degree from Jadavpur University in 2015. He is an Associate Professor with the Department of Computer Science and Engineering, JIS University, Kolkata, India. He is a Visiting Fellow of the University of Reading, U.K. From 2012 to 2015, he was a Visiting Scientist at Global Biomedical Technologies Inc., CA, USA. He has authored/edited more than 90 books with Elsevier, Wiley, CRC Press, and Springer, and published more than 300 articles. His main research interests include medical imaging, machine learning, computer aided diagnosis, and data mining. He is the Editor-in-Chief of the *International Journal of Ambient Computing and Intelligence* (IGI Global) and an Associate Editor of *International Journal of Information Technology* (Springer). He is the Series Co-Editor of *Springer Tracts in Nature-Inspired Computing* (Springer) and *Advances in Ubiquitous Sensing Applications for Healthcare* (Elsevier), and the Series Editor of *Computational Intelligence in Engineering Problem Solving and Intelligent Signal Processing and Data Analysis* (CRC).



João Manuel R. S. Tavares (Member, IEEE) received the bachelor's degree in mechanical engineering and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Universidade do Porto in 1992, 1995, and 2001, respectively, and the Habilitation degree in mechanical engineering in 2015. He is a Senior Researcher with the Instituto de Ciéncia e Inovação em Engenharia Mecânica e Engenharia Industrial (INEGI) and an Associate Professor with the Department of Mechanical Engineering (DEMec), Faculdade de Engenharia da Universidade do Porto (FEUP). He is a Co-Editor of more than 60 books and a coauthor of more than 50 book chapters and 650 articles in international and national journals and conferences. He holds three international and three national patents. His main research areas include computational vision, medical imaging, computational mechanics, scientific visualization, human-computer interaction and new product development. He has been a committee member of several international and national journals and conferences. He is a Co-Founder and a Co-Editor of the book series *Lecture Notes in Computational Vision and Biomechanics* (Springer), Founder and the Editor-in-Chief of the journal *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization* (Taylor and Francis), the Editor-in-Chief of the journal *Computer Methods in Biomechanics and Biomedical Engineering* (Taylor and Francis), and a Co-Founder and a Co-Chair of the international conference series: ComplIMAGE, ECCOMAS VipIMAGE, ICCEBS, and BioDental.



Ageing effect of lower limb muscle activity for correlating healthy and osteoarthritic knees by surface electromyogram analysis



Md. Moznuzzaman ^a, Tawhidul Islam Khan ^{a,*}, Budrun Neher ^b, Kenbu Teramoto ^c, Shuya Ide ^d

^a Department of Science and Advanced Technology, Saga University, 1 Honjo-machi, Saga 840-8502, Japan

^b Dept. of Physics, Jahangirnagar University, Savar, Dhaka 1342, Bangladesh

^c Department of Advanced Health Sciences, Saga University, 1 Honjo-machi, Saga 840-8502, Japan

^d Department of Orthopedic Surgery, Faculty of Medicine, Saga University, 5-1-1 Nabeshima, Saga 849-8501, Japan

ARTICLE INFO

Keywords:

Knee osteoarthritis
Ageing
Electromyography
Lower limb muscles
Sit-stand-sit

ABSTRACT

The ubiquity of osteoarthritis (OA) in human knees has been intensified among the ageing society of the world. The intention of this work is to assess the impact of the ageing on lower limb muscles associated with OA knees through sit-stand-sit movements. Out of total eighty-four participants, 51 healthy and 33 OA patients participated in the research. Surface EMG technique was used to investigate the activities of the vastus lateralis (VL), rectus femoris (RF), vastus medialis (VM), and the medial head of the gastrocnemius (MHGM) with the increase of age. Healthy participants were classified into three groups based on their age. Jarque-Bera normality test and independent *t*-test were performed for statistical clarifications. Muscle activities of the middle age and older people were found noticeably higher and lower compare to that of the young and OA group respectively. The quadricep muscles were found to contribute more than the calf muscle.

1. Introduction

Osteoarthritis (OA) is a widely spreading musculo-skeletal disorder and about 7% of global population having more than 500 million of older population has recently been affected by this severe disease [1]. OA is responsible to initiate damage in human bone joint particularly in knee, elbow, hip and spine, though the knee joints are generally vulnerable to get affected [2]. Consequently, having affected by OA may turn into a great extent of multimorbidity as people suffer from severe pain in their joints and lose functional ability day by day, along with the elderly adults get prone to a significant risk of mortality [3]. The cost of hospitalization and medication associated with OA is increasing in a noticeable manner as well [4]. However, the occurrence and prevalence of knee OA are intensified with increasing of age [5]. Furthermore, above the age of 60 years, the severity of OA in affected people continues to increase strongly with age [6]. Therefore, knee osteoarthritis has recently been identified as a major cause of debility and physical weakness in the older society [7]. The damage to the articular cartilage and the meniscus of the knee joint has been reported along with the damage to the surrounding skeletal muscles [8]. Furthermore, the tenacious feebleness of the quadricep muscles plays a significant role in increasing the stress on the knee joints and eventually, in advancing the

joint impairment [9]. The focus of improving the functional strengths of the quadricep muscles of knee OA patients is achieving great attention in the literature [10,11]. Therefore, an efficient, reliable, safe, and user-friendly assessment of muscle strength and activity has become a crucial demand for the detection and regular monitoring of knee OA conditions.

Electromyography (EMG) is one of the most widely used techniques to obtain information on muscle damage. The EMG recordings permit to identify the key grounds of nonphysiological stance and movements, for example, abnormal readings at static condition, compensating actions of synergistic and anti-synergistic muscles, and muscle hyperactivity [12,13]. This technique works on the principle that a burst of electric potentials is produced in muscles with the association of nerve stimulation whenever a muscle contracts or extends. Therefore, EMG sensing has been frequently used to collect information on activities of lower limb muscles during sit-to-stand perturbations [14,15]. The measurement of EMG responses of muscles during knee extension would particularly reflect the applied load to the quadricep muscles, which elucidates that these are the main muscles to be focused for conducting and monitoring the rehabilitation treatment of knee OA patients.

The analysis of the structure, strength, and functional activity of the quadricep muscles has been reinforced by the researchers in the

* Corresponding author at: Department of Science and Advanced Technology, Saga University, 1 Honjo-machi, Saga 840-8502, Japan.

E-mail address: khan@me.saga-u.ac.jp (T.I. Khan).

diagnosis of the onset and progression of knee OA [16]. Quadriceps are the collective term for the vastus lateralis (VL), rectus femoris (RF), vastus medialis (VM), and vastus intermedius (VI) muscles. However, the VI muscle is located between VL and VM in the anterior compartment of the femur but deep in the RF muscle [17]. Thus, the effect of electrode location on EMG signal of the VI muscle remains obtrusive which leads to consider RF muscle for surface EMG signal acquisition instead of VI muscle. In addition, the medial head of gastrocnemius muscle (MHGM) has drawn researchers' attention in the diagnosis and therapeutic treatments of knee muscle injuries due to its ability to strain more frequently than the lateral head of gastrocnemius [18]. The positions of VL, RF, VM, and MHGM muscles in a human lower limb are shown in Fig. 1. These quadricep muscles usually prolong the legs at the knee and are vital for standing, walking, running, jumping, squatting, and all other practical activities related to the leg [19]. Furthermore, in quadricep muscles specifically, the VL plays an immense role in stabilizing the patella as well as the knee joint during movement [20]. On the other hand, the gastrocnemius muscle has two heads which collectively work to flex the plantar of the ankle and knee. Consequently, this muscle is responsible for the key activities such as running, jumping, and many other rapidly moving activities along with walking. Apparently, all of these are the main dynamic and energetic muscles for the sit-stand-sit movements.

The muscle activity of the human mobility system can be executed both in static and dynamic conditions. The length of the muscles does not change at all in static condition though the muscles remain under tension. On the other hand, the activities responsible to change in muscle lengths with the application of tension are known as dynamic muscle activities. However, muscle activity begins to work when an electrical signal impulse from the central nervous system is provided to muscle cells, and therefore chemical and mechanical activities are observed in muscle cells [49]. The typical implications of this muscle analysis mentioned in the literature are hand gesture identification [22], prosthetic control [23], sports and rehabilitation [24], and medical studies [14]. Therefore, the evaluation of the quadricep and calf muscles also demonstrates the functional status of patients with knee OA.

The sit-stand-sit perturbations are fast, easy, and generally used tasks which record time and repetitions over a certain period. Nevertheless,

the requirement of sufficient force from lower limb muscles is compulsory to complete the sit-stand-sit perturbations. Therefore, damage to the lower limb muscles should be easily experienced and identified during these perturbations. Several studies have evaluated the effects of knee joint damage on muscles activities using surface EMG technique during the sit-stand-sit movements [25–27]. Jang et al. have reported the comparative activities of gluteus medius and RF during the sit-stand-sit movements of young and elderly women [28]. The ageing and the seat height effects on lower limb muscles activities has been examined during the sit-stand-sit movements of young and older adults participants [29]. The contribution of activity intensity in repeated sit-stand-sit endurance task has been analyzed and compared between young and older participants [30,31]. The change of the effort of muscle in responding to the sit-stand-sit movements has been investigated by analyzing the torque output for young and older adults [32,33]. Repetitive sit-stand-sit trials have been analyzed of young and older adults for evaluating the change of muscle activities [34]. In spite of the ubiquity of research results focusing on a limited number of muscles of healthy young subjects and OA patients only, detail analysis of muscle activities for growing ages are not properly explained, specifically the effects of the middle age on muscle activities suffer from sufficient research evidence in the above literatures. In the present study, the activities of four different muscles in both legs of four groups of people; young aged, middle-aged, older aged people, and OA patients during their standing and sitting movements have been analyzed for clear understanding of the ageing effects of lower limb muscle activities to the initiation as well as progression of OA in knee joints. Therefore, the objectives of this research are to evaluate EMG activity of the lower limb muscles, particularly, vastus lateralis (VL), rectus femoris (RF), vastus medialis (VM), and the medial head of the gastrocnemius (MHGM) of calf muscle and to assess and correlate the ageing effects of these muscles with the occurrence of OA in knee joints.

The rest of the article is organized as follows. In Section II, we have described the experimental methods and materials required for this research, including the selection criteria of the participants and the EMG data acquisition procedure. Section III has been dedicated to acquisition and analysis of the EMG response from the experimental data and discussing the results of the work pronounced in Section II. Finally, in Section IV, we concluded the outcomes of the research and considered some works for the future.

2. Materials and method

The total experimental and statistical analysis performed in this work has been elucidated by the block diagram shown in Fig. 2. The detailed discussions of the collection and selection of participants from healthy people with different ages and OA patients, have been described under the following sections.

2.1. Equipment setup

Participants were given an instrumented sit-stand-sit performing arrangement with four EMG sensors to collect EMG data from four different lower limb muscles. The experiment included four electrodes of surface EMG sensor with built-in amplifier (REF SX230 1000, Biometrics Ltd., UK), an analog amplifier with 8-channels (K 800, Biometrics Ltd., UK), a wrist strap called grand strap, two-axis angular position measuring goniometer (K 800), AD conversion unit, and a PC with TRIAS EMG data acquisition and analysis software (Q'sfix, Japan). The 8-channel analog amplifier was also connected to boost the analog EMG signals up. The grand strap was attached to one of the wrists of the subject to limit the noises during data collection. The two-axis goniometer was pasted on the skin of the leg to measure the angular position of the legs. Finally, the EMG analog signal was converted to digital signal using an A/D converter. To reduce noise and instability in the EMG signals, three valid sit-stand-sit perturbations were recorded keeping

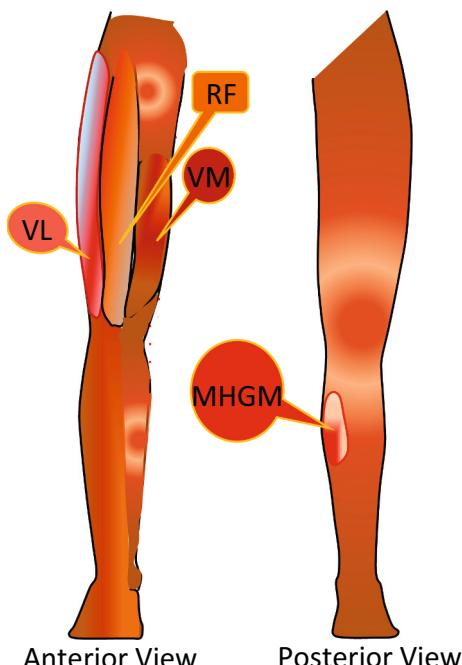


Fig. 1. Schematics of examined human lower limb muscles.

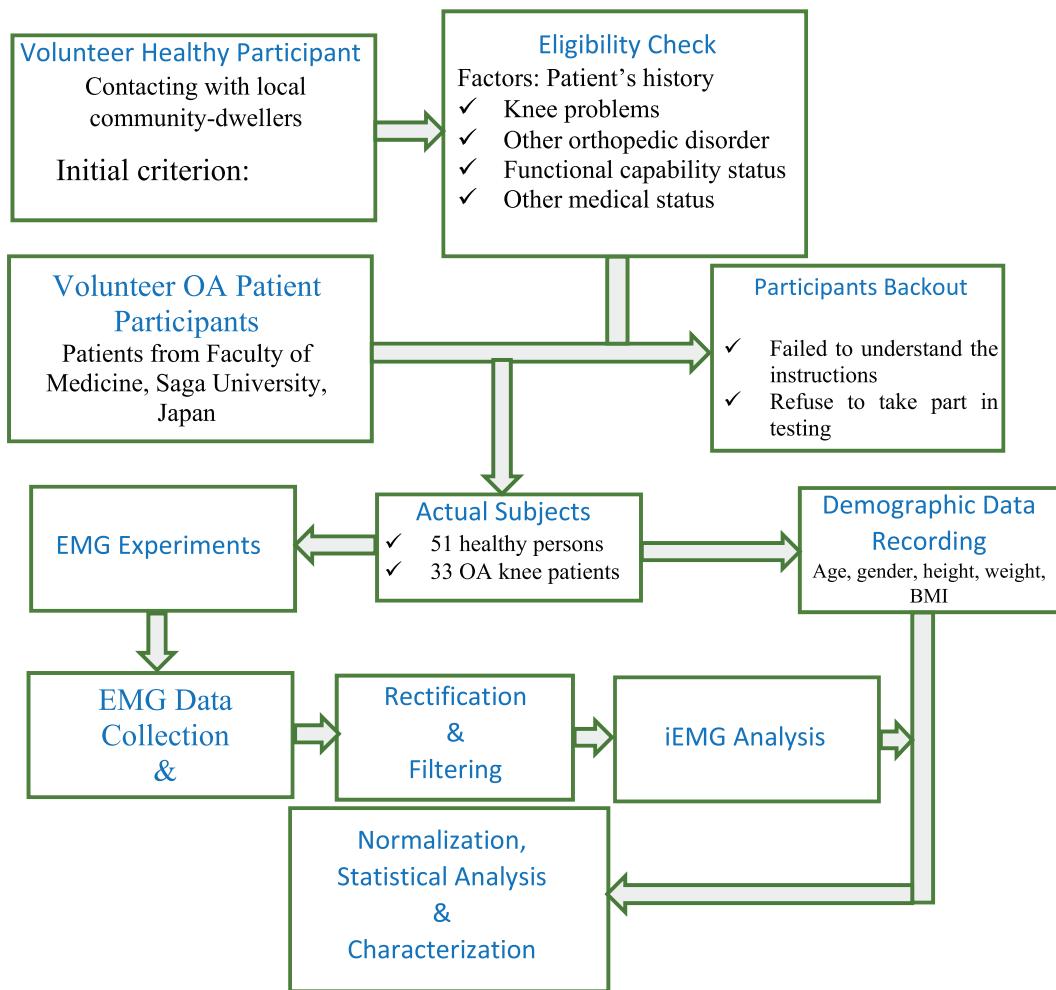


Fig. 2. Complete functional block diagram for the acquisition and analysis of EMG signals.

both hands on the chest.

2.2. About participants

In this experiment, all of the subjects (participants) took part voluntarily. Before the trial, each participant signed a written agreement form that outlines the experimental procedures and ethical concerns. Fifty-one nonpatients, participated in the experiment. The primary and mandatory criterion for selecting an eligible subject was that the age of the participants must be 20 years or above. In addition, participants did not suffer from neurological or rheumatological diseases, muscle disease or injuries of the legs. All information was recorded by interviewing the individuals. Thirty-three local knee OA patients participated in the experiment. The knee conditions of OA patients were examined by X-ray, magnetic resonance imaging (MRI), and computed tomography (CT) techniques in hospital. The demographic information was taken by a body composition analyzer (BC-622-BK, TANITA Corporation, Japan). Non-patient participants were grouped according to their ages of 20–39 years old participants as Group-A (young, male:11, female:5, total:16), 40–59 years old as Group-B (middle age, male:10, female:7, total:17), 60 years and older as Group-C (older age, male:6, female:12, total:18). All participants with knee OA were grouped into Group D (male:9, female:24, total:33). Description of the demographic measurements of the participants has been mentioned in Table 1.

The age difference between two groups of participants (nonpatient and OA patients) was found statistically significant ($p < 0.05$). Furthermore, the height of the group of patients with knee OA was found

Table 1
Demographic characteristics of the participants.

Variables	Group-A Mean (SD)	Group-B Mean (SD)	Group-C Mean (SD)	Group-D Mean (SD)	t-test
Age (years)	24(4.56)	49.94 (6.44)	66.5 (3.50)	59.73 (8.19)	AB*, AC*, AD*, BC*, BD*, CD§
Height (cm)	168.31 (8.83)	166.79 (8.09)	159.08 (9.08)	161.21 (9.07)	AB, AC*, AD*, BC*, BD*, CD
Weight (Kg)	61.97 (13.07)	69.69 (15.61)	60.73 (10.53)	67.68 (12.20)	AB, AC, AD, BC, BD, CD*
BMI (Kg/ m ²)	21.97 (5.39)	24.91 (4.63)	23.91 (3.18)	26.00 (4.02)	AB, AC, AD*, BC, BD, CD

BMI: Body Mass Index.

* Significant difference between Group-A and each of other age groups ($p < 0.05$).

† Significant difference between Group-B and each of the other age groups except Group-A ($p < 0.05$).

§ Significant difference between Group-C and Group-D ($p < 0.05$).

significantly shorter ($p < 0.05$) than that of the Group-A and the Group-B. Besides, a significant difference ($p < 0.05$) was found in weight between Group-C and Group-D. A significant difference ($p < 0.05$) in BMI was found only between the Group-A and the Group-D.

2.3. Statistical analysis

Statistical analysis for the normality check were carried out on the dependent variables to confirm the eligibility of the variables for parametric tests. Jarque-Bera tests were performed for the normality test [35]. Two-tailed *t*-tests [36] were performed to analyze the differences in muscle activities between OA patients group and the three non-patient groups (Group-A, Group-B, and Group-C). As the ratio of higher variance to lower variance of the samples were less than four (a rule of thumb), the equal variance *t*-tests were performed [37]. However, in the case of a variance ratio larger than four, two-tailed unequal variance *t*-tests were performed [37]. The significance level for the statistical test was set to 0.05. To quantify the magnitude of the effects or differences that were emerged from the experimental data, Jaco Cohen's approach was used [38]. The effect size of the above *t*-tests was designed according to the assumptions shown in Table 2. The differences of different demographic data were also analyzed using independent sample *t*-tests. The number of samples of this study was fixed. Therefore, to justify the accuracy level of the *t*-test analysis, statistical power and degree of freedom were analyzed and presented in the tables. All these statistical analyses were performed using MATLAB statistical tools (MATLAB R2021a).

3. Results and discussion

3.1. Measurements and data acquisition

Participants were prepared properly before attaching the EMG sensor electrodes to the target portion of the skin by ensuring a good electrode-skin contact. Electrodes were placed on the most protruding section of each muscle belly to obtain good quality EMG signals.

Each participant stretched at a high level to obtain maximum voluntary contraction (MVC) of his target leg and thus maximum values of EMG signals were recorded. Finally, the highest value was determined from the maximum values of the three sit-stand-sit perturbations and the recorded MVC value of each muscle. The normalized EMG data were calculated with the measured highest EMG value. All participants were to stand up from sitting position and again sit down from standing position with a normal and stress-free constant speed. A complete functional block diagram of surface EMG signal measurement, acquisition, and recording has been shown in Fig. 3. Data acquisition was performed with a bandwidth of 5–480 Hz and a common mode rejection ratio (CMRR) of 110 dB. Since the data were collected during the sit-to-stand and stand-to-sit movements, there was a risk of motion artifacts [41]. Therefore, a Butterworth band-pass filter with cut-off frequencies of 20 and 450 Hz was applied for the elimination of motion artifacts in TRIAS EMG data recording software. Filtered EMG signals were amplified by a preamplifier to obtained stronger signals, which was required for the further analysis. An A/D converter was used to convert the analog EMG signals to digital signals just before sending them to the computer. Finally, the EMG signals were full-wave rectified to measure the iEMG values of the signals for each subject before recording and exporting them to an Excel file for the further analysis.

The purpose of this study was to determine the activities of the lower limb muscles in response to sit-stand-sit movements of the three groups

of healthy participants of different age s using the surface EMG technique along with knee OA patients. Accordingly, in this section, an experimental result of the EMG activity has been presented.

The instantaneous values of EMG recordings obtained from three quadricep and one of the gastrocnemius muscles of a healthy participant during his/her sit-stand-sit movements have been shown in Fig. 4. The instantaneous values of EMG signals were visualized in millivolts. Recurrent tests in a participant showed the similar pattern of EMG signals with a very little amount of signal variation in the present experiment. Likewise, similar patterns of EMG signals were obtained for different participants with greater variability in magnitude, frequency, and phase.

3.2. Frequency analysis for lower limb muscles

Frequency analysis of EMG signals collected from each muscle was performed. The power spectral density (PSD) for each muscle was estimated using periodogram method. The sit-to-stand and stand-to-sit segmentations were windowed using Kaiser window. Mean frequencies were estimated from each of the segments of EMG signal. The results presented in Fig. 5, showed that the mean frequency characteristic of the lower limb muscles shifted downward with the growing of age with an exceptional behavior of RF muscle during only sit-to-stand movement. Since the requirement of force generation is higher during the sit-to-stand movements and RF muscle cannot be innervated much due to passive insufficiency it shows exceptional behavior at older age. Mean frequencies of the lower limb muscles of OA patients were found lower even than that of the older participants. The downward shifting rate of the mean frequency was found higher during both the sit-stand-sit perturbations for the VL muscle compare to that of the other muscles. The frequency content of the signal is generally related to the recruitment of the different types of motor units. The faster the recruited motor units, the higher the mean frequency of the recorded EMG signal. Furthermore, the rate of decrease of mean frequency was observed to be greater in middle age people than in young people. This observation supports the physiotherapeutic evidence according to which elderly muscles undergo a change toward type I muscle fibers [42]. Therefore, this is the possible reason of being changing the mean frequency of the VL muscle with higher rate with ageing as the content of type II muscle fibers in VL is greater than the other three muscles. This is an important finding which could prevent to overestimating the ageing effects with respect to the whole lower limb muscles activities.

3.3. Ageing effect in lower limb muscle activities

3.3.1. Comparison of muscle activities for left and right legs

Lower limb muscles activities of left and right legs of three groups of healthy participants were analyzed. The results of the analysis have been shown in Table 3. No statistically significant difference ($p > 0.05$) was found between the EMG activities of the muscles of the left and right leg of each age group of healthy participants. Although there was a little difference between the mean iEMG activity of the lower limb muscles of the left and right legs but it was not statistically significant. The possible reason for the difference in iEMG activity between the muscles of the left and right legs may be the partially different physical activity of the participants in one of the legs. Therefore, the right leg and left leg of a participant were not considered two different legs in the present analysis. However, for the case of OA patients only the affected legs were considered for the experiment.

3.3.2. Comparison of muscle activities between Group-A and group-B.

The calculation of iEMG values of the EMG signals of the four lower limb muscles of Group-A and Group-B was performed and statistical analysis was performed as well. The findings of the analysis have been presented in Table 4. The iEMG intensities of the participants in Group-B were observed greater than those of Group-A for both sit-to-stand and

Table 2
Descriptor table for the magnitudes of Cohen *d* (Effect size).

Descriptor	Effect Size, <i>d</i>	Ref.
Very Small	0.01	[39]
Small	0.20	[40]
Medium	0.50	[40]
Large	0.80	[40]
Very large	1.20	[39]
Huge	2.00	[39]

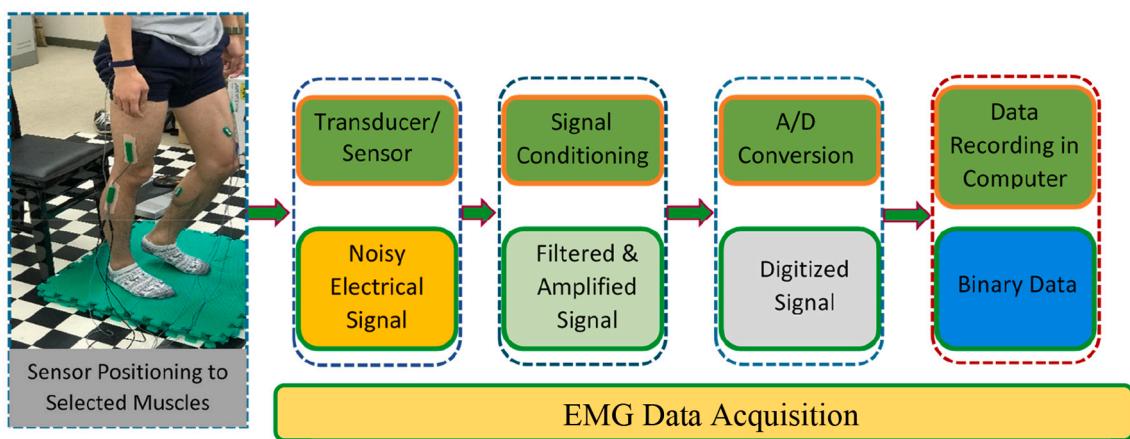


Fig. 3. Positioning of the sensors and block diagram of the acquisition of EMG data from lower limb muscles.

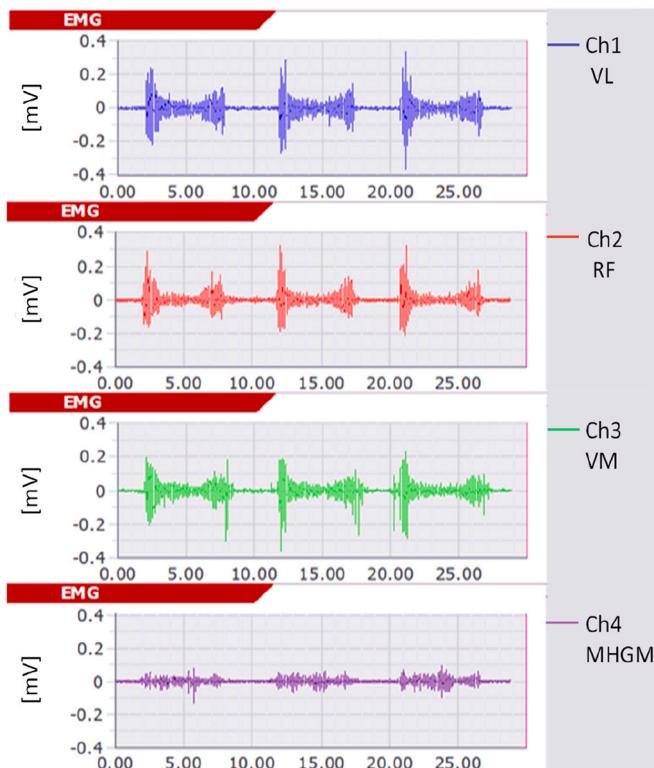


Fig. 4. The recorded EMG signals of the VL, RF, VM and MHGM muscles of a participant.

stand-to-sit perturbations. Furthermore, the differences between the muscle activities of these groups were found statistically significant ($p < 0.05$) and the effect sizes were also found huge ($d > 2.00$) and large ($d > 0.80$) in all cases of sit-stand-sit perturbations and for all selected muscles. Thus, the difference between the muscle activities of young and middle age were found noticeable. Therefore, it was experimentally identified that the weakening of the muscles of Group-B were apparent. In other words, the weakening of the muscle due to the ageing was started at the middle age. Since after a certain age muscles loss their masses, they become incapable to produce same amount of force that could produce at the younger age. Therefore, this reduction of the muscle mass leads a greater number of the motor units in the involvement in innervating a greater number of muscle fibers. Consequently, the EMG response of the muscles become higher with the increase of the

age. This increase of muscle EMG activities indicates the weakening of the muscle.

3.3.3. Comparison of muscle activities between Group-A and Group-C

To examine the ageing effect on lower limb muscles of older people, a comparison was performed between the iEMG activities of Group-A and Group-C. It was found that the lower limb muscles iEMG intensities of Group-C were significantly ($p < 0.05$) higher than those of the Group-A except only the MHGM muscle ($p = 0.59$) during the stand-to-sit perturbation. Furthermore, the experimental effect on the difference were found huge ($d > 2.00$), very large ($d > 1.20$), large ($d > 0.80$), and medium ($d > 0.50$) for the selected lower limb muscles in sit-stand-sit perturbations except the MHGM muscle (Table 5). Therefore, the magnitudes of the experimental effect on the difference of the muscle activities of young and middle age participants were identified conspicuous. Thus, the noticeable weakening of the quadricep muscles was identified at older age.

3.3.4. Comparison of muscle activities between Group-B and Group-C

The average intensities of iEMG of the lower limb muscles of the participants in Group-C increased due to the increase in age compared to those of Group-B, except for the MHGM muscle which indicated further weakening of the quadricep muscles due to ageing. However, no significant difference ($p > 0.05$) was found between the activities of the quadricep muscles of these two groups (Table 6) except the VM muscle ($p = 0.027$) only for the perturbations of stand-to-sit. However, the percentage differences between the corresponding quadricep muscles activities of Group-B and Group-C were analyzed for both of the sit-stand-sit perturbations and found that the percentage change in iEMG activity of the VM muscle due to ageing was greater than that of the other quadricep muscles (Table 7).

Since the contribution of calf muscle in sit-stand-sit movement was lower compared to quadricep muscles and the performance of MHGM muscle was found exceptional for older age participants, the percentage calculation was carried out only on the quadricep muscles. It was found that VL and VM showed higher muscle activities compared to the other two lower limb muscles during sit-stand-sit perturbations for all the groups of participants. However, only the VM muscle was found to show significantly weakened at older age. The fraction of type II muscle fibers in VL is greater than that of the VM muscles and male participants contains more type II muscle fibers than the those of the female participants [43]. Mainly, type II muscle fibers are responsible for the greater force generation capacity of the VL muscle compare to the VM muscle. Since the number of female participants in Group-C were greater than that of Group-B in the present research, the additional muscle fibers of VM of Group-C were recruited for performing the sit-stand-sit perturbations. Therefore, the change of EMG activity of the VM muscle of

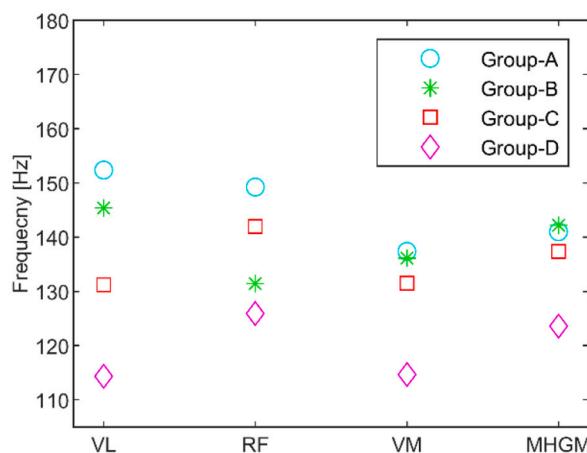


Fig. 5. Mean frequency lower limb muscles activities of young (Group-A), middle (Group-B), older (Group-C) age participants, and OA patients (Group-D) during: (a) sit-to-stand and (b) stand-to-sit perturbations.

Table 3

Independent *t*-test between iEMG activity of the lower limb muscles for the left and right legs of Group-A, Group-B, and Group-C.

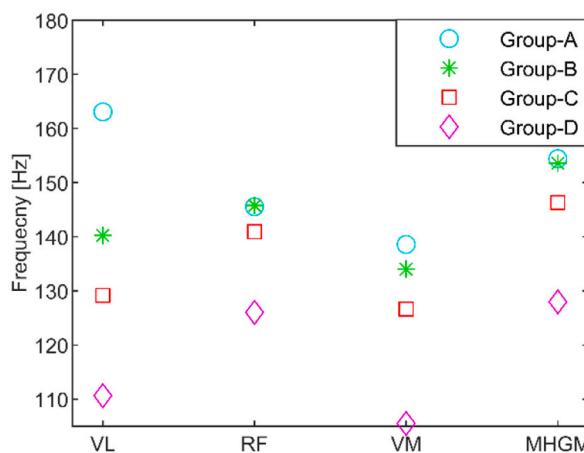
Muscle	Group-A		Group-B		Group-C							
	p-value (t)	(df)	p-value (t)	(df)	p-value (t)	(df)						
	Sit-to-Stand	Stand-to-Sit	Sit-to-Stand	Stand-to-Sit	Sit-to-Stand	Stand-to-Sit						
VL	0.29	(-1.07)	0.48	(-0.71)	0.33	(-0.97)	0.18	(-1.34)	0.11	(-1.63)	0.31	(-1.02)
	(30)	(30)	(30)	(32)	(32)	(32)	(30)	(32)	(34)	(34)	(34)	
	0.30	0.27	0.07	0.07	0.06	0.06	0.19	0.19	0.07	0.07	0.07	
RF	0.30	(-1.05)	0.27	(-1.12)	0.07	(-1.90)	0.06	(-1.97)	0.19	(-1.32)	0.19	(-1.86)
	(30)	(30)	(30)	(32)	(32)	(32)	(34)	(32)	(34)	(34)	(34)	
	0.82	0.90	0.18	0.18	0.13	0.13	0.10	0.10	0.96	0.96	0.96	
VM	0.82	(0.22)	0.90	(-0.12)	0.18	(-1.36)	0.13	(-1.52)	0.10	(-1.64)	0.10	(0.04)
	(30)	(30)	(30)	(32)	(32)	(32)	(34)	(32)	(34)	(34)	(34)	
	0.86	0.99	0.68	0.68	0.62	0.62	0.53	0.53	0.78	0.78	0.78	
MHGM	0.86	(0.16)	0.99	(0.003)	0.68	(-0.41)	0.62	(-0.49)	0.53	(-0.62)	0.53	(-0.27)
	(30)	(30)	(30)	(32)	(32)	(32)	(34)	(32)	(34)	(34)	(34)	

t: t-value of the *t*-test; df: Degree of freedom.

Table 4

Activity of lower limb muscles of young (Group-A) and middle age (Group-B) participants.

Muscle	Group-A	Group-B	t-test p-value	Effect Size	Power (df)	Perturbations
	Mean (SD)	Mean (SD)				
VL	10.94 (3.09)	19.29 (5.14)	0.0000037	1.969	0.999 (31)	Sit-to-Stand
RF	8.05 (2.86)	15.96 (4.68)	0.0000021	2.038	0.895 (31)	Sit-to-Stand
VM	9.38 (3.72)	19.49 (7.62)	0.0000388	1.685	0.254 (31)	Sit-to-Stand
MHGM	6.66 (5.23)	12.30 (5.63)	0.0056565	1.036	0.084 (31)	Sit-to-Stand
VL	8.06 (3.05)	11.53 (5.16)	0.0266973	0.816	0.736 (31)	Stand-to-Sit
RF	8.14 (3.69)	13.52 (6.83)	0.0090793	0.978	0.162 (31)	Stand-to-Sit
VM	6.78 (1.93)	8.85 (2.51)	0.0128721	0.923	0.999 (31)	Stand-to-Sit
MHGM	7.36 (5.45)	11.74 (4.69)	0.0188772	0.860	0.267 (31)	Stand-to-Sit



Group-C was found larger than that of other two quadricep muscles.

3.4. Comparing muscle activities with OA patients

3.4.1. Comparison of muscle activities between Group-A and Group-D

The graphical representations of the activities of the four lower limb muscles of the participants in Group-A and Group-D have been explained for the cases of sit-to-stand and stand-to-sit movements in Fig. 6(a) and Fig. 6(b), respectively. The bar height and error bar in the graph represent the average iEMG and standard deviation, respectively. Since participants of Group-D were collected randomly from OA patients without considering any age boundary and the stage of OA in knee, the SD (error bar) of Group-D were observed higher compare to other participants of the three age groups.

A comparison was made between the EMG activity response of the lower limb muscles of Group-A and Group-D during the sit-stand-sit movements using an independent two-tailed *t*-test. The iEMG intensities of the lower limb muscles of Group-A participants were found significantly smaller ($p < 0.05$) than those of the Group-D during the sit-to-stand and stand-to-sit perturbations. In addition, the size of the experimental effect on the difference of the muscle activities of Group-A and Group-D were found huge ($d > 2.00$). Thus, the muscle weakening of OA patients was experimentally identified noticeable compare to those of the young. The OA patients have lower muscle activity which reveals lower muscle force generation capacity compare to that of healthy young muscle and the deterioration of force generation capacity is related to OA progression and knee instability [44]. Furthermore, the weight-bearing axis of the lower extremities of patients with OA is severely affected (either medial deviation or lateral deviation) with age [45]. The angular deviation of the lower limb weight bearing axis creates an additional demand of force generation to accomplish the same amount of work (sit-stand-sit). Therefore, to meet this additional force demand of patients with OA, additional muscle fibers and innervating motor units are recruited to produce the force required to perform the same amount of works. Since EMG sensor senses as the sum of these innervating potentials of those motor units, larger EMG potentials are appeared for the affected muscles. Therefore, the four selected muscles of the young participants were identified to show significantly lower EMG values than those of OA patients and hence, it was indicated that the muscles of the young participants were healthy and unaffected.

3.4.2. Comparison of muscle activities between Group-B and Group-D

Similar analysis was carried out to inspect the effect of ageing on lower limb muscles and a graphical comparison was performed between Group-B and Group-D. The average iEMG responses of VL, RF, VM and MHGM muscles of the participants of Group-B and Group-D during their

Table 5

Activity of lower limb muscles of young (Group-A) and older age (Group-C) participants.

Muscle	Group-A		Group-C		t-test p-value	Effect Size	Power (df)	Perturbations
	Mean	(SD)	Mean	(SD)				
VL	10.94	(3.09)	21.57	(6.22)	0.00000060	2.1617	0.986 (32)	Sit-to-Stand
RF	8.05	(2.86)	17.78	(5.95)	0.00000120	2.0836	0.517 (32)	Sit-to-Stand
VM	9.38	(3.72)	21.97	(5.99)	0.00000003	2.5225	0.756 (32)	Sit-to-Stand
MHGM	6.66	(5.23)	11.09	(5.90)	0.02802100	0.7937	0.088 (32)	Sit-to-Stand
VL	8.06	(3.05)	12.02	(5.44)	0.01499500	0.8976	0.651 (32)	Stand-to-Stand
RF	8.14	(3.69)	14.40	(4.80)	0.00019100	1.4597	0.696 (32)	Stand-to-Stand
VM	6.78	(1.93)	11.91	(4.90)	0.00044400	1.3757	0.807 (32)	Stand-to-Stand
MHGM	7.36	(5.45)	8.17	(3.31)	0.59930300	0.1797	0.843 (32)	Stand-to-Stand

Table 6

Activity of lower limb muscles of middle age (Group-B) and older age (Group-C) participants.

Muscle	Group-B		Group-C		t-test p-value	Power (df)	Perturbations
	Mean	(SD)	Mean	(SD)			
VL	19.29	(5.14)	21.57	(6.22)	0.249	1.000(33)	Sit-to-Stand
RF	15.96	(4.68)	17.78	(5.95)	0.323	0.999(33)	Sit-to-Stand
VM	19.49	(7.62)	21.97	(5.99)	0.292	0.999(33)	Sit-to-Stand
MHGM	12.30	(5.63)	11.09	(5.90)	0.541	0.995(33)	Sit-to-Stand
VL	11.53	(5.16)	12.02	(5.44)	0.785	0.896(33)	Stand-to-Sit
RF	13.52	(6.83)	14.40	(4.80)	0.660	0.936(33)	Stand-to-Sit
VM	8.85	(2.51)	11.91	(4.90)	0.027	0.995(33)	Stand-to-Sit
MHGM	11.74	(4.69)	8.17	(3.31)	0.013	1.00(33)	Stand-to-Sit

Table 7

Quadricep muscle activity of older age (Group-C) participants and middle age (Group-B) participants.

Muscle	Sit-to-Stand		Stand-to-Sit	
	Mean	(SD)	Mean	(SD)
VL	11.82%		04.08%	
RF	11.40%		06.50%	
VM	12.72%		34.57%	

sit-to-stand and stand-to-sit movements have been shown in Fig. 7(a) and Fig. 7(b) respectively. The muscle iEMG activities of the lower limb of Group-B (age 40–59 years) were found significantly lower ($p < 0.05$) than the corresponding muscles of Group-D (age 44–74 years). The effect size of the analysis was also found very large ($d > 1.20$) for the selected muscles during both perturbations. Therefore, this observation signified the activities of the lower limb muscle of Group-B were noticeably better than those of OA patients (Group-D). In other words,

the muscle weakening of Group-B was not so much significant as the case of OA patients. Furthermore, the deterioration of muscle activities of the lower limb muscles of Group-B was statistically found significant ($p < 0.05$) and the size of the effect of muscle activities was also found large ($d > 0.80$) (Table 4) comparing with the Group-A. Therefore, from the above analysis, it was clarified that the significant initiation of weakening or damaging in muscles of lower limb occurred at the middle age.

3.4.3. Comparison of muscle activities between Group-C and Group-D

In order to examine the ageing effect on the muscles of the participants of older age, a statistical analysis was performed between the iEMG activities of Group-C and Group-D. The results of the analysis have been shown in Fig. 8 (a) and Fig. 8 (b) for the sit-to-stand and stand-to-sit perturbations, respectively.

In Fig. 8 it was revealed that the iEMG response of the lower limb muscles of Group-C (Age: > 60 years) was statistically found significantly lower ($p < 0.05$) from the corresponding muscle of the patients in Group-D. Furthermore, the effect size of these lower limb muscles was analyzed and found large ($d > 0.80$) and very large ($d > 1.20$) except that the VM muscle showed a significant difference ($p < 0.05$) with a medium effect size ($d = 0.667$) during the sit-to-stand perturbations. Therefore, a considerable difference was experimentally identified between the muscles of Group-C and Group-D except VM muscle. Thus, though the initiation and progression of weakening of the lower limb muscles due to ageing were identified at middle age and older aged participants, the level of muscle weakening were still not reached to that of the OA patients. Thus, muscle damage or deterioration due to age and OA in the knee were identified as significantly discriminable using the surface EMG technique. Furthermore, it was also found that the weakness in the muscle due to knee OA was greater compared to the weakness in the muscle due to ageing.

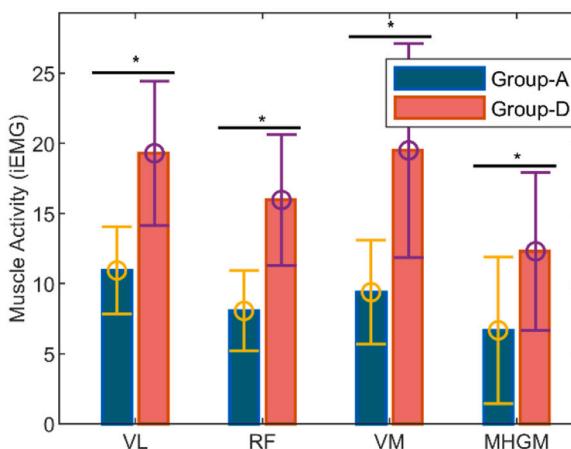
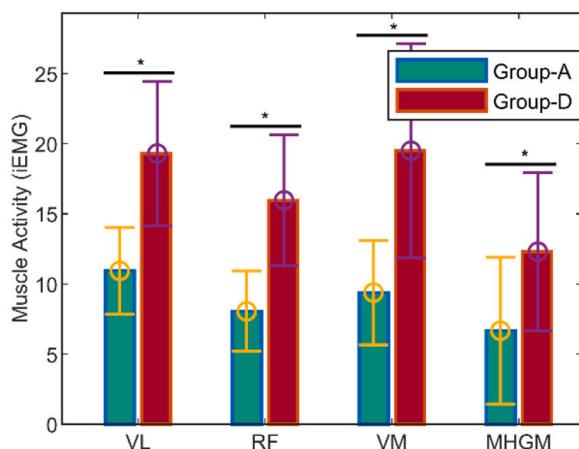


Fig. 6. iEMG activities of the lower limb muscles of young participants (Group-A) and people with knee OA (Group-D) during: (a) sit-to-stand and (b) stand-to-sit perturbations.

*Significant difference between the muscle activity of the Group-A and Group-D ($p < 0.05$).

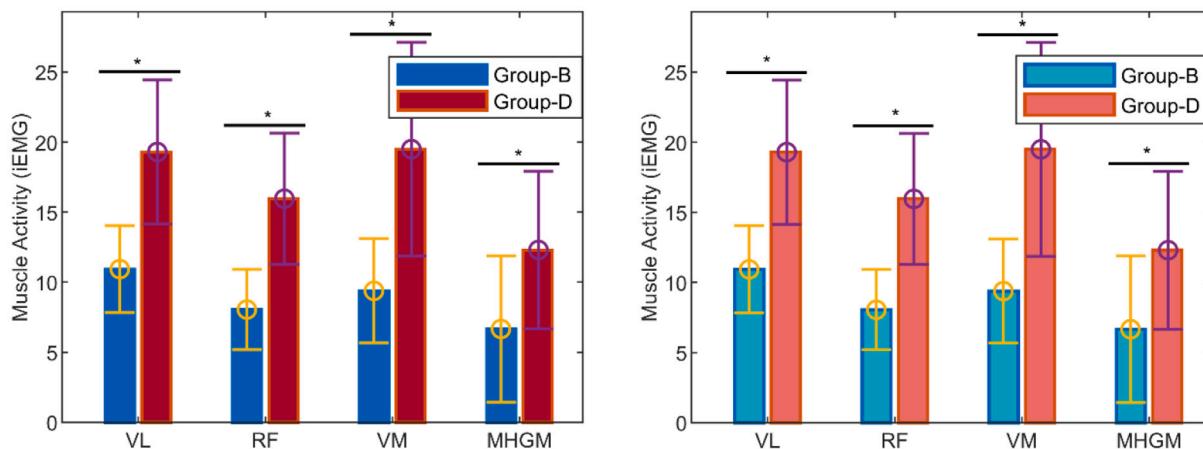


Fig. 7. iEMG activities of the lower limb muscles of middle-aged people (Group-B) and people with OA (Group-D) during (a) sit-to-stand and (b) stand-to-sit perturbations.

*Significant difference between the muscle's activities of the Group-B and Group-D ($p < 0.05$).

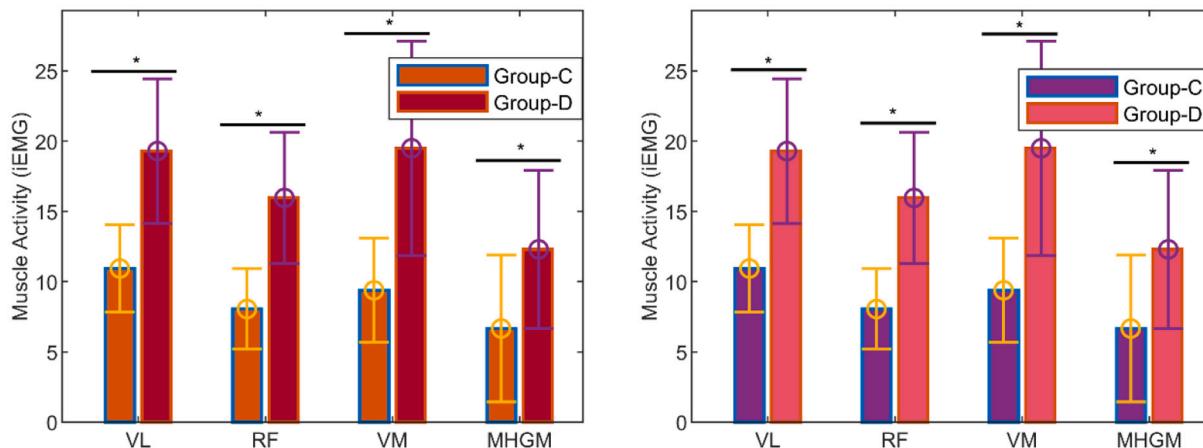


Fig. 8. iEMG activities of the lower limb muscles of older people (Group-C) and people with OA (Group-D) during (a) sit-to-stand and (b) stand-to-sit perturbations.

*Significant difference between the muscle's activities of the Group-C and Group-D ($p < 0.05$).

3.5. Discussion

In both sit-to-stand and stand-to-sit perturbations, the VL and VM muscles of the four groups of participants showed more response to iEMG activity than the other muscles. RF muscle contributes in both hip and knee extension and flexion activities and usually these activities are revealed by sit-stand-sit perturbations. However, it cannot be able to contribute more to the extension of the knee from the seated position to the standing position when it is already engaged in hip flexion in the seated position, which is known as passive insufficiency of the RF muscle [46]. Similarly, it is also not possible for RF muscle to contribute more at the same time in knee flexion and hip flexion from standing to sitting position. Furthermore, the MHGM plays an important role in the flexing of the knee joint and ankle, but cannot produce sufficient force to perform both tasks simultaneously [47]. Since during knee extension, MHGM could not contribute in ankle flexion and vice versa. Therefore, in the present experiment, major contributions for the sit-to-stand perturbations were found by the VL and VM instead of RF muscle. Similarly, MHGM muscle showed lower EMG activity in sit-stand-sit perturbations compared to VL and VM muscles.

Muscles of the lower limb showed higher activities during sit-to-stand movements compare to stand to sit movements. Additional force generation is required during sit-to-stand movements compare to stand-to-sit movements which leads to a demand to recruit additional motor

units to innervate the additional muscle fibers. As a result, higher myoelectric potentials appear at the EMG sensors. Therefore, another observation clarified in the present experiment that the lower limb muscles of the same group of participants showed higher EMG values during sit-to-stand perturbations compare to those of the stand-to-sit perturbations.

Four muscles of lower limb were examined based on their myoelectric potentials and the changes of the potential level with the changes of age were found statistically significant. For three different age ranges, the muscle activities were noticeably found to increase with ageing. The weakening of muscles reflected with the increase of muscle activities in EMG technique. In this study, muscles were found to be weakened at middle age as well as older age. This important and detail finding will help clinicians to distinguish between strong and weak muscle of different populations by quantifying the ageing effect on sit-stand-sit perturbations. In addition, the findings of this study could be beneficial in characterizing the sit-stand-sit functional movement in rehabilitation programs. However, this finding should be interpreted with caution, as there are other variables that could influence the sit-stand-sit perturbation, such as the seat height, the use of an armrest, and the BMI of the individuals.

4. Conclusion

In this study, a widely used electromyography (EMG) technique was applied to extract sufficient information and to evaluate the mean frequency and iEMG activities of the quadricep (VL, RF, and VM) and calf (MHGM) muscles on healthy and clinically diagnosed knee OA patients during their sit-stand-sit perturbations. Accordingly, three healthy groups of participants with different age levels, young, middle, and older age, along with knee OA patients participated in the experiments. The findings of this study are concluded as follows.

The quadriceps and gastrocnemius muscles of patients with knee OA were found severely weakened in perspective to sit-stand-sit perturbations because statistically significant differences were found in the activities of these muscles. According to the main objectives of this research, it was successfully clarified that the lower limb muscles were found weakened due to ageing. However, the level of weakening due to ageing was significantly lower than the level of weakening due to OA which reveals that the weakening of muscle plays influential role in the occurrence of OA to knee joint. These findings are relevant to the researchers, clinicians, and physiotherapists who are interested in working to treat functional decline of the lower limbs among older societies. Therefore, the early identification of muscle weakening will help enormously in understanding the OA disease process and in identifying the modified treatments to assure a quality life of the older society. Quadricep muscles were found to show higher muscle activities compared to calf muscle in response to sit-stand-sit perturbations. However, both the quadricep and calf muscles were observed to show higher EMG activities in response to sit-to-stand movements than that of the stand-to-sit movements. The finding was observed in both the healthy and the OA patients.

Though the pain in muscle changes the forces and EMG activities, the effect of pain in sit-stand-sit movements were not clarified in the present research. Characterization of the effects of pain in lower limb muscles due to both ageing and OA would be a potential future work.

Author's contribution

Md. Moznuzzaman and Tawhidul Islam Khan have jointly developed the concepts of this work and also contributed in manuscript developing. MM has performed the data analysis and interpretation. TIK is the lead supervisor of this project. Kenbu Teramoto and Budrun Neher contributed to the manuscript development by critiquing and reviewing the paper. Shuya Ide contributed in diagnosing the OA patients and critiquing the data during the analysis.

Funding

This work was financially supported by the KAKENHI Grant Number 21K03948 by Japan Government (MEXT).

Availability of data and materials

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Ethics approval

The research has been approved by the Ethics Committee of Saga University, Japan.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Authors gratefully acknowledge the MEXT for financial support. The authors also express their gratitude to Daishin Co., Ltd., Saga, Japan, for the kind support in conducting this research. Authors also thank the participants in this research.

References

- [1] I. for H.M. and Evaluation, Global Burden of Disease Collaborative Network. Global Burden of Disease Study 2016 (GBD 2016) Results, 2017.
- [2] P. Dieppe, J. Cushnaghan, M. Tucker, S. Browning, L. Shepstone, The Bristol "OA500 study": progression and impact of the disease after 8 years, *Osteoarthr. Cartil.* 8 (2000) 63–68, <https://doi.org/10.1053/joca.1999.0272>.
- [3] F. Landi, R. Liperoti, A. Russo, E. Capoluongo, C. Barillaro, M. Pahor, R. Bernabei, G. Onder, Disability, more than multimorbidity, was predictive of mortality among older persons aged 80 years and older, *J. Clin. Epidemiol.* 63 (2010) 752–759.
- [4] A. Kumar, A.M. Karmarkar, J.E. Graham, L. Resnik, A. Tan, A. Deutsch, K. J. Ottenbacher, Comorbidity indices versus function as potential predictors of 30-day readmission in older patients following Postacute rehabilitation, *J. Gerontol. A Biol. Sci. Med. Sci.* 72 (2017) 223–228, <https://doi.org/10.1093/gerona/glw148>.
- [5] Y. Zhang, J.M. Jordan, Epidemiology of osteoarthritis, *Clin. Geriatr. Med.* 26 (2010) 355–369.
- [6] K.D. Brandt, Diagnosis and nonsurgical management of osteoarthritis, *Professional Communications* (2010).
- [7] S. Jadi, A review of non-surgical pain management in osteoarthritis, *Cureus.* 12 (2020).
- [8] E. Shorter, A.J. Sannicandro, B. Poulet, K. Goljanek-Whysall, Skeletal muscle wasting and its relationship with osteoarthritis: a mini-review of mechanisms and current interventions, *Curr. Rheumatol. Rep.* 21 (2019) 1–8.
- [9] N.A. Segal, J.C. Torner, D. Felson, J. Niu, L. Sharma, C.E. Lewis, M. Nevitt, Effect of thigh strength on incident radiographic and symptomatic knee osteoarthritis in a longitudinal cohort, *Arthritis Care Res.* 61 (2009) 1210–1217, <https://doi.org/10.1002/art.24541>.
- [10] O.M. Olagbegi, B.O.A. Adegoke, A.C. Odole, Effectiveness of combined chain exercises on pain and function in patients with knee osteoarthritis, *Bangladesh, J. Med. Sci.* 15 (2016) 178–188.
- [11] C.A. Gbiri, U.A.C. Okafor, M.T. Alade, Comparative efficacy of open-chain and close-chain kinematics on proprioception, *Muscles' strength and functional performances in individual with knee osteoarthritis, Occup med, Health Aff.* 1 (2013) 1–5.
- [12] C.M. Bauer, F.M. Rast, M.J. Ernst, A. Meichtry, J. Kool, S.M. Rissanen, J.H. Suni, M. Kankaanpää, The effect of muscle fatigue and low back pain on lumbar movement variability and complexity, *J. Electromogr. Kinesiol.* 33 (2017) 94–102.
- [13] J. Jubany, L. Danneels, R. Angulo-Barroso, The influence of fatigue and chronic low back pain on muscle recruitment patterns following an unexpected external perturbation, *BMC Musculoskelet. Disord.* 18 (2017) 1–10.
- [14] F. Sadikoglu, C. Kavalcioglu, B. Dagman, Electromyogram (EMG) signal detection, classification of EMG signals and diagnosis of neuropathy muscle disease, *Procedia Comput. Sci.* 120 (2017) 422–429.
- [15] X. Ma, C. Ma, J. Huang, P. Zhang, J. Xu, J. He, Decoding lower limb muscle activity and kinematics from cortical neural spike trains during monkey performing stand and squat movements, *Front. Neurosci.* 11 (2017) 44.
- [16] A.G. Culveron, A. Ruhdorfer, C. Juhl, F. Eckstein, B.E. Øiestad, Knee extensor strength and risk of structural, symptomatic, and functional decline in knee osteoarthritis: a systematic review and meta-analysis, *Arthritis Care Res.* 69 (2017) 649–658.
- [17] K. Grob, M. Manestar, L. Filgueira, M.S. Kuster, H. Gilbey, T. Ackland, The interaction between the vastus medialis and vastus intermedius and its influence on the extensor apparatus of the knee joint, *knee surgery, Sport. Traumatol. Arthrosc.* 26 (2018) 727–738.
- [18] P. Counsel, W. Breidahl, Muscle injuries of the lower leg, in: *Semin. Musculoskelet. Radiol.*, © Thieme Medical Publishers, 2010, pp. 162–175.
- [19] C.T. Donaldson, J.C. Deesre, Hamstring and quadriceps injuries (2006), <https://doi.org/10.1097/01.bco.0000192894.27510.f4>.
- [20] P. Miao, Y. Xu, C. Pan, H. Liu, C. Wang, Vastus medialis obliquus and vastus lateralis activity during a double-leg semisquat with or without hip adduction in patients with patellofemoral pain syndrome, *BMC Musculoskelet. Disord.* 16 (2015) 1–8, <https://doi.org/10.1186/s12891-015-0736-6>.
- [22] J. McIntosh, A. Marzo, M. Fraser, Sensir: Detecting hand gestures with a wearable bracelet using infrared transmission and reflection, in: *Proc. 30th Annu. ACM Symp. User Interface Softw. Technol.*, 2017, pp. 593–597.
- [23] E. Fujiwara, C.K. Suzuki, Optical fiber force myography sensor for identification of hand postures, *J. Sensors.* 2018 (2018).
- [24] H.A. Feldner, D. Howell, V.E. Kelly, S.W. McCoy, K.M. Steele, "Look, your muscles are firing!": a qualitative study of clinician perspectives on the use of surface electromyography in neurorehabilitation, *Arch. Phys. Med. Rehabil.* 100 (2019) 663–675.
- [25] S. Fu, T. Duan, M. Hou, F. Yang, Y. Chai, Y. Chen, B. Liu, Y. Ma, A. Liu, X. Wang, Postural balance in individuals with knee osteoarthritis during stand-to-sit task, *Front. Hum. Neurosci.* 15 (2021).

- [26] M.T.I. Khan, T. Kurita, EMG signals for co-activations of major lower limb muscles in knee joint dynamics, *Biomed. Sci. Eng.* 3 (2015) 9–14.
- [27] M. Moznuzzaman, T.I. Khan, B. Neher, S. Ide, Effects of Lower Limb Muscle Activities to Osteoarthritic Knees, in: 2021 Jt. 10th Int. Conf. Informatics, Electron. Vis. 2021 5th Int. Conf. Imaging. Vis. Pattern Recognit., IEEE, 2021, pp. 1–6.
- [28] E.-M. Jang, W.-G. Yoo, Comparison of the gluteus medius and rectus femoris muscle activities during natural sit-to-stand and sit-to-stand with hip abduction in young and older adults, *J. Phys. Ther. Sci.* 27 (2015) 375–376.
- [29] S.T. Hurley, D.J. Rutherford, C. Hubley-Kozey, The effect of age and seat height on sit-to-stand transfer biomechanics and muscle activation, *Phys. Occup. Ther. Geriatr.* 34 (2016) 169–185.
- [30] M. Bryanton, M. Bilodeau, The role of thigh muscular efforts in limiting sit-to-stand capacity in healthy young and older adults, *Aging Clin. Exp. Res.* 29 (2017) 1211–1219.
- [31] H.S. Al Amer, M.A. Sabbahi, H.N. Alrowayeh, W.J. Bryan, S.L. Olson, Electromyographic activity of quadriceps muscle during sit-to-stand in patients with unilateral knee osteoarthritis, *BMC Res. Notes.* 11 (2018) 1–6.
- [32] M.A. Bryanton, M. Bilodeau, The influence of knee extensor fatigue on lower extremity muscle activity during chair rise in young and older adults, *Eur. J. Appl. Physiol.* 119 (2019) 61–71.
- [33] J.O. Pinzón-Arenas, R. Jiménez-Moreno, A. Rubiano, Percentage estimation of muscular activity of the forearm by means of EMG signals based on the gesture recognized using CNN, *Sens. Bio-Sensing Res.* 29 (2020), 100353.
- [34] P.C.R. dos Santos, C.J.C. Lamoth, L.T.B. Gobbi, I. Zijdewind, F.A. Barbieri, T. Hortobágyi, Older compared with younger adults performed 467 fewer sit-to-stand trials, accompanied by small changes in muscle activation and voluntary force, *Front. Aging Neurosci.* 13 (2021) 325.
- [35] C.M. Jarque, A.K. Bera, A test for normality of observations and regression residuals, *Int. Stat. Rev. Int. Stat.* (1987) 163–172.
- [36] T.K. Kim, T test as parametric statistic, *Korean J. Anesthesiol.* 68 (2015) 540.
- [37] C.A. Boneau, A comparison of the power of the U and t tests, *Psychol. Rev.* 69 (1962) 246.
- [38] J. Cohen, Things I have learned (so far)., in: *Annu. Conv. Am. Psychol. Assoc.* 98th, Aug, 1990, Boston, MA, US; Present. Aforementioned Conf., American Psychological Association, 1992.
- [39] S.S. Sawilowsky, New effect size rules of thumb, *J. Mod. Appl. Stat. Methods* 8 (2009) 26.
- [40] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, Academic press, 2013.
- [41] C.J. De Luca, L. Donald Gilmore, M. Kuznetsov, S.H. Roy, Filtering the surface EMG signal: movement artifact and baseline noise contamination, *J. Biomech.* 43 (2010) 1573–1579, <https://doi.org/10.1016/j.jbiomech.2010.01.027>.
- [42] A. Macaluso, G. De Vito, Muscle strength, power and adaptations to resistance training in older people, *Eur. J. Appl. Physiol.* 91 (2004) 450–472.
- [43] V.R. Edgerton, J.L. Smith, D.R. Simpson, Muscle fibre type populations of human leg muscles, *Histochem. J.* 7 (1975) 259–266.
- [44] R.M. Palmieri-Smith, A.C. Thomas, C. Karvonen-Gutierrez, M.F. Sowers, Isometric quadriceps strength in women with mild, moderate, and severe knee osteoarthritis, *Am. J. Phys. Med. Rehabil. Acad. Psychiatr.* 89 (2010) 541.
- [45] T. Matsumoto, M. Hashimura, K. Takayama, K. Ishida, Y. Kawakami, T. Matsuzaki, N. Nakano, T. Matsuhashita, R. Kuroda, M. Kurosaka, A radiographic analysis of alignment of the lower extremities-initiation and progression of varus-type knee osteoarthritis, *Osteoarthr. Cartil.* 23 (2015) 217–223.
- [46] T. Suehiro, M. Mizutani, M. Okamoto, H. Ishida, K. Kobara, D. Fujita, H. Osaka, H. Takahashi, S. Watanabe, Influence of hip joint position on muscle activity during prone hip extension with knee flexion, *J. Phys. Ther. Sci.* 26 (2014) 1895–1898.
- [47] M.T.I. Khan, A. Nakagaki, S. Ide, Kinematics of knee muscles influencing by aging effects as parametric index to OA, in: 2017 6th Int. Conf. Informatics, Electron. Vis. 2017 7th Int. Symp. Comput. Med. Heal. Technol., IEEE, 2017, pp. 1–4.
- [49] T. Bober, et al., The kinematics of trunk and upper extremities in one-handed and two-handed backhand stroke, *Journal of human kinetics* 30 (2011) 37–47.

Comparative Analysis of Machine Learning Techniques for the Classification of Knee Abnormality

1st Ankit Vijayvargiya

Department of Electrical Engineering
Malviya National Institute of Technology
Jaipur, India
ankitvijayvargiya29@gmail.com

3rd Nilanjan Dey

Department of Information Technology
Techno India College of Technology
Kolkata, India
neelanjan.dey@gmail.com

2nd Rajesh Kumar

Department of Electrical Engineering
Malviya National Institute of Technology
Jaipur, India
rkumar.ee@mnit.ac.in

4th João Manuel R. S. Tavares

Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial
Departamento de Engenharia Mecânica
Faculdade de Engenharia, Universidade do Porto
Porto, Portugal
tavares@fe.up.pt

Abstract—Knee abnormality is a major problem in elderly people these days. It can be diagnosed by using Magnetic Resonance Imaging (MRI) or X-Ray imaging techniques. X-Ray is only used for primary evaluation, while MRI is an efficient way to diagnose knee abnormality, but it is very expensive. In this work, Surface EMG (sEMG) signals acquired from healthy and knee abnormal individuals during three different lower limb movements: Gait, Standing and Sitting, were used for classification. Hence, first Discrete Wavelet Transform (DWT) was used for denoising the input signals; then, eleven different time-domain features were extracted by using a 256 msec windowing with 25% of overlapping. After that, the features were normalized between 0 (zero) to 1 (one) and then selected by using the backward elimination method based on the p-value test. Five different machine learning classifiers: k-nearest neighbor, support vector machine, decision tree, random forest and extra tree, were studied for the classification step. Our result shows that the Extra Tree Classifier with ten cross-validations gave the highest accuracy (91%) in detecting knee abnormality from the sEMG signals under analysis.

Index Terms—Knee Abnormality, Surface Electromyography (sEMG), Discrete Wavelet Transform (DWT), Machine Learning Classifiers.

I. INTRODUCTION

Nowadays, knee pain is a most common healthcare issue in the elderly. The major reasons behind the knee pain are related to injury, aging, repeated stress on the joint or due to an underlying condition such as arthritis. According to a study, more than one in ten adults in the USA suffer from knee osteoarthritis (Knee OA), which is a form of knee abnormality [1].

The Knee joint is one of the complex joints in the human body and provides the leg movement, the stability of the human body, and also acts as a shock absorber. Bones, ligaments, tendons, muscles, cartilage, and fluids are the different parts of the knee joint. Tibia, femur and patella are the three major bones that form the knee joint.

X-Ray, Magnetic Resonance (MR) and Computer Tomography (CT) are different imaging modalities that commonly use to detect knee abnormalities [2], [3], [4]. X-Ray is used for the initial evaluation of knee pain, but it is of low image resolution. On the other hand, MRI imaging is commonly successfully used to assess knee pain, but is very costly. As per the literature, knee abnormalities can also be diagnosed during daily life activities by means of wearable sensors like EMG, Gyrometer and Accelerometer sensors [5], [6], [7], and visual sensor like image camera [8], [9]. Respect for privacy, pervasiveness and low complexity are some of the advantages of wearable sensors over the visual sensors. In the wearable sensors family, EMG sensors are interesting for human activity recognition applications because they allow the forecast of the movement in advance and also take less time to detect the related signal variations.

Electromyography (EMG) is a technique used to analyze and record the electrical activity generated during muscle contraction by skeletal muscles. Surface EMG (sEMG) and intramuscular EMG (iEMG) are the two ways used to collect EMG signals [10]. sEMG has several advantages over iEMG, therefore, sEMG signals have had a critical role in analyzing lower limb movements, particularly to detect anomalies related to the limbs. Artificial Neural Network based knee abnormalities classification was proposed by Erkamaz et al. [4]. Vijayvargiya et al. analyzed the early detection of knee osteoarthritis by using a support vector machine classifier with different kernels [11]. Ertugrul et al. used the classification of surface EMG signals of the lower and upper limbs based on adaptive local binary patterns [12].

In this work, we present a comparative analysis of the performance of various machine learning classifiers for knee

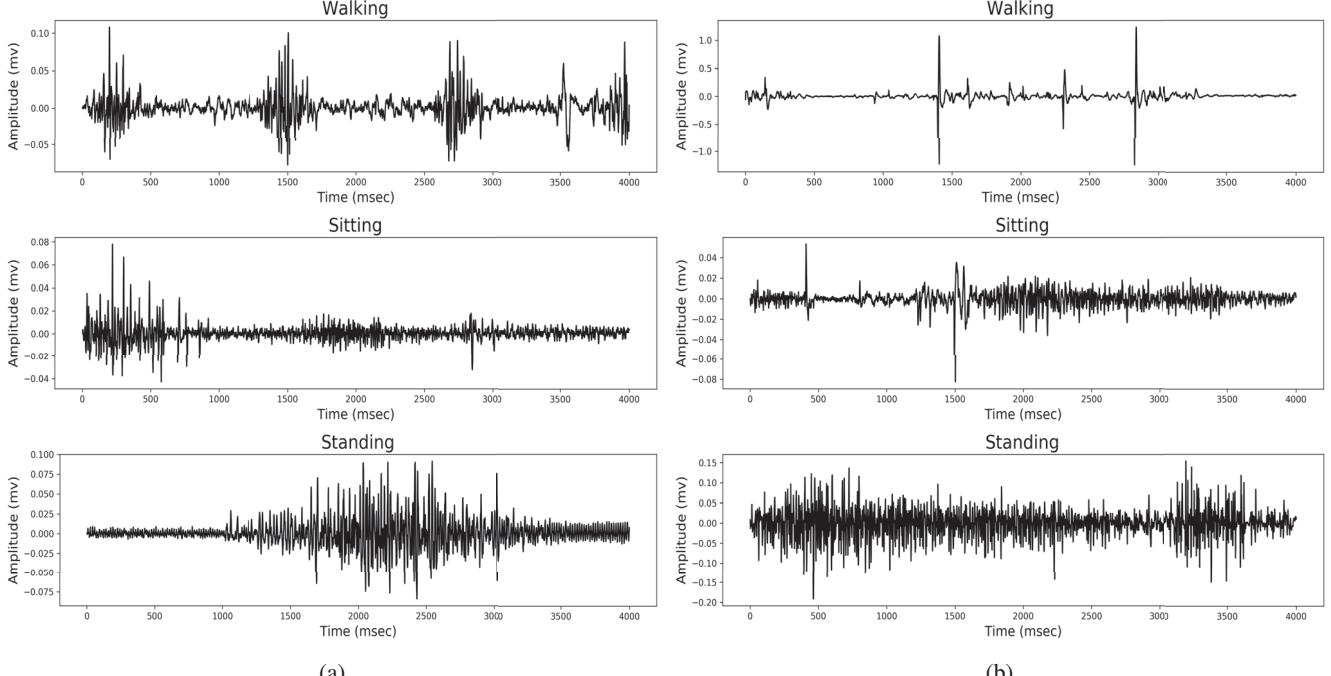


Fig. 1: sEMG signals acquired during three different movements: Walking, Standing and Sitting, from a (a) Normal subject and a (b) Abnormal subject.

abnormality detection from sEMG signals. This article is divided into five sections. A brief overview of the used EMG dataset is provided in Section II. The detail of the proposed methodology is given in Section III. Results and discussion are presented in Section IV. Conclusions and possible future research are discussed in Section V.

II. DATASET

In this article, we used a dataset with the surface EMG signals acquired during three different movements: standing, walking and sitting, performed by twenty-two subjects [13]. All of the subjects were more than 18 years old, and eleven of the individuals were healthy and the remaining were suffering from knee abnormalities. The healthy individuals did not have any record of a knee injury while the unhealthy individuals had suffered any knee abnormality already diagnosed by professionals. A DataLog MWX8 and a goniometer were used to collect the data. The surface EMG data were collected around four distinct muscles: rectus femoris (RF), biceps femoris (BF), vastus medialis (VM) and semitendinosus (ST). The goniometer was attached to the external side of the knee joint. All the acquired data were stored on the computer. Fig 1 shows examples of the normal and abnormal subject's sEMG signals acquired during each movement, respectively.

III. PROPOSED METHODOLOGY

This section presents the methodology proposed for knee abnormality detection from surface EMG signals. Fig. 2 illustrates the basic steps involved in the proposed classification

of the sEMG signals. First, the discrete wavelet transform (DWT) is used to denoise the raw sEMG signals and then different features are extracted using overlapping windowing techniques. To reduce the feature space dimensionality and improve the classification performance, a backward elimination method for feature selection is used. After that, the selected features are fed to the machine learning classifiers and their performance analyzed.

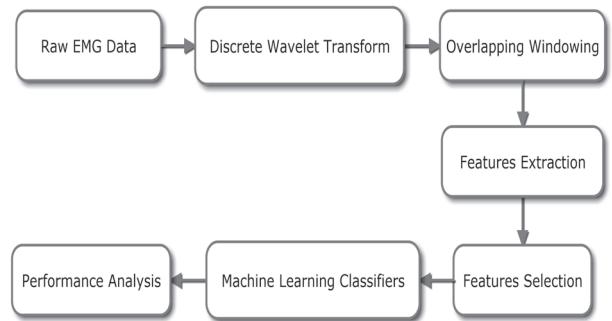


Fig. 2: Flowchart of the proposed methodology for knee abnormality detection from sEMG signals.

A. Discrete Wavelet Transform

During the acquisition of the sEMG signal, several different kinds of noise are introduced in the data. Particularly, the noises are due to external disturbances and psychological disturbances. Owing to the combination of various noise signals or artifacts, the specificities of an EMG signal can be lost. The

attributes of the acquired EMG signal depend on the skin temperature, internal structure of the subject, skin formation, blood flow rate, tissue structure, measurement location, etc.

It is not possible to use traditional filtering techniques such as high-pass, low-pass and band-pass to eliminate unwanted noise like impulse within the EMG signal spectrum band. Several noise removal techniques have been proposed for sEMG signals including Independent Component Analysis (ICA), Discrete Wavelet Transform (DWT) and Empirical Mode Decomposition (EMD) [14], [15], [16], [17], [18]. In this work, DWT is applied to denoise the raw sEMG signals because it has minimum signal distortion and gives information in both frequency and time domains.

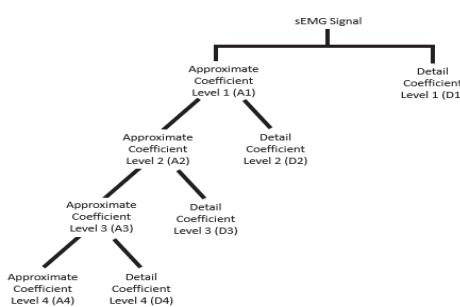


Fig. 3: A wavelet decomposition up to level 4.

A sEMG signal can be decomposed into various levels in a discrete wavelet transformation using various wavelets such as Haar, Daubechies, Marlet and Symlet. The transformation can be implemented as a bank of filters which contain low pass filters (approximate coefficients) and high pass filters (detail coefficients). Further, the signal is passed through the next level of low and high pass filters. The number of coefficients depends on the level of decomposition. Fig. 3 shows a wavelet decomposition up to level 4. A wavelet is generated from a mother wavelet (ψ_t), by scaling (s) and translation (τ) [19]:

$$\psi_{s,\tau} = \frac{1}{\sqrt{s}} \psi \left(\frac{t - \tau}{s} \right). \quad (1)$$

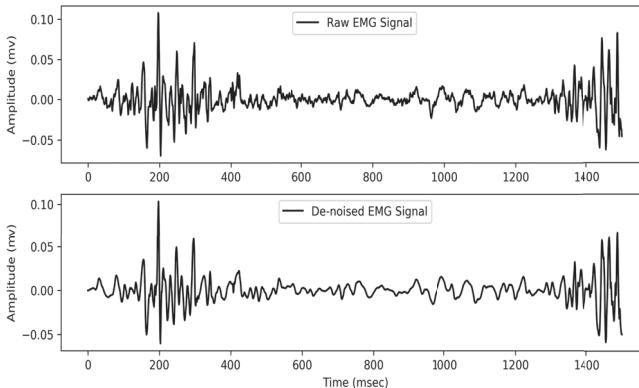


Fig. 4: A raw sEMG signal and correspondnt denoised signal by using DWT denoising.

To remove the noise after the signal decomposition, several thresholding techniques such as soft thresholding and hard thresholding have been used. In this study, wavelet denoising was applied to the sEMG signals using the Daubechies 7 wavelet family (db7) till the fourth level of decomposition. Garotte thresholding was applied to detail coefficients D2, D3 and D4, as shown in Fig. 3. Fig. 4 shows the a raw EMG signal and the corespondent denoised signal obtained by wavelet denoising.

B. Segmentation

EMG signals are random in nature. Due to this randomness, the segmentation of a EMG signal is necessary. Windowing technique has been used for the segmentation of EMG signals. There are two different techniques (Fig 5) for EMG data segmentation: overlapping windowing and adjacent windowing [20], [21]. In this work, we used overlapping windowing with 256 msec window length and 25% of overlapping [7].

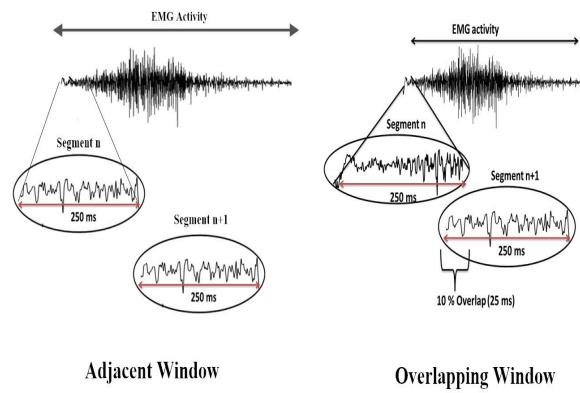


Fig. 5: Windowing techniques used in EMG sinal segmentatn.

C. Feature Extraction and Selection

Different types of artifacts and noises are still present after the usual pre-processing of the EMG data. Those noises degrade the performance of the further classification model. Different kinds of features in time domain, frequency domain and time-frequency domain can be extracted from the pre-processed EMG data to improve the performance of the classification model. In this study, eleven time-domain features, which are indicated in Table I, were extracted from each of the four different muscle EMG signals (i.e. 44 features in total) and used in the classification of knee abnormality.

After the features extraction, the selection of the relevant features or removal of the redundant features is a very challenging task. We applied the backward elimination technique for the diagnosis of knee abnormality. In this approach, we begin with all of the model's attributes, followed by their removal based on the p-value test. Those attributes are discarded with p-values greater than 0.05 and the model is refitted

TABLE I: Extracted sEMG features and their mathematical formulation.

	Extracted Feature	Mathematical formulation
1	Mean Absolute Value (MAV)	$\frac{1}{N} \sum_{i=1}^N x_i $ where x_i is a sample of the EMG signal
2	Root Mean Square (RMS)	$\sqrt{\frac{1}{N} \sum_{i=1}^N x_i ^2}$
3	Zero Crossing (ZC)	where $f(x_i) = \begin{cases} 1 & \text{if, } (x_i > 0 \text{ and } x_{i+1} < 0) \\ & \text{or } (x_i < 0 \text{ and } x_{i+1} > 0) \\ 0 & \text{otherwise} \end{cases}$
4	Slope Sign Change (SSC)	where $f(x_i) = \begin{cases} 1 & \text{if, if, } (x_i > x_{i-1} \text{ and } x_i > x_{i+1}) \\ & \text{or } (x_i < x_{i-1} \text{ and } x_i < x_{i+1}) \\ 0 & \text{otherwise} \end{cases}$
5	Variance (VAR)	$\frac{1}{N-1} \sum_{i=1}^N x_i^2$
6	Willison Amplitude (WAMP)	where $f(x_i) = \begin{cases} 1 & \text{if, } (x \geq \text{Threshold}) \\ 0 & \text{otherwise} \end{cases}$
7	Myopulse Percentage Rate (MYOP)	where $f(x_i) = \begin{cases} 1 & \text{if, } (x \geq \text{Threshold}) \\ 0 & \text{otherwise} \end{cases}$
8	Difference Absolute Standard Deviation Value (DASDV)	$\sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} (x_{i+1} - x_i)^2}$
9	Average Amplitude Change (AAC)	$\frac{1}{N} \sum_{i=1}^{N-1} x_{i+1} - x_i $
10	Skewness (Skew)	$\frac{E[(x-\mu)^3]}{\sigma^3}$
11	Kurtosis (Kurt)	where σ is the Standard deviation of the signal dataset, μ = Mean of the dataset and E is the Expected value estimator of the dataset.

with the remaining attributes. This process is iterated several times until each existing variable has a significant level for the model. After the selection of the feature, all selected features are normalized between 0(zero) and 1 (one) according to:

$$X_{Fnew} = \frac{X_F - X_{Fmin}}{X_{Fmax} - X_{Fmin}}, \quad (2)$$

where X_{Fnew} is the normalized EMG feature, X_F is the actual EMG feature, X_{Fmax} is the maximum value of the actual EMG feature, and X_{Fmin} is the minimum value of the actual EMG feature.

D. Machine Learning Classifiers

In this section, support vector machine, decision tree, k-nearest neighbor, random forest and extra tree machine learning classifiers, which were used in this work in the classification between knee healthy and unhealthy individuals from sEMG signals, are introduced.

Support Vector Machine (SVM) [22] is a supervised machine learning classifier that can be used for both linear and non-linear classification by using different kernels. In a linear SVM classifier is built a hyperplane with maximum margin width.

K-Nearest Neighbour (KNN) [23] is used for both regression and classification problems. A KNN classifier is trained with all the available cases and then categorized new cases by neighboring majority votes. The case is assigned to that

class, most similar to its nearest k neighbors, determined by a distance function which can be the Euclidean, Manhattan or Minkowski distance.

Decision Tree (DT) [24] is a supervised learning classifier that can be used for both numerical or categorical data. It uses a kind of tree structure of decision based on entropy:

- 1) Calculate the entropy of each feature:

$$H(X) = - \sum_{t \in Y} p(t) \log_2 p(t) \quad (3)$$

where X is the dataset, Y is the set of classes in s , and $p(t)$ is number of elements in class Y to the number of elements in X .

- 2) Split the dataset into subsets using the attribute with maximum information gain:

$$IG(X, A) = H(X) - H(X|A) \quad (4)$$

- 3) No further splitting is required if the value of entropy is 0 (zero); otherwise, further splitting is performed as above.

Random Forest (RF) [25] is a similar kind of decision tree algorithm, but it creates several trees rather than a single tree:

- 1) Select randomly the number of samples in the dataset randomly which is called bootstraps sample.
- 2) Create the decision tree for every bootstrap sample and calculate the prediction results from each decision tree.

TABLE II: Classification performance achieved by each classifier in terms of different performance metrics.

Classifier	Accuracy		Sensitivity		Specificity		F1-Score	
	μ	σ	μ	σ	μ	σ	μ	σ
SVM	0.701	0.019	0.404	0.028	0.864	0.024	0.550	0.027
DT	0.700	0.018	0.567	0.043	0.773	0.022	0.653	0.028
KNN	0.793	0.012	0.641	0.028	0.878	0.017	0.740	0.018
RF	0.888	0.017	0.789	0.037	0.943	0.015	0.859	0.024
ET	0.913	0.013	0.825	0.034	0.962	0.011	0.888	0.02

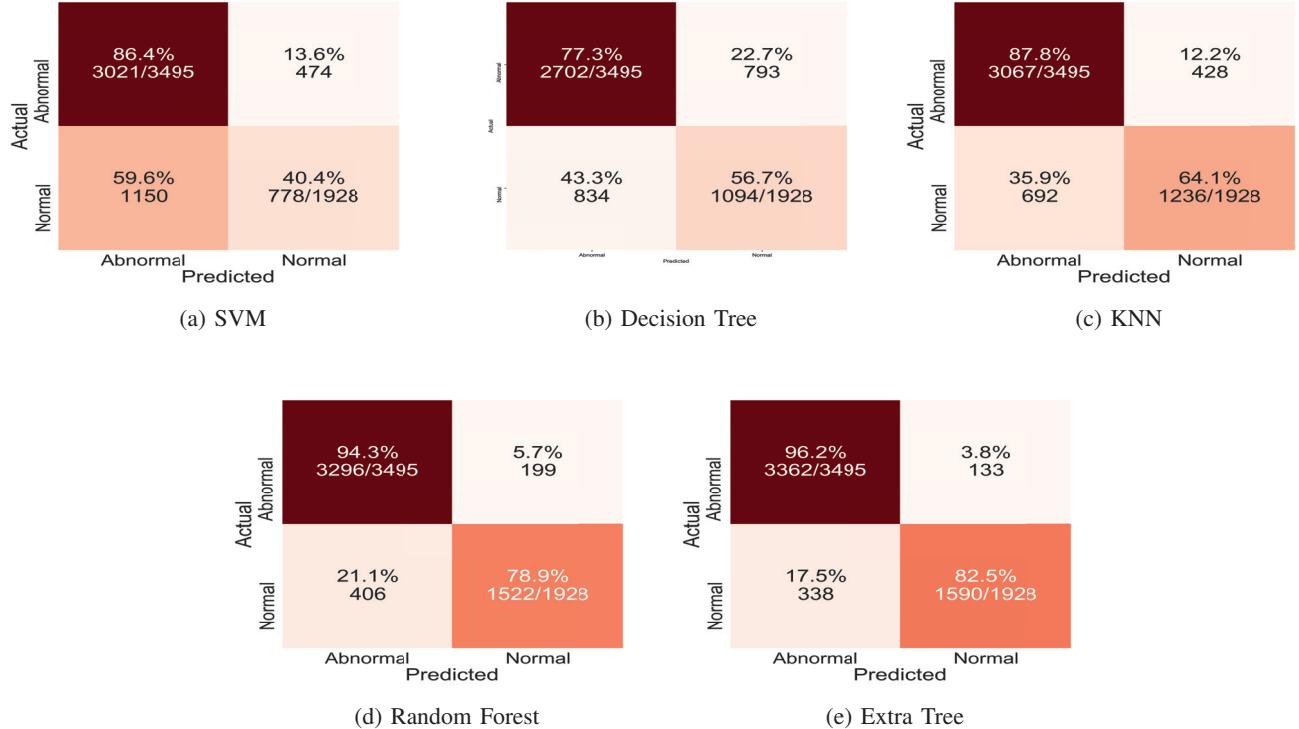


Fig. 6: Confusion matrices built from the results obtained by each classifier under study.

- 3) Voting is performed between the results of decision trees, and the most voted prediction result shows the output of the overall prediction.

Extra Tree (ET) [26] is very similar to the random forest classifier. It considers the entire input dataset instead of the bootstrap dataset.

IV. RESULTS AND DISCUSSION

This section presents the performance analysis of the different machine learning classifiers studied. In this application, the problem is a binary classification problem: either subject is knee healthy or unhealthy. A machine learning algorithm for binary classification gives four possible outcomes, which are: True Negative (TN), True Positive (TP), False Negative (FN) and False Positive (FP). The classification performance metrics were computed for each classifier under study from its number of occurrences of each of these four possible results:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (5)$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (6)$$

$$\text{Sensitivity}(Recall) = \frac{TP}{TP + FN}, \quad (7)$$

$$\text{Specificity} = \frac{TN}{TN + FP}, \quad (8)$$

$$F1 - Score = \frac{2 * Recall * Precision}{Recall + Precision}. \quad (9)$$

K-fold cross-validation is a re-sampling method that uses a constrained data test to evaluate the classification performance of machine learning models. In this k-fold approach, the input data are randomly divided into k groups of equal size. Then the model is trained using $k - 1$ groups of data and validated with the k^{th} group of data. This process is repeated for all the groups of obtained from the input data.

In this work, the classification performance metrics were evaluated for each model under study by using 10 fold cross-validation. The obtained classification performance results for each of the used metrics and studied classifiers are indicated in Table II. The accuracy for the extra tree classifier was of 91.3% while it was equal to 70.1%, 70.0%, 79.3% and 88.8% for the support vector machine, decision tree, k-nearest neighbor and random forest classifiers, respectively. Extra Tree classifier shows a superior classification performance to the other classifiers. Fig. 6 shows the confusion matrices built for each of the studied classifiers in the knee abnormality classification from sEMG signals.

V. CONCLUSION AND FUTURE WORK

A comparative analysis of different machine learning classifiers in the classification of knee abnormality from sEMG signals acquired from the lower limb of subjects suffering from knee abnormality and healthy subjects was presented. First, raw EMG signal was denoised using a discrete wavelet transform then eleven signal features were extracted from the pre-processed signal by using an overlapping windowing technique. Then, the relevant signal features were selected by using the backward selection method. Afterwards, five different machine learning models were used and their classification performance was evaluated, where the Extra Tree classifier shown its superiority.

In this study, a public available EMG dataset was used. The dataset only includes data from 22 subjects, therefore, as a future work, we will collect EMG data from more subjects and study more advanced machine learning classifiers as Convolution Neuronal Networks (CNNs), which have obtained very promising results in several classification problems.

REFERENCES

- [1] Daniel K White, Catrine Tudor-Locke, David T Felson, K Douglas Gross, Jingbo Niu, Michael Nevitt, Cora E Lewis, James Torner, and Tuhina Neogi. Do radiographic disease and pain account for why people with or at high risk of knee osteoarthritis do not meet physical activity guidelines? *Arthritis & Rheumatism*, 65(1):139–147, 2013.
- [2] J Bedson, K Jordan, and P Croft. How do gps use x rays to manage chronic knee pain in the elderly? a case study. *Annals of the rheumatic diseases*, 62(5):450–454, 2003.
- [3] Charles Peterfy and Manish Kothari. Imaging osteoarthritis: magnetic resonance imaging versus x-ray. *Current rheumatology reports*, 8(1):16, 2006.
- [4] Okan Erkaymaz, Şenyer İrem, and Rukiye Uzun. Detection of knee abnormality from surface emg signals by artificial neural networks. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE, 2017.
- [5] Tahir Hussain, Hafiz Farhan Maqbool, Nadeem Iqbal, Mukhtaj Khan, Salman, and Abbas A Dehghani-Sanij. Computational model for the recognition of lower limb movement using wearable gyroscope sensor. *International Journal of Sensor Networks*, 30(1):35–45, 2019.
- [6] Orkun Tunçel, Kerem Altun, and Billur Barshan. Classifying human leg motions with uniaxial piezoelectric gyroscopes. *Sensors*, 9(11):8508–8546, 2009.
- [7] Ganesh R Naik, S Easter Selvan, Sridhar P Arjunan, Amit Acharyya, Dinesh K Kumar, Arvind Ramanujam, and Hung T Nguyen. An ica-ebm-based semg classifier for recognizing lower limb movements in individuals with and without knee pathology. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(3):675–686, 2018.
- [8] Mario Nieto-Hidalgo, Francisco Javier Ferrández-Pastor, Rafael J Valdivieso-Sarabia, Jerónimo Mora-Pascual, and Juan Manuel García-Chamizo. A vision based proposal for classification of normal and abnormal gait using rgb camera. *Journal of biomedical informatics*, 63:82–89, 2016.
- [9] He Zhang and Cang Ye. Rgb-d camera based walking pattern recognition by support vector machines for a smart rollator. *International journal of intelligent robotics and applications*, 1(1):32–42, 2017.
- [10] Dario Farina and Francesco Negro. Accessing the neural drive to muscle and translation to neurorehabilitation technologies. *IEEE Reviews in biomedical engineering*, 5:3–14, 2012.
- [11] Ankit Vijayvargiya, Puru Lokendra Singh, Samidha Mridul Verma, Rajesh Kumar, and Sanjiv Bansal. Performance comparison analysis of different classifier for early detection of knee osteoarthritis. In *Sensors for Health Monitoring*, pages 243–257. Elsevier, 2019.
- [12] Ömer Faruk Ertuğrul, Yılmaz Kaya, and Ramazan Tekin. A novel approach for semg signal classification with adaptive local binary patterns. *Medical & biological engineering & computing*, 54(7):1137–1146, 2016.
- [13] OFA Sanchez, JLR Sotelo, MH Gonzales, and GAM Hernandez. Emg dataset in lower limb data set. *UCI machine learning repository*, pages 2014–02, 2014.
- [14] Ching-Fen Jiang and Shou-Long Kuo. A comparative study of wavelet denoising of surface electromyographic signals. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1868–1871. IEEE, 2007.
- [15] Adriano O Andrade, Slawomir Nasuto, Peter Kyberd, Catherine M Sweeney-Reed, and FR Van Kanijn. Emg signal filtering based on empirical mode decomposition. *Biomedical Signal Processing and Control*, 1(1):44–55, 2006.
- [16] Alexandros Karagiannis and Philip Constantinou. Noise-assisted data processing with empirical mode decomposition in biomedical signals. *IEEE Transactions on Information Technology in Biomedicine*, 15(1):11–18, 2010.
- [17] M Kania, M Ferencic, and R Maniewski. Wavelet denoising for multi-lead high resolution ecg signals. *Measurement science review*, 7(4):30–33, 2007.
- [18] Zhi-dong Zhao, Min Pan, Guang Li, and Yu-quan CHEN. Independent component analysis (ica) for denoising emg signal from ecg signal. *JOURNAL-ZHEJIANG UNIVERSITY ENGINEERING SCIENCE*, 38(1):103–107, 2004.
- [19] Clemens Valens. A really friendly guide to wavelets. *ed. Clemens Valens*, 1999.
- [20] Nurhazimah Nazmi, Abdul Rahman, Mohd Azizi, Shin-Ichiroh Yamamoto, Siti Anom Ahmad, Hairi Zamzuri, and Saiful Amri Mazlan. A review of classification techniques of emg signals during isotonic and isometric contractions. *Sensors*, 16(8):1304, 2016.
- [21] Mohammadreza Asghari Oskoei and Huosheng Hu. Support vector machine-based classification scheme for myoelectric control applied to upper limb. *IEEE transactions on biomedical engineering*, 55(8):1956–1965, 2008.
- [22] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [23] Sahibsingh A Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (4):325–327, 1976.
- [24] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.
- [25] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [26] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.



Deep ensemble learning approach for lower limb movement recognition from multichannel sEMG signals

Pratibha Tokas¹ · Vijay Bhaskar Semwal¹ · Sweta Jain¹

Received: 27 May 2023 / Accepted: 14 January 2024 / Published online: 17 February 2024
© The Author(s), under exclusive licence to Springer-Verlag London Ltd, part of Springer Nature 2024

Abstract

Walking is a complex task that requires consistent practice to master, and it involves the synchronisation between the lower limbs and the brain, making it challenging. While bipedal robots have been developed to mimic human walking, they must achieve an efficient gait due to structural differences and walking challenges. This study aims to produce a more human-like walk by analysing human lower extremity activities. To capture the bipedal robot locomotion learning process, an ensemble classifier based on deep learning is introduced to recognise human lower activities. A publicly available UC Irvine Machine Learning Repository (UCI) dataset on surface electromyography (sEMG) signal for the lower extremity of 11 fit participants and 11 participants with knee disorders for sitting while performing knee extension, walking, and standing while performing knee flexion is used. A hybrid ensemble of deep learning models comprising long short-term memory and convolution neural network is employed to classify activities, with reported average accuracies of 98.8%, 98.3%, and 99.3% for healthy subjects for sitting, standing and walking, respectively. Moreover, the ensemble model reported average accuracies of 98.2%, 98.1%, and 99.0% for individuals with knee pathology. Notably, this study holds promising significance, as it has yielded a considerable enhancement in performance as opposed to state-of-the-art work. The applications of this work are diverse and include improving postural stability in elderly subjects, aiding in the rehabilitation of patients recovering from stroke and trauma, generating walking trajectories for robots in complex environments, and reconstructing walking patterns in individuals with impairments.

Keywords Human activity recognition (HAR) · sEMG · Deep learning · Hybrid ensemble classifier · Signal processing · Bipedal robots

1 Introduction

Gait analysis is crucial in rehabilitation because it gives reliable information regarding a patient's walking pattern and lower limb motions. This information can be used to identify divergences from normal gait and assess rehabilitation interventions' effectiveness. By examining gait, clinicians can demarcate the degree of impairment, monitor

progress during rehabilitation, and adjust the treatment plan accordingly. Gait analysis plays a crucial role in rehabilitation by providing valuable insight into the patient's operational abilities and enabling clinicians to optimise the treatment plan to attain the most suitable outcomes. Injuries like knee osteoarthritis, sciatica, meniscus and anterior cruciate ligament (ACL) are among the leading sources of impairment worldwide, affecting people of all ages [1, 2]. It has been demonstrated that assistive technology has the potential to enhance the standard of life for individuals with these injuries, particularly by tracking their rehabilitation progress [3, 4]. Gait analysis is a standard diagnostic tool for neuromuscular and skeletal disorders that helps classify and assess lower extremity motion [5]. However, conventional clinical rehabilitation techniques involving gait analysis require comprehensive laboratory settings, which can be time-consuming, costly, and inconvenient,

✉ Pratibha Tokas
213112001@stu.manit.ac.in
Vijay Bhaskar Semwal
vsemwal@manit.ac.in
Sweta Jain
Swetajain@manit.ac.in

¹ Maulana Azad National Institute of Technology, Bhopal, M.P., India

especially in secluded areas [6]. Remote monitoring of rehabilitation improvement using wearable devices has become critical to overcoming these limitations. These wearables can not only control assistive devices like exoskeletons but also yield improvement feedback to users and assist clinicians in assessing and treating patients [7–9]. Lower extremity motions are integral to many human activities, including sitting, standing, stair ascent and descent, and squatting. Gait analysis, which involves classifying and evaluating lower leg motions [3, 10], is crucial for diagnosing neuromuscular and skeletal disorders. However, traditional gait analysis techniques require extensive laboratory setups, making it necessary to develop more straightforward methods for assessing gait dysfunction. Various noninvasive and kinematics techniques have been proposed for this purpose [11].

In recent years, electromyography (EMG) has emerged as a widely used approach for recording muscle activities in the skeletal muscles, which is valuable for investigating neuropathic and myopathic conditions, controlling prosthetic devices, and aiding in rehabilitation. EMG signals can be obtained using surface or concentric needle electrodes. sEMG is mainly used in rehabilitation and prosthetic applications. In contrast, concentrically arranged needles diagnose neuromuscular disorders affecting motor units (MUs). In [12], Gautam et al. have used a transfer learning-based LRCN model on the publicly available UCI dataset to predict joint angles and classify lower extremity activities. They have achieved a mean classification accuracy of 92.4% and 98.1% for participants with knee disorders and fit participants, respectively. A general deep-

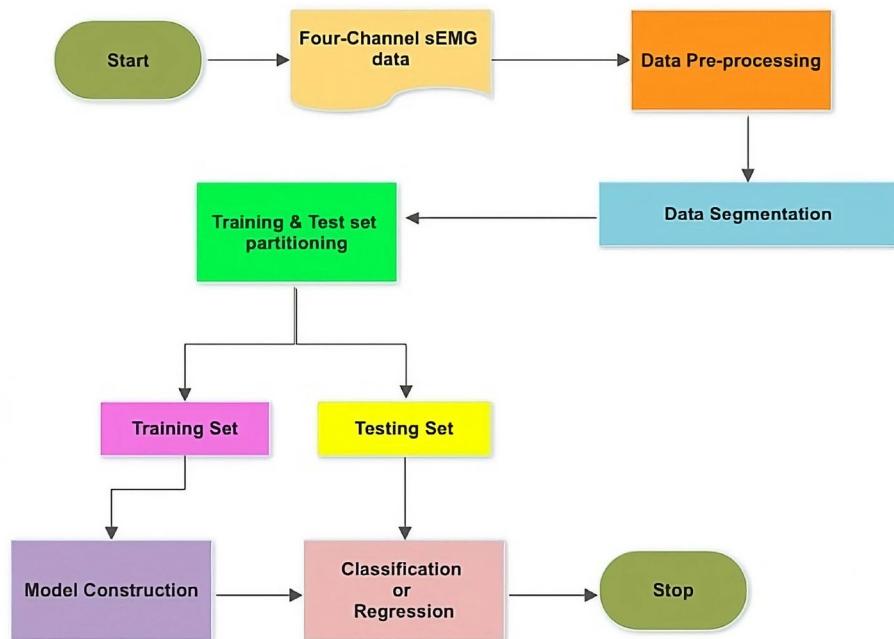
learning approach is shown in Fig. 1. In [13], the authors have generated a dataset using a Microsoft Kinect V2 sensor for 12 different human activities. They have used a hybrid deep learning model for classification, and an average accuracy of 90.89% has been achieved.

Imbalanced class distribution is a significant problem in medical datasets, where the number of samples in different classes varies greatly. This imbalance can lead to prejudiced results towards the majority class, negatively impacting diagnosis accuracy [14]. Therefore, balancing the data by either oversampling the minority class or undersampling the majority class is essential to enhance diagnostic success. In a previous study [15], Rajesh et al. used the AdaBoost ensemble classifier to classify five groups of heartbeats with imbalanced ECG beats. Other studies have also shown that oversampling techniques, such as the Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Technique (ADASYN), can address the class disproportion problem [16–19]. Taft et al., for example, used SMOTE to improve the classification model's performance in identifying adverse medication events in females hospitalised for childbirth and labour. [17].

The major contributions of the authors are ensue:

- Data Standardisation:** The data were scaled using a standard scaler, which rescales the features to have a zero mean and unit variance. It ensures that all features have the same scale and distribution, improving the model's efficiency and convergence.

Fig. 1 Flowchart of general deep learning model



2. **Managing the imbalanced dataset:** The dataset is highly imbalanced. We used ADASYN to balance the dataset.
3. **Data representation:** The four-channel sEMG input signal data were segmented into a 256ms window size with an overlap of 64ms.
4. **Model Design:** Our proposed hybrid ensemble deep neural network architecture for lower extremity recognition utilises an ensemble of CNN and LSTM models trained on four channels of sEMG data. By combining CNN and LSTM, the model can effectively capture human activity data's spatial and temporal aspects.
5. **Performance analysis:** The model has been validated rigorously by investigating the sEMG data and has obtained an average accuracy of 98.8%, 98.3%, and 99.3% for fit individuals for sitting, standing and walking, respectively. Furthermore, an average accuracy of 98.2%, 98.1%, and 99.0% for individuals with knee pathology is a remarkable improvement over the previously published state-of-the-art work.
6. **Statistical Analysis:** The proposed ensemble model has been statistically tested to be significantly different from all competing algorithms using **Friedman test**, **Bonferroni-Dunn test** and **Wilcoxon-signed rank test**.

An ensemble deep learning model comprising CNN-LSTM using four-channel sEMG signal data has been proposed. Our hybrid model utilises both CNN and LSTM to capture human activity data's spatial and temporal characteristics obtained through sEMG. While CNN captures spatial information, LSTM captures temporal information in the data.

Moreover, the proposed work dealt with the imbalanced dataset by employing **Adaptive Synthetic Sampling** (ADASYN) technique. This data augmentation technique addresses the problem of imbalanced data by creating synthetic data for classes with fewer samples. ADASYN uses weight distribution while generating synthetic samples for minority classes. Unlike SMOTE, where the synthetic samples are generated uniformly for all minority classes, ADASYN generated synthetic samples according to their difficulty in learning. Hence, more synthetic samples are generated for minority classes that are harder to learn and most likely to be misclassified. The ADASYN algorithm has been shown to improve the performance of classifiers on imbalanced datasets by effectively increasing the size of the minority class and reducing the bias towards the majority class. None of the previous work published on this dataset has dealt with the problem of imbalanced classes, which can make the model overly biased towards the majority class. Moreover, none of the previously published work has performed statistical tests like the **Bonferroni**-

Dunn test, the **Friedman test** or **Wilcoxon test** to demonstrate whether the proposed hybrid ensemble model is significantly different from the existing models or not. Our proposed model (ensemble) not only outperforms existing competing algorithms but is also proven to be significantly different statistically. The same has been demonstrated in the results (Sect. 6) of this paper.

The remaining portion of this article is laid out as follows. Section 2 comprises the related work of lower extremity activity recognition using wearable sensors. Section 3 contains the description of the individual components of the proposed hybrid ensemble model. Section 4 contains the comprehensive description of the architecture and functionality of our model. Section 5 is the methodology section that entails data collection and pre-processing information. Section 6 outlines the formulas for evaluating the model's performance. It also comprises a detailed description of the results obtained and provides a comparative analysis with other cutting-edge work. Section 6 also gives the results of all statistical tests performed. Lastly, Section 7 comprises the conclusion and the future scope.

2 Literature review

Extensive research has been conducted on gait analysis and human activity recognition. The data can be collected as sensor-based or video-based [20]. Both of these methodologies have their benefits and drawbacks. Sensor-based systems are more successful at capturing the tiny subtleties of human motion that may be difficult to see in video frames. Gait analysis can be performed on data obtained from IMU sensors. IMUs typically comprise accelerometers, gyroscopes, and sometimes magnetometers and can be worn on various body parts to capture motion data [10, 21–23]. In [24], Naik et al. investigated 11 fit individuals and 11 with knee disorders. Independent Component Analysis (ICA) separated the sEMG signals into independent components representing individual muscle activations. Six time-domain features were extracted, and an ensemble-based modelling approach was later used for classification. Fit subjects had a mean classification accuracy of 96.1%, while people with knee problems had an accuracy of 86.2%. Zhang et al. introduced a new technique for identifying multi-channel electromyography (EMG) signals using noise-assisted multivariate empirical mode decomposition (NA-MEMD) in [25]. The NA-MEMD technique was used to partition multi-channel EMG data into a set of intrinsic mode functions (IMFs) that capture the signal's numerous frequency components. However, their study was limited to fit participants and was not extended to participants with knee disorders. The

average classification activity obtained was 79% for walking and 83% for both sitting and standing, respectively. In [12], Gautam et al. proposed a combination of convolutional and recurrent neural networks (LRCN) for the recognition of lower extremity movements, prediction of knee joint angles, and classification of lower extremity activities using surface electromyography (sEMG) signals for both fit and participants with known knee disorders. According to [12], the authors reported mean classification accuracy for fit and people with knee ailments of 98.1% and 92.4%, respectively. None of the previous work [12, 24, 25] dealt with a significant problem of an imbalanced dataset. In [26], the authors have proposed a deep neural network model for the consecutive estimation of lower extremity motions using sEMG signals. The model comprises multiple branches, each processing a different segment of the sEMG signal. The outputs from these branches are then combined to produce the final prediction. The model achieves a mean classification accuracy of 90.92% for speed-dependent and 85.4% for speed-independent. In [27], the authors have used the sEMG data of 11 fit participants and 11 participants with known knee disorders. They have denoised the data and obtained time-domain features. Various anomaly detection techniques were used to enhance the model's classification accuracy. Various machine learning classifiers, notably random forest and light gradient boosting, were used to classify the activities. The best classification accuracy of 98.5% was obtained using the iforest anomaly detection algorithm with a light gradient boosting machine. The LDA-PSO-LSTM algorithm combines three techniques: long-short-term memory (LSTM) neural networks, particle swarm optimisation (PSO), and linear discriminant analysis (LDA). To extract discriminant characteristics from sEMG data, the LDA is used. The PSO algorithm is then used to optimise the hyperparameters of the LSTM, which is employed to recognise the gait phases [28]. In [29], authors used a convolutional network for feature selection from the denoised data and a kernel extreme learning machine to classify lower limb activity. The accuracy reported was 95.90% for classification. In [30], the authors have proposed a bimodal hybrid classifier on IMU sensor data for HAR, which can improve the robustness and accuracy of the recognition system. However, the approach may require a relatively large amount of training data to achieve high accuracy. Secondly, the approach may be affected by sensor placement and calibration, which can affect the quality of the motion data.

3 Deep neural networks (DNNs)

Deep neural network (DNN) is a form of artificial neural network (ANN) that consists of multiple hidden layers between the input and output layers. There are various deep learning models like convolution neural network (CNN), long short-term memory (LSTM) and bi-directional LSTM, etc. In this section, the individual components of the proposed hybrid ensemble deep learning model are detailed.

3.1 Convolution neural network (CNN)

Convolutional neural networks (CNNs) perform various operations such as convolution, pooling, dropout, fully connected layer and activation functions.

- Convolution:** The convolution operation involves sliding a filter or kernel over the input data and performing element-wise multiplication, followed by a summation. The equation for the convolution operation is Eq. 1
- Pooling:** Pooling is a downsampling operation that reduces the spatial size of the activation maps while preserving the essential features. The two most common types of pooling operations are maximum pooling and average pooling. Maximum pooling selects the maximum value within a window of pixels, while average pooling calculates the average value within the window. In our proposed algorithm, maximum pooling has been employed.
- Activation Function:** An activation function adds nonlinearity to the network after each convolution or pooling operation. The most popular activation function is the rectified linear unit, denoted as ReLU, which returns a maximum of 0 or the input value. ReLU and softmax activation functions have the equations Eqs. 2 and 3, respectively.
- Fully Connected Layers:** After several convolutional and pooling layers, the feature maps are flattened into a vector and passed through one or more fully connected layers. These layers are similar to feed-forward neural networks, where each neuron is connected to all neurons in the previous layer. The output of the final dense layer is then supplied to a softmax layer for classification. The equation for the same is Eq. 4.
- Dropout** Throughout the training process, dropout neurons are selected at random and eliminated by setting their synaptic values to zero in every layer. The pace at which these synaptic are lost is known as the dropout rate. Additionally, it speeds up a model's learning process and inhibits overfitting.

Fig. 2 Threefold cross-validation architecture

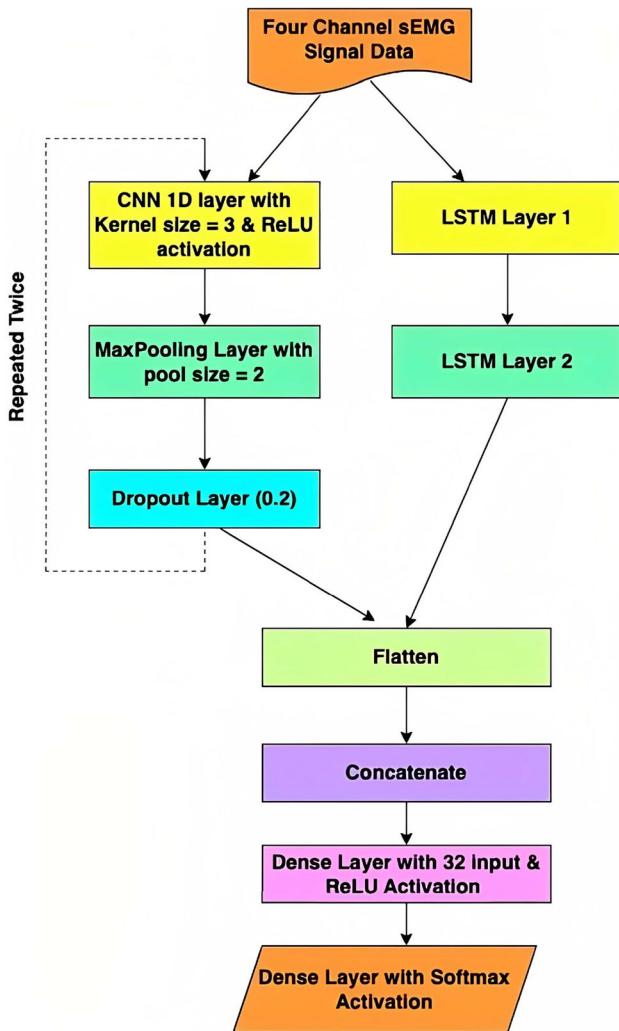
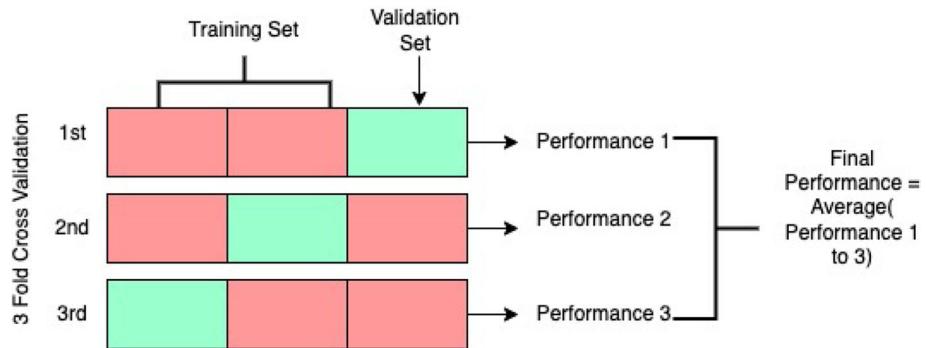


Fig. 3 Architecture of our proposed model

$$c(t_s) = (a * b)(t_s) \quad (1)$$

$$\text{ReLU}(x) = \max(0, x) \quad (2)$$

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^K \exp(z_j)} \quad (3)$$

$$X = Wz + \text{bias} \quad (4)$$

where $c(t_s)$ denotes the convolution of a and b , depicted by $(a * b)(t_s)$, and is defined as the integral of the product of the two functions, shifted by a parameter t_s . Furthermore, $\exp z_i$ and $\exp z_j$ denote the standard exponential function for input and output vector, respectively, and k denotes the number of categories.

3.2 Long short term memory (LSTM)

It is commonly employed in analysis of time-series and natural language processing.

- Input Gate:** The input gate controls what data from the current input are stored in the memory cell. The input gate's equation is Eq. 5.
- Forget Gate:** The forget gate determines which information from the previous memory cell should be discarded. Equation 6 gives the equation for the forget gate.
- Output Gate:** It identifies which information from the current input and previous memory cell should be outputted. The equation for the output gate is Eq. 7.
- Cell State:** It is the long-term memory of the LSTM. It is updated based on the forget, input and output gates. The equations for the same are Eqs. 8 and 9.
- Hidden State:** The hidden state is the short-term memory of the LSTM. It is calculated based on the cell state and the output gate. The equation for the hidden state is shown in Eq. 10.

$$h_{ts} = \sigma(W_{\text{input}} * [h_{ts-1}, x_{ts}] + b_{\text{input}}) \quad (5)$$

$$f_{ts} = \sigma(W_{\text{forget}} * [h_{ts-1}, x_{ts}] + b_{\text{forget}}) \quad (6)$$

$$o_{ts} = \sigma(W_{\text{output}} * [h_{ts-1}, x_{ts}] + b_{\text{output}}) \quad (7)$$

$$c_{ts} = \tanh(W_c * [h_{ts-1}, x_{ts}] + b_c) \quad (8)$$

$$c_{ts} = f_{ts} * c_{ts-1} + i_{ts} * \tilde{c}_{ts} \quad (9)$$

$$h_{ts} = o_{ts} * \tanh(c_{ts}) \quad (10)$$

where f_{ts} , o_{ts} and h_{ts} denote the forget, output and input gate, respectively. W implies the weight for the corresponding gate (g), h_{ts} denotes the previous LSTM block's output at time step $ts-1$, x_{ts} denotes the input at current time step ts , and b implies the bias for the corresponding gate (g). c_{ts} implies cell state at time step ts , and \tilde{c}_{ts} denotes the candidate for cell state at time step ts .

3.3 Hybrid ensemble deep learning

A hybrid deep ensemble model is a model that combines both deep learning and ensemble learning. Basic deep-learning models can learn complex patterns in data but may suffer from overfitting and have difficulty generalising to new data. Ensemble learning, on the other hand, combines multiple models to improve performance by leveraging the strengths of each deep learning model and reducing the risk of overfitting. In a hybrid deep ensemble model, multiple deep learning models are trained on the same data but with different architectures, hyperparameters, or random initialisations. The outputs of these models are then combined. This approach can improve the performance of deep learning models by reducing the risk of overfitting and increasing the diversity of the models. It can also provide more robust predictions and better generalisation of new data. This proposed work uses both LSTM and CNN to create a deep hybrid ensemble model that considers the data's spatial and temporal aspects.

4 Proposed model

The ensemble model is made up of LSTM and CNN models that run in tandem. Two convolutional layers precede a maximum pooling layer and a dropout layer in the model design. The convolutional layers employ 3×1 filters with increasing filters, commencing with 32 and doubling to 64. Using max pooling layers diminishes the

Table 1 Model architecture

Layer type	Output shape	No. of parameters
Input Layer (CNN)	(None, 256, 4)	0
Conv1D_1	(None, 254, 32)	416
MaxPooling1D_1	(None, 127, 32)	0
Dropout_1	(None, 127, 32)	0
Conv1D_2	(None, 125, 64)	6208
MaxPooling1D_2	(None, 62, 64)	0
Dropout_2	(None, 62, 64)	0
Flatten (CNN)	(None, 3968)	0
Input Layer (LSTM)	(None, 256, 4)	0
LSTM_1	(None, 256, 32)	4736
LSTM_2	(None, 16)	3136
Concatenate	(None, 3984)	0
Dense	(None, 32)	127520
Dense	(None, 3)	99

dimensionality of the feature maps and extracts the most significant characteristics. By randomly turning a percentage of the input units to 0 during training, the dropout layers help circumvent overfitting by manipulating the network to acquire more robust and generalisable features. The LSTM model is split into two layers, each with 32 and 16 units. Finally, the output of the final convolutional layer is transformed into a vector. The outputs of the LSTM and CNN models are concatenated and supplied into a dense layer consisting of 32 units and an activation function of ReLU. The dense layer's output is then transmitted to the dense layer that follows it, which uses a softmax activation function to generate the final estimations for the three categories. To train and access the model on the training set, threefold cross-validation is employed. The 3-fold cross-validation is depicted in Fig. 2. To avoid overfitting, a premature end of the callback is also used. Figure 3 depicts the architecture. Table 1 contains details regarding the model's architecture.

Furthermore, the approach of the suggested study is shown in Algorithm 1.

Algorithm 1 Proposed Hybrid Ensemble Learning-based Human Activity Recognition System

Require: Raw data collected through four channel sEMG signal.

Ensure: Classification accuracy, F1-score, precision and recall values of different lower extremity activities.

- 1: Pre-process the data: standardise and balance it using the ADASYN algorithm and then segment data into a window of size 256ms with 64ms overlap.
- 2: Prepare the dataset by dividing it into training and validation sets.
- 3: Design the ensemble deep learning model using CNN and LSTM.
- 4: Combine the performance of individual models using an ensemble learning classifier.
- 5: Calculate performance measures for various activities, which include precision, recall, accuracy, and F1-score.

Output: The class-wise accuracy, classification report and confusion matrix of the different activities.

Result: The proposed hybrid ensemble learning-based system achieves high accuracy, precision, recall, and F1-score for recognizing different lower extremity activities based on the sEMG data.

5 Experimental setup and details

The content of this section encompasses a description and pre-processing of the dataset used, as well as a schematic diagram illustrating the proposed architecture and the algorithm.

5.1 Data acquisition and setup

This study gathered signal data from 22 male volunteers aged 18 and up. Four sEMG channels and one lower limb goniometer measurement channel are included in the data. The participants were split into two groups: eleven fit people and eleven with knee disease, including one with sciatica pain, four with meniscal rupture and six with an anterior cruciate ligament (ACL) injury. Biometrics Ltd. and Datalog MWX8 with the goniometer SG150B were utilised to obtain data on sEMG and knee joints. To ensure interference-free sampling, the SEMG electrodes are separated by 20 mm, and the input impedance is greater than 10 M ohm. The data capture sampling rate was 1000 Hz, and the range for filtering the data instances was between 20 Hz and 460 Hz. The goniometer is positioned on the outer side of the knee. The sEMG electrode channels were positioned on the rectus femoris (rf), semitendinosus (st), vastus medialis (vm), and biceps femoris (bf). The participants' left and right extremities were selected for fit individuals and those with knee disorders. sEMG data and knee joint angles were collected while the subject engaged in three types of physical tasks: standing while making knee flexion motions, walking at ground level, and sitting while performing knee extension actions. These exercises are commonly performed in daily life and rehabilitative activities and do not necessitate the use of additional weights, dumbbells, or fitness equipment. The database

does not include knee joint angle measurements or the sEMG signal during transitional activities such as sit-to-stand or walk-to-sit.

5.2 Data pre-processing

The four-channel sEMG signal data was first standardised to have unit variance, and zero mean to ensure the faster convergence of gradient descent and to preclude the overfitting problem [31]. We also employed the ADASYN algorithm to balance the imbalanced dataset. ADASYN creates synthetic samples of the classes with a smaller number of samples. It was done to ensure the model is impartial and unbiased towards the majority class. Later, the four-channel sEMG signal data were segmented into a 256ms window size with 64 ms overlap using a sliding window approach, as per the research published by [24]. We employed k-fold cross-validation, where $k = 3$, to perform an equitable comparison with existing cutting-edge work. To create k-folds, we randomly divided the dataset into k equal subgroups and repeated the process k times. One of the k subsets is designated as the testing or validation set in each fold, whereas the remaining k-1 subsets are designated as the learning set. The final result is a calculation of the average classification accuracy over k-folds.

6 Results and discussion

6.1 Performance evaluation criteria

The efficiency of the suggested hybrid ensemble model is evaluated using accuracy, F1_score, recall, and precision. The following equations give the formula used for each of the performance evaluation criteria:

$$\text{Prec} = \frac{\text{Tru_Pos}}{\text{Tru_Pos} + \text{False_Pos}} \quad (11)$$

$$\text{Rec} = \frac{\text{Tru_Pos}}{\text{True_Pos} + \text{Fal_Neg}} \quad (12)$$

$$\text{F1_score} = 2 * \frac{\text{Prec} * \text{Rec}}{\text{Prec} + \text{Rec}} \quad (13)$$

$$\text{Acc} = \frac{\text{Tru_Pos} + \text{Tru_Neg}}{\text{Tru_Pos} + \text{Fal_Pos} + \text{Fal_Neg} + \text{Tru_Neg}} * 100 \quad (14)$$

where Acc denotes accuracy, Prec denotes precision, Rec denotes recall, Tru_Pos denotes true positive, Tru_Neg denotes true negative, Fal_Pos denotes false positive and Fal_Neg denotes false negative.

Apart from the metrics mentioned above, we have also performed statistical tests, namely Friedman test [32], Bonferroni-Dunn test [33] and Wilcoxon test [34], to compare our proposed ensemble model with other competing algorithms.

6.1.1 Comparative analysis

The effectiveness of the suggested model was assessed using the aforementioned equations: Eqs. 11–14. The results obtained have been consolidated in Tables 2 and 3. Here, Tables 2 and 3 show the individual subject analysis for healthy and subjects with knee pathology on three different activities, namely walking, standing and sitting obtained by proposed hybrid ensemble model. However, the point is that none of the previously published state-of-the-art work has dealt with the imbalanced dataset problem. Hence, it would be inappropriate to compare the accuracy, as accuracy alone is not the best metric to evaluate the performance of a model, especially when the class distribution is imbalanced. The average classification accuracy, precision, recall and f1_score obtained by the proposed hybrid ensemble model are 98.8%, 98.8%, 98.7% and 98.7%, respectively, for all the healthy participants and 98.4%, 98.4%, 98.3% and 98.3% for the participants with knee pathology. Moreover, this paper also includes the participant-wise classification accuracy for fit individuals obtained by the proposed hybrid ensemble model in Table 4. Table 4 also compares the suggested model with

Table 2 Average precision, accuracy, F1_score and recall obtained of each healthy subject for each activity

Subject	Precision			Recall			F1_score			Accuracy		
	Walking	Standing	Sitting	Walking	Standing	Sitting	Walking	Standing	Sitting	Walking	Standing	Sitting
1	99.6 ± 0.7	100.0 ± 0	100.0 ± 0	100.0 ± 0	99.6 ± 0.7	100.0 ± 0	99.8 ± 0.3	99.8 ± 0.4	100.0 ± 0	99.6 ± 0.7	100.0 ± 0	100.0 ± 0
2	100.0 ± 0	99.2 ± 1.4	98.3 ± 3.0	98.7 ± 2.3	99.6 ± 0.7	99.1 ± 1.5	99.3 ± 1.2	99.4 ± 1.1	98.7 ± 2.3	100.0 ± 0	99.2 ± 1.4	98.3 ± 3.0
3	98.8 ± 2.1	97.2 ± 4.9	96.2 ± 4.9	98.8 ± 2.1	99.2 ± 0.6	92.7 ± 12.7	98.8 ± 2.1	98.2 ± 2.7	94.3 ± 9.0	98.8 ± 2.1	97.2 ± 4.9	96.2 ± 4.9
4	99.1 ± 1.5	96.3 ± 6.4	98.0 ± 3.5	97.9 ± 3.7	99.6 ± 0.7	95.4 ± 8.0	98.5 ± 2.6	97.9 ± 3.7	96.6 ± 5.9	99.1 ± 1.5	96.3 ± 6.4	98.0 ± 3.5
5	99.5 ± 0.8	99.1 ± 1.6	99.6 ± 0.7	99.1 ± 1.6	100.0 ± 0	99.2 ± 1.5	99.3 ± 1.2	99.5 ± 0.8	99.4 ± 1.1	99.5 ± 0.8	99.1 ± 1.6	99.6 ± 0.7
6	96.8 ± 5.5	94.1 ± 9.4	99.5 ± 0.8	95.8 ± 6.3	97.5 ± 0.7	95.4 ± 7.9	96.3 ± 5.9	95.7 ± 7.0	97.3 ± 4.6	96.8 ± 5.5	94.1 ± 9.4	99.5 ± 0.8
7	100.0 ± 0	99.2 ± 1.4	99.6 ± 0.8	100.0 ± 0	99.6 ± 0.7	99.1 ± 1.5	100.0 ± 0	99.4 ± 1.1	99.3 ± 1.2	100.0 ± 0	99.2 ± 1.4	99.6 ± 0.8
8	100.0 ± 0	98.3 ± 0.0	98.8 ± 2.1	100.0 ± 0	99.2 ± 1.5	97.6 ± 0	100.0 ± 0	98.7 ± 0.8	98.2 ± 1.0	100.0 ± 0	98.3 ± 0.0	98.8 ± 2.1
9	100.0 ± 0	99.7 ± 0.5	98.4 ± 2.8	98.0 ± 2.8	100.0 ± 0	100.0 ± 0	99.0 ± 1.4	99.9 ± 0.3	99.2 ± 1.4	100.0 ± 0	99.7 ± 0.5	98.4 ± 2.8
10	100.0 ± 0	99.3 ± 0.6	99.7 ± 0.6	99.8 ± 0.3	99.6 ± 0.8	99.7 ± 0.3	99.9 ± 0.2	99.5 ± 0.7	99.7 ± 0.4	100.0 ± 0	99.3 ± 0.7	99.7 ± 0.6
11	99.0 ± 1.9	99.2 ± 1.3	98.3 ± 3.0	99.3 ± 1.3	98.1 ± 3.3	99.4 ± 1.1	99.1 ± 1.6	98.7 ± 2.3	98.8 ± 2.1	99.0 ± 1.9	99.2 ± 1.4	98.3 ± 3.0
Average	99.3 ± 1.1	98.3 ± 2.5	98.8 ± 2.0	98.8 ± 1.9	99.3 ± 1.2	98.0 ± 3.1	99.1 ± 1.5	98.8 ± 1.9	98.3 ± 2.6	99.3 ± 1.1	98.3 ± 2.5	98.8 ± 2.0

Table 3 Average precision, accuracy, F1_score and recall obtained of each subject with knee pathology for each activity

Subject	Precision			Recall			F1_score			Accuracy		
	Walking	Standing	Sitting	Walking	Standing	Sitting	Walking	Standing	Sitting	Walking	Standing	Sitting
1	99.7 ± 0.5	99.3 ± 1.3	99.4 ± 1.0	99.3 ± 0.2	100.0 ± 0	99.6 ± 0.8	99.5 ± 0.8	99.6 ± 0.6	99.5 ± 0.9	99.7 ± 0.5	99.3 ± 1.3	99.4 ± 1.0
2	99.5 ± 0.8	98.1 ± 3.3	98.0 ± 2.3	100.0 ± 0	97.7 ± 2.9	98.0 ± 3.4	99.8 ± 0.4	98.0 ± 3.1	98.0 ± 2.8	99.5 ± 0.8	98.1 ± 3.3	98.0 ± 2.3
3	99.7 ± 0.4	98.7 ± 1.8	98.8 ± 2.1	99.3 ± 1.3	99.3 ± 0.8	98.4 ± 1.5	99.4 ± 0.5	99.0 ± 1.3	98.6 ± 1.8	99.7 ± 0.4	98.7 ± 1.8	98.8 ± 2.1
4	99.9 ± 0.2	99.9 ± 0.2	98.9 ± 1.1	99.4 ± 0.5	99.9 ± 0.2	99.6 ± 0.7	99.7 ± 0.3	99.9 ± 0.2	99.2 ± 0.9	99.9 ± 0.2	99.9 ± 0.2	98.9 ± 1.1
5	99.6 ± 0.7	98.0 ± 1.5	99.3 ± 0.6	98.6 ± 0.9	99.7 ± 0.4	99.3 ± 1.2	99.1 ± 0.7	98.9 ± 1.0	99.3 ± 0.8	99.6 ± 0.7	98.0 ± 1.5	99.3 ± 0.6
6	98.6 ± 2.5	94.9 ± 8.9	97.9 ± 3.7	94.4 ± 9.7	99.7 ± 0.5	95.5 ± 7.7	96.3 ± 6.4	97.1 ± 5.0	96.7 ± 5.8	98.6 ± 2.5	94.9 ± 8.9	97.9 ± 3.7
7	100.0 ± 0	98.7 ± 2.3	98.4 ± 1.9	100.0 ± 0	98.2 ± 2.1	98.8 ± 2.1	100.0 ± 0	98.4 ± 2.2	98.6 ± 2.0	100.0 ± 0	98.7 ± 2.3	98.4 ± 1.9
8	98.4 ± 1.9	96.7 ± 2.9	97.0 ± 5.1	95.0 ± 5.2	98.7 ± 1.3	98.5 ± 2.7	96.6 ± 3.4	97.7 ± 1.6	97.7 ± 3.9	98.4 ± 1.9	96.7 ± 2.9	97.0 ± 5.1
9	96.0 ± 5.5	97.5 ± 3.9	97.4 ± 4.4	98.0 ± 3.0	100.0 ± 0	90.6 ± 14.1	97.0 ± 4.2	98.7 ± 2.0	93.7 ± 9.8	96.0 ± 5.5	97.5 ± 3.9	97.4 ± 4.4
10	99.1 ± 1.6	99.4 ± 1.1	97.0 ± 5.2	96.6 ± 5.8	99.4 ± 1.1	99.4 ± 1.0	97.8 ± 3.8	99.4 ± 1.1	98.2 ± 3.2	99.1 ± 1.6	99.4 ± 1.1	97.0 ± 5.2
11	99.0 ± 1.7	98.3 ± 1.9	97.6 ± 4.1	97.9 ± 3.6	99.7 ± 0.4	97.4 ± 3.4	98.5 ± 2.7	99.0 ± 1.1	97.5 ± 3.7	99.0 ± 1.7	98.3 ± 1.9	97.6 ± 4.1
Average	99.0 ± 1.4	98.1 ± 2.6	98.2 ± 2.9	98.0 ± 2.8	99.3 ± 0.9	97.7 ± 3.5	98.5 ± 2.1	98.7 ± 1.7	97.9 ± 3.2	99.0 ± 1.4	98.1 ± 2.6	98.2 ± 2.9

other cutting-edge work, namely MyoNet [12], ICA-EBM [24], NA-MEMD [25], on the same dataset for fit participants. The average classification accuracy of walking activity obtained by the proposed model is $99.3\% \pm 1.1$ as opposed to $98.2\% \pm 1.6$ in [12], $96.0\% \pm 1.3$ in [24], and 79.0% in [25]. Furthermore, the average classification accuracy for standing activity for fit participants obtained by the proposed hybrid ensemble model is $98.3\% \pm 2.5$ and for sitting is $98.8\% \pm 2.0$ which shows a significant improvement over the previously published state-of-the-art work: $97.7\% \pm 1.3$, $98.4\% \pm 1.4$ [12] and $96.2\% \pm 1.2$, $96.2\% \pm 1.1$ in [24], 83.0% and 83.0% in [25], respectively. The comparison of mean classification accuracy obtained using the proposed hybrid ensemble work with other state-of-the-art work for healthy participants presented in Table 4 is further illustrated in Fig. 4a.

Our proposed model also shows promising results for classifying various lower limb activities performed by individuals with knee pathology. Table 3 comprises the class-wise average accuracy, f1_score, precision, and recall for each individual with known knee pathology. Moreover, Table 5 compares the proposed model's performance on data obtained from participants with knee pathology with

existing cutting-edge work, namely MyoNet [12] and ICA-EBM [24]. The mean accuracy of $98.2\% \pm 2.9$, $98.1\% \pm 2.6$, $99.0\% \pm 1.4$ for sitting, standing, and walking activities for participants having knee disorders using the proposed hybrid ensemble model. It shows a significant improvement from the average accuracy obtained by previous cutting-edge work; 86.4% , 85.5% , 86.6% in [24], 92.2% , 92.3% , 92.8% in [12] for walking, standing, and sitting, respectively. The details presented in Table 5 can also be seen in Fig. 4b. The previously published state-of-the-art work has struggled to perform activity classification for subjects with knee pathology, as for such individuals, each pathology is different and can lead to different functional limitations and activity restrictions. Therefore, developing a single classification model that accounts for different pathologies in different individuals is challenging. Our proposed model achieved an average accuracy of 98.4% across all individuals with pathology as opposed to 92.4% in [12], 86.2% in [24]. The proposed model outperforms the previously published state-of-the-art work [24] by 12.2% and [12] by 6.0% . Moreover, it is worth noting that in the real world, signals contain noise, and our proposed method works on noisy signals and does not

Table 4 Comparative analysis of classification accuracy (%) for different methods on the three activity classes for the healthy subjects dataset

Subject	MyoNet [12]			ICA-EBM [24]			NA-MEMD [25]			Our method		
	Walking	Standing	Sitting	Walking	Standing	Sitting	Walking	Standing	Sitting	Walking	Standing	Sitting
1	98.2 ± 1.8	98.5 ± 1.2	97.3 ± 1.8	95.6 ± 1.2	96.3 ± 1.3	96.2 ± 1.3	78.0	76.0	84.0	99.6 ± 0.7	100.0 ± 0	100.0 ± 0
2	97.6 ± 2.1	97.2 ± 1.6	98.6 ± 1.7	96.3 ± 1.3	96.4 ± 1.3	96.4 ± 1.2	79.0	76.0	84.0	100.0 ± 0	99.2 ± 1.4	98.3 ± 3.0
3	97.3 ± 1.7	93.9 ± 1.7	99.2 ± 0.6	95.8 ± 1.4	96.2 ± 1.2	96.1 ± 1.2	83.0	80.0	88.0	98.8 ± 2.1	97.2 ± 4.9	96.2 ± 4.9
4	98.4 ± 0.8	97.3 ± 1.4	99.1 ± 0.2	96.4 ± 1.3	96.3 ± 1.3	96.3 ± 1.1	76.0	76.0	80.0	99.1 ± 1.5	96.3 ± 6.4	98.0 ± 3.5
5	99.1 ± 1.5	99.6 ± 2.2	98.2 ± 2.1	96.1 ± 1.4	95.6 ± 1.1	96.2 ± 1.0	81.0	82.0	76.0	99.5 ± 0.8	99.1 ± 1.6	99.6 ± 0.7
6	98.1 ± 1.6	97.0 ± 1.7	99.5 ± 0.6	95.6 ± 1.6	96.1 ± 1.4	96.3 ± 1.4	84.0	80.0	75.0	96.8 ± 5.5	94.1 ± 9.4	99.5 ± 0.8
7	97.5 ± 2.2	98.5 ± 1.1	98.2 ± 2.1	96.2 ± 1.1	95.8 ± 1.2	96.1 ± 1.1	79.0	89.0	77.0	100.0 ± 0	99.2 ± 1.4	99.6 ± 0.8
8	99.4 ± 0.6	95.7 ± 0.7	98.9 ± 1.6	96.3 ± 1.5	96.2 ± 1.3	96.2 ± 1.4	79.0	89.0	86.0	100.0 ± 0	98.3 ± 0.0	98.9 ± 2.1
9	96.5 ± 2.6	98.6 ± 1.5	97.3 ± 2.6	95.8 ± 1.3	96.4 ± 1.3	96.1 ± 1.2	79.0	92.0	87.0	100.0 ± 0	99.7 ± 0.5	98.4 ± 2.8
10	100 ± 0.2	99.3 ± 1.6	98.5 ± 1.4	95.9 ± 1.2	96.5 ± 1.2	96.3 ± 1.2	72.0	89.0	91.0	100.0 ± 0	99.3 ± 0.7	99.7 ± 0.6
11	99.0 ± 1.5	99.3 ± 1.3	98.3 ± 1.3	95.8 ± 1.2	96.2 ± 1.5	96.3 ± 1.3	79.0	88.0	86.0	99.0 ± 1.9	99.3 ± 1.4	98.3 ± 3.0
Average	98.2 ± 1.6	97.7 ± 1.3	98.4 ± 1.4	96.0 ± 1.3	96.2 ± 1.2	96.2 ± 1.1	79.0	83.0	83.0	99.3 ± 1.1	98.3 ± 2.5	98.8 ± 2.0

employ any denoising technique. Furthermore, the proposed model showed an average precision, recall and f1_score of 98.4%, 98.3% and 98.4%, as opposed to 93.4%, 92.6% and 92.9%, respectively, across subjects with knee pathology. The comparison of the overall performance of the proposed model with the MyoNet method [12] for healthy subjects and subjects with knee pathology has been shown in Fig. 4c and d, respectively. It can be noted that the proposed model outperforms the Myonet method [12] in terms of recall, F1_score and accuracy for healthy subjects and in terms of accuracy, recall, precision and F1_score for subjects the knee pathology.

In this research, a subject-wise comparison of the suggested ensemble model with CNN-only and LSTM-only models for healthy subjects and subjects with knee disorder is also shown in Tables 6 and 7, respectively. The mean accuracy for walking, standing, and sitting using the CNN-only model is 98.3%, 96.7% and 96.0% for fit individuals and 97.6%, 97.4% and 97.2%, respectively, for subjects with knee disorders. Similarly, the LSTM-only models achieved an average accuracy of 86.0%, 81.7% and 91.2% for walking, standing, and sitting activities for fit participants and 80.1%, 79.3% and 78.3%, respectively, for

participants with knee disorder. It can be noted that LSTM_only suffers massively in classifying the lower limb-related activities for both fit and individuals with knee pathology, and CNN_only performs better than LSTM_only but the hybrid deep ensemble model performs the best as it captures both the spatial and temporal aspect of the data. The comparison of accuracy obtained using CNN_only, LSTM_only and proposed hybrid model have also been further illustrated in Fig. 4e and f.

6.1.2 Statistical test

To compare the efficiency of the suggested work with that of other cutting-edge approaches, we have performed **friedman** [32], **bonferroni-dunn** [33] and **Wilcoxon signed rank test** [34] on the proposed ensemble model, ICA [24], MyoNet [12]. In addition to the proposed ensemble model, we have also compared the CNN-only and LSTM-only models. We have not statistically compared our proposed model with NA-MEMD [25] because in [25], the study was limited to healthy subjects, and pathological subjects were not considered.

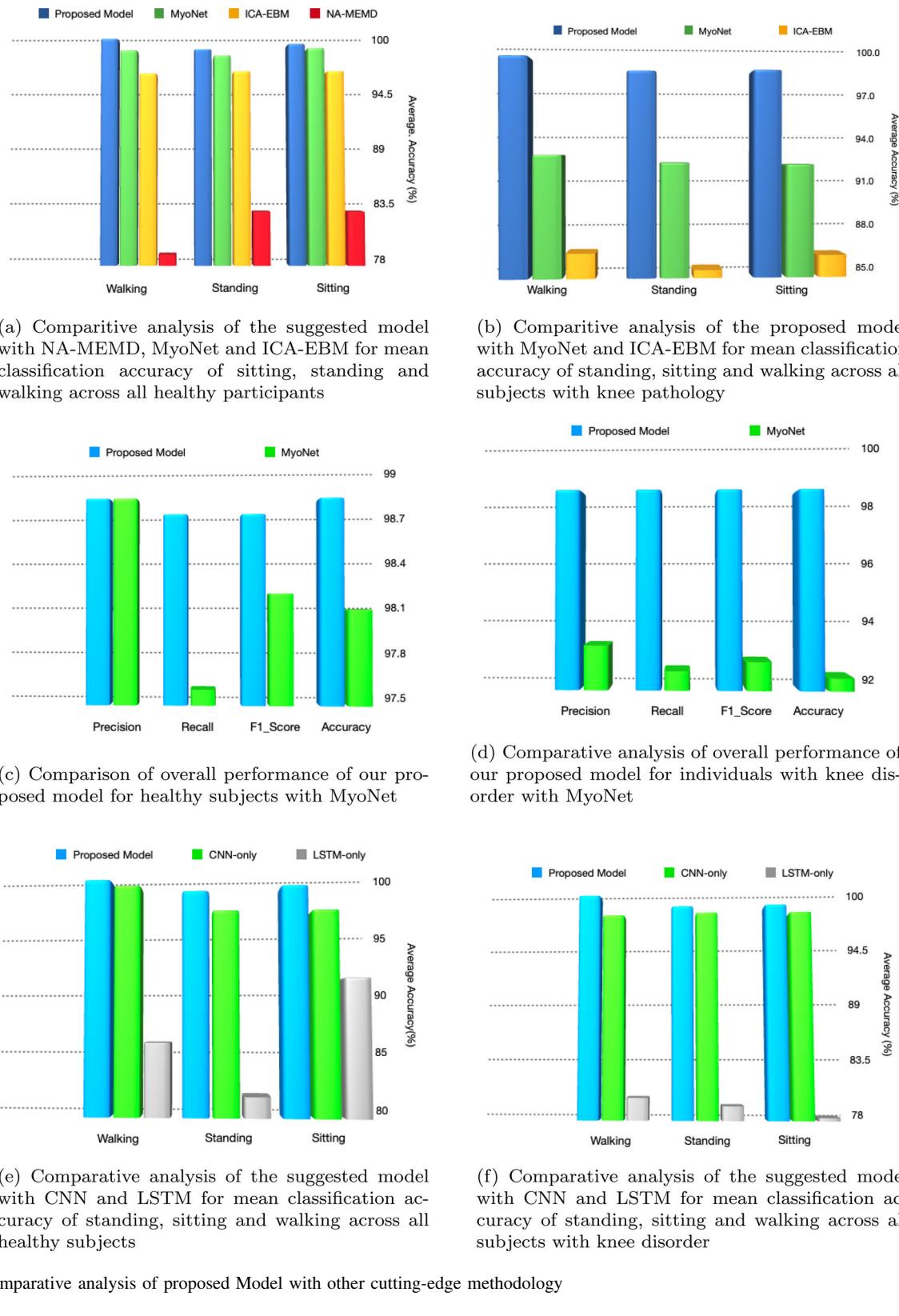


Fig. 4 Comparative analysis of proposed Model with other cutting-edge methodology

The Friedman test is a widely accepted statistical test to compare the performance of multiple models. Let r_j^i signify the rank of the i th algorithm over the j th dataset, r_i denote

the mean rank of the i th method, and n denotes the count of sets or samples. k signifies the count of competing

Table 5 Comparative analysis of classification accuracy (%) for subjects with knee pathology

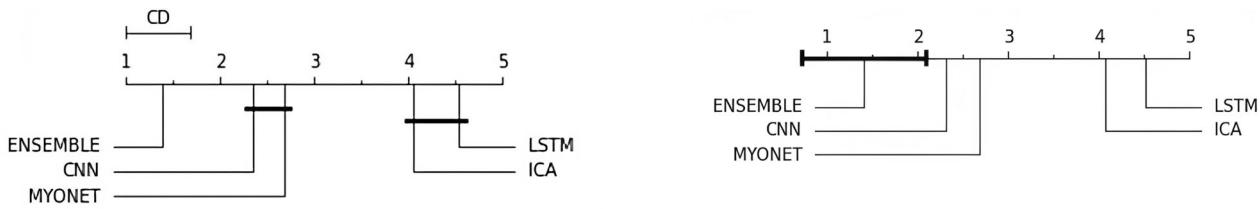
Subject	MyoNet [12]			ICA-EBM [24]			Our method		
	Walking	Standing	Sitting	Walking	Standing	Sitting	Walking	Standing	Sitting
1	92.7 ± 1.8	92.1 ± 1.5	90.7 ± 2.1	87.7 ± 1.3	85.8 ± 1.3	86.4 ± 1.4	99.7 ± 0.5	99.3 ± 1.3	99.4 ± 1.0
2	93.6 ± 2.3	91.6 ± 1.6	92.9 ± 1.2	86.8 ± 1.2	85.2 ± 1.2	86.5 ± 1.2	99.5 ± 0.8	98.1 ± 3.3	98.0 ± 2.3
3	94.2 ± 0.7	93.4 ± 0.4	92.6 ± 1.7	86.4 ± 1.3	85.4 ± 1.3	86.7 ± 1.3	99.7 ± 0.4	98.7 ± 1.8	98.8 ± 2.1
4	92.7 ± 2.1	92.9 ± 1.1	91.8 ± 2.2	86.4 ± 1.4	85.5 ± 1.2	86.4 ± 1.2	99.9 ± 0.2	99.9 ± 0.2	98.9 ± 1.1
5	92.6 ± 2.3	91.3 ± 0.7	92.5 ± 1.8	86.8 ± 1.2	85.4 ± 1.4	86.5 ± 1.6	99.6 ± 0.7	98.0 ± 1.5	99.3 ± 0.6
6	92.9 ± 2.4	91.9 ± 0.9	91.4 ± 0.9	86.3 ± 1.3	85.5 ± 1.1	86.4 ± 1.4	98.6 ± 2.5	94.9 ± 8.9	97.9 ± 3.7
7	93.3 ± 1.8	93.6 ± 2.3	92.9 ± 0.7	86.4 ± 1.4	85.7 ± 1.5	86.3 ± 1.3	100.0 ± 0.0	98.7 ± 2.3	98.4 ± 1.9
8	89.6 ± 2.1	92.5 ± 1.2	92.2 ± 1.81	86.3 ± 1.3	85.4 ± 1.2	86.8 ± 1.2	98.4 ± 1.9	96.7 ± 2.9	97.0 ± 5.1
9	94.1 ± 1.0	87.9 ± 1.7	93.1 ± 0.7	86.7 ± 1.2	85.5 ± 1.4	86.3 ± 1.3	96.0 ± 5.5	97.5 ± 3.9	97.4 ± 4.4
10	93.4 ± 0.6	94.8 ± 1.1	92.6 ± 2.2	86.8 ± 1.5	85.8 ± 1.3	86.2 ± 1.4	99.1 ± 1.6	99.4 ± 1.1	97.0 ± 5.2
11	91.9 ± 2.1	93.3 ± 1.3	92.4 ± 2.4	86.4 ± 1.2	85.5 ± 1.4	86.3 ± 1.2	99.0 ± 1.7	98.3 ± 1.9	97.6 ± 4.1
Average	92.8 ± 1.7	92.3 ± 1.2	92.2 ± 1.6	86.6 ± 1.3	85.5 ± 1.3	86.4 ± 1.3	99.0 ± 1.4	98.1 ± 2.6	98.2 ± 2.9

Table 6 Comparative analysis of suggested Ensemble model with LSTM-only and CNN-only model for healthy subjects

Subject	Ensemble model			CNN-only			LSTM-only		
	Walking	Standing	Sitting	Walking	Standing	Sitting	Walking	Standing	Sitting
1	99.6	100.0	100.0	99.6	99.6	97.8	98.7	97.8	96.2
2	100.0	99.2	98.3	100.0	98.4	98.7	81.8	93.4	90.2
3	98.8	97.2	96.2	97.9	93.5	92.8	94.2	88.1	93.5
4	99.1	96.3	98.0	99.1	92.9	88.9	89.5	73.6	88.2
5	99.5	99.1	99.6	99.1	99.1	97.9	99.5	86.6	83.5
6	96.8	94.1	99.5	97.8	90.5	93.9	41.8	51.1	86.7
7	100.0	99.2	99.6	96.0	96.2	95.6	76.3	66.05	96.8
8	100.0	98.3	98.9	99.2	98.4	96.8	95.6	77.9	90.36
9	100.0	99.7	98.4	95.8	100.0	98.8	86.3	86.8	93.5
10	100.0	99.3	99.7	97.8	98.9	97.8	96.5	95.9	93.3
11	99.0	99.3	98.3	98.8	96.6	97.5	85.8	81.9	90.7
Average	99.3	98.3	98.8	98.3	96.7	96.0	86.0	81.7	91.2

Table 7 Comparative analysis of proposed Ensemble model with LSTM-only and CNN-only model for subjects with knee pathology

Subject	Ensemble model			CNN-only			LSTM-only		
	Walking	Standing	Sitting	Walking	Standing	Sitting	Walking	Standing	Sitting
1	99.7	99.3	99.4	98.0	99.3	99.8	96.3	89.3	91.7
2	99.5	98.1	98.0	99.5	96.4	95.7	68.9	96.9	32.5
3	99.7	98.7	98.8	99.7	97.5	99.4	96.8	86.5	80.5
4	99.9	99.9	98.9	94.1	99.8	98.2	85.3	76.72	74.7
5	99.6	98.0	99.3	99.2	95.3	99.3	94.9	90.7	79.6
6	98.6	94.9	97.9	97.1	97.6	97.6	61.0	83.9	66.9
7	100.0	98.7	98.4	99.2	97.4	97.5	82.5	66.5	93.6
8	98.4	96.7	97.0	98.7	95.1	95.7	88.1	62.4	80.1
9	96.0	97.5	97.4	92.8	97.3	93.2	67.1	65.6	80.2
10	99.1	99.4	97.0	97.6	98.4	95.9	73.3	84.9	82.5
11	99.0	98.3	97.6	97.9	97.8	97.2	66.4	69.5	99.4
Average	99.0	98.1	98.2	97.6	97.4	97.2	80.1	79.3	78.3



(a) Comparison of all algorithms pair-by-pair using the Bonferroni-Dunn test

(b) Comparison of all algorithms using the Bonferroni-Dunn test with the proposed algorithm serving as the control algorithm

Fig. 5 Comparative analysis of each algorithm using the Bonferroni-Dunn test

methods. In the event of a tie between any algorithms, the average rank is shared.

The Friedman statistic(F_r) is evaluated using the Eqs. [15] and [16].

$$F_r = \frac{(n-1)\chi_f^2}{n(k-1) - \chi_f^2} \quad (15)$$

where

$$\chi_f^2 = \frac{12n}{k(k+1)} \left(\sum_{i=1}^k r_i^2 - \frac{k(k+1)^2}{4} \right) \quad (16)$$

In our case, $n = 66$, $k = 5$, chosen level of significance (α) = 0.05.

$$\chi_f^2 = \frac{12 \times 66}{5 \times 6} \times \left(51.529 - \frac{5 \times (6)^2}{4} \right) \quad (17)$$

$$\chi_f^2 = 172.367 \quad (18)$$

$$F_r = \left(\frac{65 \times 172.367}{66 \times 4 - 172.367} \right) = 122.268 \quad (19)$$

Critical value p -value = 1.1415×10^{-36}

This p -value indicates decisive statistical significance, as it is much smaller than the threshold for statistical significance (α) = 0.05. This means we can reject the null hypothesis that the models are not statistically different with high confidence. Additionally, we have used the pairwise comparison method of the Bonferroni-Dunn test [33] to determine whether algorithms are significantly different. Using the proposed model as the governing

algorithm, we have also examined whether it performs noticeably better than rival techniques. Equation 20 determines the critical distance (CD).

$$CD = x_\alpha \left[\frac{k(k+1)}{6n} \right]^{1/2} \quad (20)$$

At the level of significance, $\alpha = 0.05$, the obtained $CD = 0.6876$. The result for the Bonferroni-Dunn test with the proposed algorithm as a control algorithm and a general comparison is shown in Fig. 5a and b, respectively. Figure 5a shows that the CNN-only model and MyoNet [12] are similar and not statistically different, and LSTM-only and ICA [24] are also similar to each other and not statistically significantly different. As depicted in Fig. 5a and b, it is further emphasised that the proposed ensemble model (referred to as ensemble) is not only better performing than all the other competing algorithms in terms of accuracy but is also significantly different from them.

To further validate that the proposed hybrid ensemble model outperforms the other model and is significantly different from existing state-of-the-art work, we have employed **Wilcoxon Signed-Rank Test** [34]. The result of the Wilcoxon signed-rank test has been shown in Table 8. As shown in Table 8, the p – value is significantly less than the significance level $\alpha = 0.05$. The proposed ensemble model is significantly different from other competing algorithms and outperforms each of them.

Table 8 Wilcoxon signed-rank test results for the suggested ensemble model opposed to other models

S. No.	Model	Test statistic	P value	Result
1	Ensemble vs MyoNet [12]	109.5	7.67×10^{-10}	Different
2	Ensemble vs ICA [24]	6.5	3.24×10^{-12}	Different
3	Ensemble vs CNN-only	178.5	9.63×10^{-8}	Different
4	Ensemble vs LSTM-only	3.0	2.77×10^{-12}	Different

Table 9 Comparison with related work

Parameters	[38]	[24]	[25]	[39]	[12]	Our model
Dealt with class imbalance	No	No	No	No	No	Yes
Human intervention in feature selection	Yes	Yes	Yes	Yes	No	No
Number of different features extracted	39	7	16	6	Data-driven feature extraction	Data-driven feature extraction
Walking accuracy (%)	88	91.3	79.0	—	95.5	99.2
Sitting accuracy (%)	94	91.3	83.0	—	95	98.2
Standing accuracy (%)	92	90.8	83.0	—	95.3	98.5
Average accuracy (%)	91.3	91.1	81.6	85	95.2	98.6
Statistical comparison	No	No	No	No	No	Yes

6.2 Discussion

Individuals with knee-related illnesses like osteoarthritis, meniscus tears, and ACL tears frequently struggle with daily activities involving lower extremity motions such as walking, standing, and sitting [35]. Surface electromyography signals obtained from the muscles, like the quads and hamstrings, during motion can aid in diagnosis by clinicians and assist in rehabilitation [36, 37]. Additionally, sEMG signals can help evaluate the improvement in physiotherapy sessions when using a network-based rehabilitation approach.

We have suggested a hybrid ensemble model comprising LSTM and CNN that captures the spatial and temporal aspects of the signal. In [25], Zhang et al. investigated the use of intrinsic mode functions (IMFs) obtained from the decomposition of surface electromyography (sEMG) signals as features for identifying the lower leg motion. They explored the use of three different decomposition methods, namely multivariate empirical mode decomposition (MEMD), noise-assisted MEMD (NA-MEMD) and empirical mode decomposition (EMD), to extract the IMFs from the raw sEMG signal. One of the significant limitations of their work is that they have restricted the scope of their study to the data of healthy subjects and not extended it to subjects with knee disorders. In [24], Naik et al. have extracted six time-domain features and then applied feature dimensionality reduction. Furthermore, the authors used SVM with an RBF kernel to classify lower limb motion in fit subjects and subjects with knee abnormalities. We have compared our suggested ensemble model with previous cutting-edge work and CNN-only and LSTM-only models. The ensemble model outperforms all the other methods with an average accuracy of 99.3%, 98.3% and 98.8% for a walk, stand and sit activity for healthy subjects and 99.0%, 98.1% and 98.2% respectively, for subjects with knee pathology. Moreover, our proposed ensemble model is also statistically proven to differ significantly from all the competing methodologies. For better understanding, the

comparative analysis of the previous cutting-edge techniques with the proposed ensemble model is shown in Table 9.

7 Conclusion and future scope

A hybrid ensemble deep learning model is proposed for classifying lower extremity movement. The proposed hybrid ensemble model significantly improves the recall, f1_score, accuracy, and precision values over the previously published state-of-the-art work. Our proposed hybrid ensemble model provides an average accuracy of 98.8%, precision of 98.8%, f1_score of 98.7% and recall of 98.7%. In addition to the proposed hybrid ensemble model comprising CNN and LSTM models, this work has also standardised the data zero mean and unit variance that leads to faster convergence of the models along with enhancing model's efficiency. The dataset was highly imbalanced, and the previous works had not considered that aspect. An imbalanced dataset may lead to discriminatory behaviour by the model towards the majority class, and hence accuracy may not be the best metric to gauge the model's performance. We have handled the imbalanced dataset problem using the adaptive synthetic oversampling algorithm (ADASYN). Enhancing classification accuracy, our suggested hybrid ensemble model beats all current cutting-edge work, particularly for participants with knee pathologies. For fit participants and participants with knee disorders, our model had a mean classification accuracy of 98.8% and 98.4%, respectively. Additionally, the proposed ensemble model has been also compared with CNN and LSTM individually. And our results show that the proposed hybrid ensemble models works better in catering both temporal and spacial aspect of the data. Moreover, the proposed work has also incorporated statistical tests like the Friedman, Bonferroni-Dunn, and Wilcoxon tests to compare our proposed model with all other competing algorithms. All the statistical tests and performance metrics

proved that our model is superior and significantly different from all other competing algorithms discussed. None of the previous works has standardised the data, handled the imbalanced dataset problem, or compared their algorithms statistically with previous state-of-the-art work. All the points mentioned above have been addressed in the proposed work. In the future, we intend to create a model that can classify activities and predict joint angles. The proposed model can also be validated using a larger dataset. We will additionally employ TinyML to significantly reduce the model's complexity and make it more computationally efficient.

Acknowledgements This work was supported by the Ministry of Education, Government of India, through the project HEFA CSR under Grant SAN/CSR/08/2021-22. The authors would like to thank the volunteers who have participated and contributed in preparation of the dataset and also the creator of the dataset.

Author contributions P.T and V.B contributed to conceptualisation and methodology; P.T contributed to software; P.T, V.B and S.J contributed to validation and writing—original draft preparation and review and editing; V.B and S.J supervised the study. All authors have read and approved the final manuscript.

Funding The work is funded by Ministry of Education, Govt. of India to Dr. Vijay Bhaskar Semwal under Higher Education Financing Agency (HEFA) under CSR Grant with Sanctioned order no- SAN/CSR/08/2021-22.

Data availability The dataset analysed during the current study are available in the UCI Machine Learning repository, <http://archive.ics.uci.edu/ml/datasets/emg+dataset+in+lower+limb>.

Declarations

Conflict of interest The authors state that they do not have any conflicts of interest.

Compliance with Ethical Standards All the ethical issues have been taken care of while writing the manuscript, and we have complied with all the standards to the best of our knowledge.

References

- Kujala UM, Orava S, Parkkari J, Kaprio J, Sarna S (2003) Sports career-related musculoskeletal injuries: long-term health effects on former athletes. *Sports Med* 33:869–75
- Kianifar R, Lee A, Raina S, Kulic D (2017) Automated assessment of dynamic knee valgus and risk of knee injury during the single leg squat. *IEEE J Trans Eng Health Med* 30(5):1–3
- Dua Nidhi, Singh Shiva, Semwal Vijay, Challa Sravan (2022) Inception inspired CNN-GRU hybrid network for human activity recognition. *Multimed Tools Appl* 82:03
- Anjali Gupta and Vijay Bhaskar Semwal (2022) Occluded gait reconstruction in multi person gait environment using different numerical methods. *Multimed Tools Appl* 81(16):23421–23448
- Baker R (2006) Gait analysis methods in rehabilitation. *J Neuroeng Rehabil* 3(1):4
- Muro-De-La-Herran A, Garcia-Zapirain B, Mendez-Zorrilla A (2014) Gait analysis methods: an overview of wearable and non-wearable systems, highlighting clinical applications. *Sensors Basel Switz* 14:3362–94
- Semwal VB, Gaud N, Nandi GC. (2019) Human gait state prediction using cellular automata and classification using elm. In M. Tanveer and Ram Bilas Pachori, editors, *Machine intelligence and signal analysis*, pp 135–145, Singapore, Springer Singapore
- Semwal VB, Gupta A, Lalwani P (2021) An optimized hybrid deep learning model using ensemble learning approach for human walking activities recognition. *J Supercomput* 77(11):12256–12279
- Challa SK, Kumar A, Semwal VB (2021) A multibranch CNN-BiLSTM model for human activity recognition using wearable sensor data. *VIS Comput* 38(12):4095–4109
- Qiu Sen, Fan Tianqi, Jiang Junhan, Wang Zhelong, Wang Yongzhen, Junnan Xu, Sun Tao, Jiang Nan (2023) A novel two-level interactive action recognition model based on inertial data fusion. *Inform Sci* 633:264–279
- Del Din S, Godfrey A, Mazza C, Lord S, Rochester L. (2016) Free-living monitoring of parkinson's disease: lessons from the field. *Movement Disorders*, 31, 06
- Gautam Arvind, Panwar Madhuri, Biswas Dwaipayan, Acharyya Amit (2020) Myonet: a transfer-learning-based LRCN for lower limb movement recognition and knee joint angle prediction for remote monitoring of rehabilitation progress from semg. *IEEE J Trans Eng Health Med* 8:1–10
- Khan IU, Afzal S, Lee JW (2022) Human activity recognition via hybrid deep learning based model. *Sensors* 22(1):323
- Krawczyk Bartosz (2016) Learning from imbalanced data: open challenges and future directions. *Prog Artif Intell* 5:04
- Rajesh KN, Dhuli R (2018) Classification of imbalanced ECG beats using re-sampling techniques and AdaBoost ensemble classifier. *Biomed Signal Process Control* 41:242–54
- Nahar J, Imam T, Tickle KS, Ali AS, Chen YP (2012) Computational intelligence for microarray data and biomedical image analysis for the early diagnosis of breast cancer. *Expert Syst Appl* 39(16):12371–7
- Taft LM, Evans RS, Shyu CR, Egger MJ, Chawla N, Mitchell JA, Thornton SN, Bray B, Varner M (2009) Countering imbalanced datasets to improve adverse drug event predictive models in labor and delivery. *J Biomed Inform* 42(2):356–364
- He H, Bai Y, Garcia E, Li SA (2008) Adasyn: Adaptive synthetic sampling approach for imbalanced learning. pp 1322 – 1328, 07
- Nur Ghaniaviyanto Ramadhan (2021) Comparative analysis of Adasyn-Svm and smote-Svm methods on the detection of type 2 diabetes mellitus. *Sci J Inform* 8(2):276–282
- Jia Hairui, Chen Shuwei (2020) Integrated data and knowledge driven methodology for human activity recognition. *Inform Sci* 536:409–430
- Sekaran SR, Han PY, Yin OS (2023) Smartphone-based human activity recognition using lightweight multiheaded temporal convolutional network. *Expert Syst Appl* 227:120132
- Han Chaolei, Zhang Lei, Tang Yin, Huang Wenbo, Min Fuhong, He Jun (2022) Human activity recognition using wearable sensors by heterogeneous convolutional neural networks. *Expert Syst Appl* 198:116764
- Roetenberg Daniel, Luinge Henk, Slycke Per (2009) Xsens mvn: full 6dof human motion tracking using miniature inertial sensors. *Xsens Motion Technol BV Tech Rep* 3:01
- Naik GR, Selvan SE, Arjunan SP, Acharyya A, Kumar DK, Ramanujam A, Nguyen HT (2018) An ICA-EBM-based sEMG classifier for recognizing lower limb movements in individuals with and without knee pathology. *IEEE Trans Neural Syst Rehabil Eng* 26(3):675–86

25. Zhang Y, Xu P, Li P, Duan K, Wen Y, Yang Q, Zhang T, Yao D (2017) Noise-assisted multivariate empirical mode decomposition for multichannel emg signals. *Biomed Eng Online* 16(1):107
26. Wang Xingjian, Dong Dengpeng, Chi Xiaokai, Wang Shaoping, Miao Yinan, An Mailing, Gavrilov Alexander I (2021) SEMG-based consecutive estimation of human lower limb movement by using multi-branch neural network. *Biomed Signal Process Control* 68:102781
27. Bansal H, Chinagundi B, Rana PS, Kumar N (2022) An ensemble machine learning technique for detection of abnormalities in knee movement sustainability. *Sustainability* 14(20):13464
28. Cai Shibo, Chen Dipei, Fan Bingfei, Mingyu Du, Bao Guanjun, Li Gang (2023) Gait phases recognition based on lower limb SEMG signals using lda-Pso-lstm algorithm. *Biomed Signal Process Control* 80:104272
29. Juan Tu, Dai ZunXiang, Zhao Xiang, Huang Zijuan (2023) Lower limb motion recognition based on surface electromyography. *Biomed Signal Process Control* 81:104443
30. Venkatachalam K, Yang Z, Trojovsky P, Bacanin N, Deveci M, Ding W (2023) Bimodal HAR-An efficient approach to human activity analysis and recognition using bimodal hybrid classifiers. *Inform Sci* 628:542–57
31. Sola J, Sevilla J (1997) Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Trans Nucl Sci* 44(3):1464–8
32. Demšar Janez (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7(1):1–30
33. Olive Jean Dunn (1961) Multiple comparisons among means. *J Am Stat Assoc* 56:52–64
34. Wilcoxon F (1945) Individual comparisons by ranking methods. *Biometrics* 1:196–202
35. Hayashi M, Koga S, Kitagawa T (2023) Effectiveness of rehabilitation for knee osteoarthritis associated with isolated meniscus injury—a scoping review. *Cureus* 14(e34544):02
36. Wren TA, Do KP, Rethlefsen SA, Healy B (2006) Cross-correlation as a method for comparing dynamic electromyography signals during gait. *J Biomech* 39(14):2714–8
37. Al-Ayyad M, Owida HA, De Fazio R, Al-Naami B, Visconti P (2023) Electromyography monitoring systems in rehabilitation: a review of clinical applications, wearable devices and signal acquisition methodologies. *Electronics* 12(7):1520
38. Herrera-Gonzalez M, Martinez-Hernandez GA, Rodriguez-Sotelo JL, Aviles-Sanchez OF (2015) Knee functional state classification using surface electromyographic and goniometric signals by means of artificial neural networks. *Ing Univ* 1:51–66
39. Ertugrul OF, Kaya Y, Tekin R (2016) A novel approach for SEMG signal classification with adaptive local binary patterns. *Med Biol Eng Comput* 54:1137–46

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Hardware Implementation for Lower Limb Surface EMG Measurement and Analysis Using Explainable AI for Activity Recognition

Ankit Vijayvargiya^{ID}, Senior Member, IEEE, Puneet Singh^{ID}, Rajesh Kumar^{ID}, Senior Member, IEEE,
and Nilanjan Dey^{ID}, Senior Member, IEEE

Abstract—Electromyography (EMG) signals are gaining popularity for several biomedical applications, including pattern recognition, disease detection, human-machine interfaces, medical image processing, and robotic limb or exoskeleton fabrication. In this study, a two-channel data acquisition system for measuring EMG signals is proposed for human lower limb activity recognition. Five leg activities have been accomplished to measure EMG signals from two lower limb muscles to validate the developed hardware. Five subjects (three males and two females) were chosen to acquire EMG signals during these activities. The raw EMG signal was first denoised using a hybrid of Wavelet Decomposition with Ensemble Empirical Mode Decomposition (WD-EEMD) approach to classify the recorded EMG dataset. Then, eight time-domain (TD) features were extracted using the overlapping windowing technique. An investigation into the comparative effectiveness of several classifiers is presented, although it was hard to distinguish how the classifiers predicted the activities. Having a trustworthy explanation for the outcomes of these classifiers would be quite beneficial overall. An approach known as explainable artificial intelligence (XAI) was introduced to produce trustworthy predictive modeling results and applied the XAI technique known as local interpretable model-agnostic explanations (LIME) to a straightforward human interpretation. LIME investigates how extracted features are anticipated and which features are most responsible for each action. The accuracy of the extra tree classifier gives the highest accuracy of the other studied algorithms for identifying different human lower limb activities from sEMG signals.

Index Terms—Electromyography (EMG) signal acquisition system, explainable artificial intelligence (XAI), local interpretable model-agnostic explanations (LIME), lower limb activity recognition, machine learning (ML), signal processing.

Manuscript received 18 May 2022; revised 12 July 2022; accepted 27 July 2022. Date of publication 11 August 2022; date of current version 31 August 2022. This work was supported by the Visvesvaraya Ph.D. Scheme, Meity, Government of India, under Grant MEITY-PHD-2942. The Associate Editor coordinating the review process was Dr. Octavian Adrian Postolache. (*Corresponding author: Ankit Vijayvargiya*)

Ankit Vijayvargiya is with the Department of Electrical Engineering, Malaviya National Institute of Technology, Jaipur 302017, India, and also with the Department of Electrical Engineering, Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur 302017, India (e-mail: ankitvijayvargiya29@gmail.com).

Puneet Singh is with the Department of Electronics and Communication Engineering, Malaviya National Institute of Technology, Jaipur 302017, India (e-mail: 2019uec1355@mnit.ac.in).

Rajesh Kumar is with the Department of Electrical Engineering, Malaviya National Institute of Technology, Jaipur 302017, India (e-mail: rkumar.ee@mnit.ac.in).

Nilanjan Dey is with the Department of Computer Science and Engineering, JIS University, Kolkata 700109, India (e-mail: neelanjan.dey@gmail.com).

Digital Object Identifier 10.1109/TIM.2022.3198443

I. INTRODUCTION

IN THE past decade, research on analyzing human activity through sensor devices has garnered considerable attention. Human activity recognition (HAR) assesses a person's actions based on data that have been perceived and performed in any context required. HAR applications can be found in a variety of fields, such as intelligent homes with aided surveillance, healthcare monitoring, and tele-immersion applications [1], [2]. Typically, the HAR technique is accomplished by several stages, beginning with the analysis of recorded data, moving on to the use of various sensors, and eventually identifying the activity to be done. An HAR system can monitor the activity of individuals' lower limbs using a variety of sensing sensors. This type of HAR system can be beneficial in a variety of situations, including identifying jumping and lifting legs while performing ballet movements, identifying movements such as standing, sitting, walking, running, climbing and descending stairs, vacuuming, and situps, diagnosing neuromusculoskeletal disorders that cause knee pain, composing a robotic prosthetic limb in the case of an amputee's missing limb, and recognizing sports activities [3], [4].

Two methods are available for monitoring the activity of the human lower limb. The first method uses a visual sensor, whereas the second method uses wearable sensors [5]. Wearable sensors gather data while the subject is wearing them, whereas visual sensors such as stereo cameras do not require the subject to wear them. While vision-based activity identification systems have limited pervasiveness, privacy, and complexity in activity recognition, they are ideal for security applications due to their low cost. Significant technological advancements in wearable sensors have occurred in recent years. It leads to a reduction in the overall cost, making it more affordable. Popular wearable sensors include accelerometers, gyroscopes, Electromyography (EMG), and barometers [6], [7]. The EMG sensors outperform the others because of their ability to predict movement in a very short time. Two distinct approaches were used to record the EMG signal: surface EMG (noninvasive), and intramuscular EMG (invasive). The iEMG technique requires entering the muscle with a needle, which is a painful procedure that can turn dangerous if the infection spreads throughout the body. In contrast, the sEMG sensor is put on the external area of the body (i.e., on the surface) and makes signal acquisition simple [8].

TABLE I
COST (APPROXIMATE) COMPARISON WITH OTHER AVAILABLE COMMERCIAL DEVICES [11], [12]

Sr. No.	Product	Price in Commercial Market
1	Delsys Trigno	20,000 USD
2	Noraxon Ultium EMG	20,000 USD
3	Biometrics DataLITE sEMG	17,000 USD
4	Cometa	15,800 USD
5	Nexus	2,900 USD
6	Proposed	100 USD

Assistive devices based on sEMG signals can be used to improve the quality of an unhealthy person's daily life. Orthoses are used to assist people with weak joints to move more readily, whereas prostheses are used to replace missing limbs. The electric-powered robotic devices comprise an instrument for signal measurement, a machine learning (ML) algorithm for recognizing EMG patterns, and a mechanical framework for control. Although several signal conditioning systems and high-level EMG signal acquisition devices are commercially available, such as BIOPAC and BIONOMADX, they are costly. Pradhan *et al.* [9] proposed a two-channel EMG biopotential amplifier by deploying an INA128 instrumentation amplifier to capture EMG signals while performing seven hand gestures. Guo *et al.* [10] developed a four-channel EMG acquisition system by smoothing the original signal using an instrumentation amplifier (INA326) and operational amplifiers (AD8603). However, the system is complicated by the fact that each channel requires its own ADS8603 and INA326 for conditioning. Pancholi and Joshi [11] demonstrated the use of EMG data for signal identification by designing an eight-channel, low-cost data acquisition system. As a result of the complexity of these devices, this study focused on a two-channel sEMG data acquisition system for lower limb activity recognition that is wireless, cost-efficient, and compact in design. This system's prototype cost is around \$100, which includes mainly the MyoWare sensor and the ESP32 development board. Table I shows a cost comparison with other commercial EMG data acquisition devices.

Despite numerous successful applications, identifying leg activity with sEMG continues to be a challenge due to the noisy nature of the signals. Due to the prevalence of several artifacts such as ambient noise, inherent noise, and motion artifacts being mixed together, it is hard to distinguish between the muscle's actual sEMG signal and other artifacts. The use of novel approaches such as wavelet denoising, Wigner–Ville distribution, and empirical mode decomposition (EMD) to efficiently minimize these artifacts in the sEMG signal has recently been demonstrated by several studies. Vijayvargiya *et al.* [13] used a hybrid approach combining wavelet and ensemble EMD (EEMD) to denoise the EMG signal before feature extraction and classification in this study.

Several attempts have been undertaken during the past decade to detect activities in the legs using sEMG signals from various muscles of the lower limb [14]. Li *et al.* [15] worked on the recognition of nine different types of activities, making use of sEMG signal data recovered from forearm

TABLE II
OVERVIEW OF SEMG DATASET

Activities	Subjects	Sampling Frequency	Instrument	No. of Sensors
Running, standing, lunge stretching, jumping, walking [20]	9	2000 Hz	Delsys Trigno wireless EMG equipment	6
Walking, downstairs, upstairs, crossing obstacles and standing [21]	5	2000 Hz	Biometrics wireless sEMG sensor system	3
Walking, running, jumping, sit-to-stand [22]	5	1000 Hz	Noraxon Telemyo 2400T & 2400R	8
Sitting, standing, zero incline walking, stair upward/downwards, stairs, ramp ascent/descent [23]	10	1000 Hz	Delsys DE-2.1 sEMG Sensor	7
Walking, sitting, standing, stair upward/downwards [24]	10	1000 Hz	Biopac BN-EMG2s	9
Downstairs, upstairs, downhill, and uphill [25]	5	1000 Hz	Noraxon MyoResearch XP	9
Walking, sitting, standing [26]	22	1000 Hz	Biometrics	4

muscle movement. This uses a support vector machine and general regression neural network algorithms, which give 98.64% and 96.27% recognition rates, respectively. This is carried out to have accuracy and control of the intelligent prosthesis. Khemraj *et al.* [16] categorized the various lower limb activities using a convolutional neural network, K-nearest neighbor classifier, and support vector machine, among other methods. A significant number of healthy individuals were invited to participate in a range of tasks to collect data from the accelerometer and gyroscope sensors in this experiment. Gautam *et al.* [17] developed a unique classification approach that takes into account lower limb motions while calculating the prognosis of the knee joint angle. Naik *et al.* [18] used independent component analysis (ICA)-entropy bound minimization (EBM) to classify three distinct lower limb motions for healthy and knee deformity patients by decomposing the sEMG signal of activity using ICA by approximating the entropies of a wide range of distributions via EBM. In another study [19], deep belief networks (DBNs) are used to overcome local minima difficulties and overfitting concerns to accomplish the categorization of limb motions based on EMG signals. Table II indicates some of the previous studies on human lower limb activity recognition.

The ML systems have recently demonstrated superhuman performance in a variety of applications, leading to widespread industrial use of AI. This boost in performance is generally accomplished by increasing the model complexity, turning such systems into “black boxes” and raising ambiguity about how they work and ultimately make judgments. The ML systems have struggled to be embraced in delicate yet vital fields such as healthcare, where their benefits may be great. As a result, contemporary scientific interest in explainable artificial intelligence (XAI), a topic focused on developing new techniques to explain and comprehend ML models, has risen significantly. In this study, the XAI approach, known as local interpretable model-agnostic explanations (LIME), to achieve a simple human interpretation is used. LIME investigates how extracted features are anticipated and which features are most responsible for each action. The major contributions of this study are as follows.

- 1) A low-cost dual-channel data acquisition system with a python-based GUI is developed for sEMG signal measurement of lower limb muscles during five distinct activities.

- 2) Eight time-domain (TD) handcrafted features based on Wavelet Decomposition with Ensemble Empirical Mode Decomposition (WD-EEMD) are extracted using an overlapping windowing approach from sEMG data.
- 3) A quantitative assessment of the performance of the classification algorithms is evaluated and XAI is introduced to produce reliable predictive modeling results.

This work is structured as follows: Section II describes the hardware implementation of a two-channel sEMG data acquisition system. Section III outlines the techniques for pre-processing the sEMG signal in this investigation. Section IV contains a brief description of the evaluated ML models, followed by an explanation of the locally interpretable approach, LIME, for explaining significant features in Section V. The results and discussion are included in Section VI. Section VII discusses the conclusion and future scope.

II. PROPOSED METHODOLOGY

The current study is the hardware implementation for lower limb sEMG signal measurement for the five distinct lower limb activity recognitions. A total of five healthy subjects were considered to evaluate the performance indices as in [27] of the various classification algorithms. Finally, LIME is used to investigate the anticipation of features. Algorithm 1 presents the pseudo-code of the proposed methodology.

Algorithm 1 Proposed Methodology

- Step 1:** The development of hardware for the acquisition of sEMG signal from two lower limb muscles (X_1 and X_2);
 - Step 2:** Applied hybrid of wavelet decomposition with ensemble empirical mode decomposition (WD-EEMD) for minimizing the noise of sEMG signal;
 - Step 3:** Segmentation of sEMG signal using overlapping windowing approach with 256 ms window size and 25% overlap;
 - Step 4:** Five TD handcrafted features have been extracted from each segment of the sEMG signal;
 - Step 5:** Four ML algorithms such as decision tree, bagging, gradient boosting, and extra trees are applied for classification of five distinct lower limb activities (jump, situps, stairs down, stairs up, and walk);
 - Step 6:** Local interpretable model-agnostic explanations (LIME) technique has been used to explore how extracted features are anticipated or which features are more responsible for each action.
-

Sections II-A – II-E include the major part of the proposed methodology.

A. Hardware Implementation and Data Acquisition

The constructed module consists of an ESP32-Wroom-32 module linked to the outputs of two MyoWare muscle sensors (two-channel system) through the ESP32-Wroom-32 module, as shown in Fig. 1. Here, the system makes use of an inexpensive MyoWare Muscle Sensor from Advancer Technologies [28]. It can collect EMG signals and output both raw and

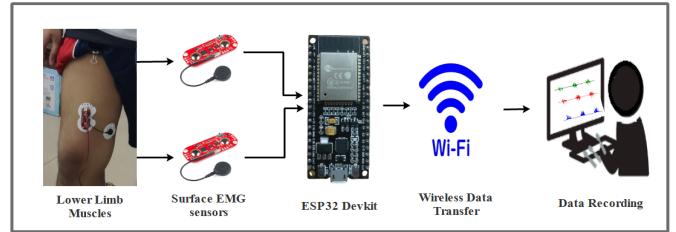


Fig. 1. Block diagram of the developed hardware.

preprocessed signals (EMG signal envelope), i.e., rectified and smoothed output. The sensor has three connection points to the body. Each measuring channel consists of three electrodes, two of which measure biopotentials, and the third serves as a relative ground. This MyoWare sensor circuit comprise a difference circuit, an offset removal circuit, a rectification circuit, and an integration circuit. The difference circuit is made up of an operational amplifier that subtracts the potential at one electrode from the potential at the other electrode to get the EMG signal. With this approach, the noise signal (e.g., power line interference) that is common to both the electrode inputs can be eliminated. A voltage offset is introduced into the EMG signal during its origination, leading the signal to be centered on a voltage value that is not zero. After the difference circuit, the rectification circuit is considered, which is an appropriate processing step for determining the total strength of the neural drive to the muscle and is correlated with the strength of muscular contraction and its output. It also removes the voltage offset from the rectified EMG signal to give an adjusted EMG signal. After the rectification circuit, an integration circuit is used to integrate the rectified EMG signal to provide an EMG output signal. This sensor incorporates a gain potentiometer for signal amplification and is supplied through a 3.3-V supply. Electrodes are snapped to the sensor module, and then the sensor is placed on the muscle of interest. These muscle sensors are intended to be used directly with a microcontroller and require no additional hardware. As a result, the sensor's output signal is not only a raw EMG signal but also it is amplified, rectified, and integrated.

An ESP32-Wroom-32 microcontroller is used to interface the sensors. It offers an in-built 12-bit ADC (sensors are connected to it) and throughput of around 1 Mbps (minimum) is observed till 15 m. It is powered by a 1850-mAh Li-ion battery charged via the USB connector, while the ESP32 module's 3.3-V input powers the MyoWare Muscle sensors. On-board Wi-Fi support is present and is used for data transmission. Sampling frequency of 1 kSps is programmed along with the communication protocol for uploading signals to the receiver station with the help of Arduino IDE. Wi-Fi credentials to connect with the Wi-Fi access point on which the receiver is present are provided to ESP32. It is assumed that there are no obstacles/walls between ESP32 and the receiver during the entire data acquisition procedure. A GET request is made via Python script on the receiver's end. The received JavaScript Object Notation (JSON) data are decoded into signals and are stored in an Excel file. To calculate



Fig. 2. Placement of EMG electrodes on the right leg muscle. (a) VM. (b) GS.

TABLE III
SUBJECTS' GENDER, AGE, AND ANTHROPOMETRIC DATA

ID	Age	Gender	Height	Weight
M1	34	Male	165cm	75Kg
M2	26	Male	170Cm	70Kg
M3	26	Male	170cm	64Kg
F1	25	Female	167.5cm	62Kg
F2	24	Female	158cm	47Kg

throughput requirements, the size of transmitted data per second via system is considered. The JSON encoded data take 78.24 kb which is being transmitted per second. This concludes that the required throughput of the system should be greater than 78.24 kbps. The only constraint here is that as the obstacles between the system and receiver increase, throughput decreases and it is observed that in the presence of obstacles, the throughput reduces to 100 kbps minimum within a 15 m range. Further increase in distance can cause signal loss but a 15-m range is enough for functioning.

To collect sEMG data, a total of five healthy participants (three males and two females) over the age of 18 years were recruited. Their relevant anthropometric measurements can be found in Table III. There was no previous history of knee pain or injury. The data were collected while performing one of five different activities: jumping, situps, stairs down, stairs up, and walking. Here, a 20-ft walk on a flat area is considered, along with a 7.75-in rise and 10-in run considering 15 steps, seven situps, and a 20 jumps. Each exercise was repeated ten times by each individual. For the evaluation, the sEMG signal from each subject for 8 s (approx.) during each activity is taken into account. The hardware that was developed has a sampling rate of 1 kHz, so there are 8000 samples (approx.) of the sEMG signal collected in 8 s. So each subject has approximately 40 000 samples (8000×5). As indicated in Fig. 2, sEMG electrodes were put on the following two muscles of the subject's right limb: vastus medialis (VM) and gastrocnemius (GS). The sEMG electrodes are positioned according to the surface electromyography for the noninvasive assessment of muscles (SENIAM) standard. To obtain valuable information from the EMG signal, the skin's impedance must be minimized, and the skin should be completely cleaned. The dead skin was removed using an abrasive gel. In addition,

moisture content can diminish the signal, which is why no sweat droplets should be present on the skin. The sampling frequency of the system is 1 kHz.

B. WD-EEMD-Based Signal Denoising

Because numerous noise signals and artifacts are combined, it is difficult to distinguish between the muscle's actual sEMG signal and other artifacts. These noises or artifacts may affect the handcrafted feature extraction and hence the EMG signal interpretation. As a result, signal denoising is necessary before using signals for any applications. In recent investigations, innovative approaches such as wavelet denoising, EMD, principal component analysis (PCA), and ICA have been successfully used to reduce noise in sEMG data.

The scheme to denoise the sEMG signal with the help of a wavelet denoising algorithm is demonstrated by Phinyomark *et al.* [29]. Among sEMG signals, several irregular noises, such as white Gaussian noise, are inconvenient to reduce via signal filtering. Hence, wavelet denoising is used to eliminate it. The expression for noisy signal $X(n)$ is stated as

$$X(n) = Y(n) + K(n) \quad (1)$$

wherein $Y(n)$ demonstrates the original signal, and $K(n)$ is the white Gaussian noise.

The wavelet denoising based on discrete wavelet transform (DWT) comprises the following steps:

- 1) signal decomposition is done by the DWT;
- 2) choose a threshold value for every wavelet decomposition;
- 3) to regenerate the denoised signal, the inverse wavelet transforms along with the threshold function are used.

As wavelet denoising over signals, discrete wavelet coefficients are formed when the signal passes through low-pass and high-pass filters. After signal decomposition and thresholding, wavelet denoising is used to obtain details and approximation coefficients. The decomposition level determines the overall number of coefficients. The traditional wavelet coefficient thresholding methods include soft and hard thresholding [30]. Soft thresholding uses a continuous divergence from the original signal, whereas hard thresholding uses merely partial information. This work implements universal thresholding across detailed coefficients. The form of universal thresholding selection is represented as

$$\lambda = \sigma \sqrt{2 \ln(M)}. \quad (2)$$

Herein; $\sigma = (\text{MAD})/0.6745$ MAD is designated as the median absolute deviation, and M is the length of the signal. The db7 wavelet from the Daubechies family is used up to the fourth level of signal decomposition, with Garotte thresholding applied to the detail coefficients.

As a consequence of the nonlinear and nonstationary nature of sEMG signals, decomposition techniques based on linear and stable processes may produce misleading findings. EEMD is a highly effective technique for decomposing nonstationary, nonlinear signals into full or nearly orthogonal intrinsic mode functions (IMFs). Functions or oscillation modes with a single

TABLE IV
MATHEMATICAL EXPRESSIONS OF EXTRACTED HANDCRAFTED FEATURES

Feature Extracted	Mathematical Formulation
MAV	$\frac{1}{N} \sum_{k=1}^N x_k $
RMS	$\sqrt{\frac{1}{N} \sum_{k=1}^N x_k^2}$
AAC	$\frac{1}{N} \sum_{k=1}^{N-1} x_{k+1} - x_k $
VAR	$\frac{1}{N-1} \sum_{k=1}^N x_k^2$
SSC	$\sum_{k=2}^N g(x_k)$
DASDV	$\sqrt{\frac{1}{N-1} \sum_{k=1}^{N-1} (x_{k+1} - x_k)^2}$
Kurt	$\frac{E(x-\mu)^4}{\sigma^4}$
Skew	$\frac{E(x-\mu)^3}{\sigma^3}$

component have a single instantaneous frequency. A signal can be iteratively decomposed into IMFs using the EMD approach. Due to the susceptibility of the EMD algorithm to noise, appearance mode mixing happens often. IMFs were defined in the EEMD approach for noise-assisted data processing as the ensemble average of trials.

In this study, like in our previous study [13], a preprocessing approach combining WD-EEMD is used. The white Gaussian noise and undesired signals, such as noise produced by other muscles, are first filtered out using wavelet denoising. The signal is then decomposed using EEMD to eliminate power line interference and baseline wandering noise.

C. Segmentation

Segmentation is used in preprocessing because it is an efficient technique to handle time-series data needed in a deep-learning-based model. The windowing method is used to implement segmentation. Segmentation may be accomplished through the use of two different types of windowing techniques: adjacent windowing and overlapping windowing [31]. The classification accuracy and response time rely on the signal window size. This study used the same overlapping windowing approach as other studies [4], [18].

D. Feature Extraction

There are three approaches to extracting features: TD, frequency domain (FD), and time–frequency domain (TFD). TD handcrafted features are used in signal classification applications such as muscular activity identification, onset detection, and muscle contraction. These characteristics are commonly used because they improve signal classification in low noise and low computing complexity situations [32]. Therefore, in this research, where classification of lower limb movement will be conducted, eight TD features will be exploited [13]. The mathematical expressions of eight extracted handcrafted features are presented in Table IV.

Here, x denotes the sEMG signal, N denotes the total number of samples, k denotes the sample instance, σ denotes the standard deviation of the sample, μ denotes the mean of the sample, and E is the signal's expected value estimator.

E. Classification Techniques

After extracting the information in the form of handcrafted features, ML approaches may be used to identify the data as pertaining to distinct lower limb actions. While there are numerous ML algorithms for identification, decision trees and their ensembles, such as bagging, gradient boosting, and extra trees, have been evaluated in this section.

1) *Decision Tree Classifier*: As the name indicates, a decision tree classifies the data points described by a collection of attributes. These are shown as trees, with each node representing an attribute and each child representing a possibility. Each leaf corresponds to a decision rule. Several common decision tree training algorithms are Iterative Dichotomiser 3 (ID3), classification and regression tree (CART), C4.5, and multivariate adaptive regression splines (MARS) [4], [33]. Here, the most straightforward learning algorithm, ID3, uses greedy techniques to select the best attribute on each iteration locally. The ID3 classifier follows the following steps.

- 1) *Step 1*: First, estimate the entropy of each feature in the dataset.
- 2) *Step 2*: Split the dataset into subgroups using the property that provides the highest information gain.
- 3) *Step 3*: If the value of entropy is 0, the node is considered to be a leaf node, and no splitting is needed. However, if the entropy value is greater than 0, the signal should be split as specified in step 2.

2) *Bagging Classifier*: This is a method for combining the results of many ML algorithms to generate an accurate prediction model [34]. This classifier creates random selections from the training dataset using bootstrap. It generates n decision tree and individually estimates their predictions, and then aggregates them to produce a final prediction using the mean or vote of the acquired individual findings. It is also known as the bootstrap algorithm because it is capable of reducing substantial amounts of variation by introducing randomness into the process of assembling and subsequently generating an ensemble.

3) *Gradient Boosting Classifier*: In this classifier [35], a weak learning algorithm is applied repeatedly. In contrast, the results from the previous ones are improved successively, and then the classification is built up like in a robust classifier. The decision tree is typically used as a weak learner in a situation. With the aid of the gradient descent method, the performance is frequently monitored at each iteration using the loss function and optimized as a result of the process.

4) *Extra Trees Classifier*: This approach is also known as the extremely randomized decision tree since it is based on the decision tree algorithm [36]. This classifier generates a huge number of classification trees rather than a single classification tree. Each tree is trained using the whole training set, and random splits are used rather than computing the locally optimal splits for each feature, resulting in a very efficient, quick, and accurate approach.

F. XAI for Predictive Model

Artificial intelligence (AI) was regarded a theoretical study with few real-world applications for years. ML has made huge

TABLE V
PERFORMANCE PARAMETERS OF THE STUDIED ML MODELS (IN %)

Classifier / Subjects	Decision Tree				Bagging				Gradient Boosting				Extra Trees			
	Acc	Spec	Sens	F-score	Acc	Spec	Sens	F-score	Acc	Spec	Sens	F-score	Acc	Spec	Sens	F-score
M1	93.42	98.36	93.50	93.47	96.71	99.18	96.76	96.71	98.36	99.59	98.35	98.35	99.18	99.79	99.17	99.18
M2	90.35	97.61	89.89	89.93	95.50	98.88	95.29	95.29	96.46	99.13	96.34	96.33	98.07	99.52	97.97	97.97
M3	91.05	97.78	90.66	90.63	92.46	98.12	92.07	92.09	95.13	98.80	94.92	94.87	97.49	99.36	97.31	97.40
F1	86.87	96.69	86.37	86.51	92.12	97.99	92.26	92.22	92.50	98.14	92.77	92.38	95.12	98.81	95.06	94.80
F2	83.82	95.95	83.43	83.40	91.18	97.80	91.44	91.09	92.48	98.11	92.50	92.45	94.28	98.56	94.36	94.25
Mean	89.10	97.28	88.77	88.78	93.59	98.40	93.57	93.48	94.99	98.75	94.98	94.88	96.83	99.21	96.77	96.72

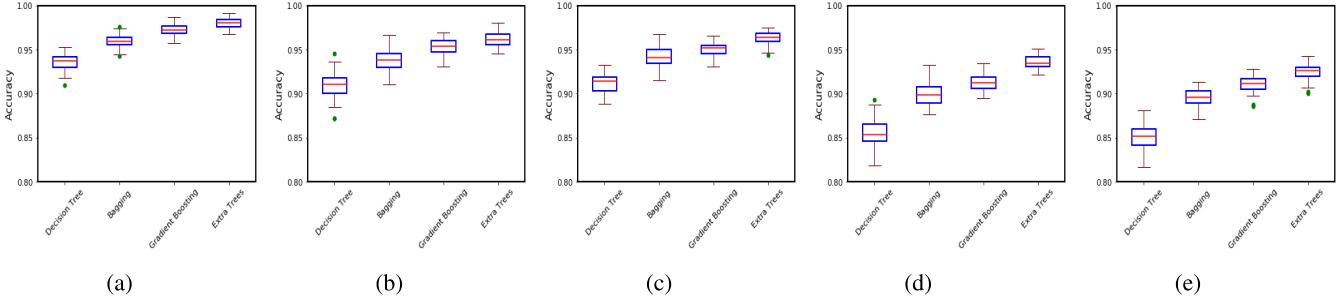


Fig. 3. Box plot for all the studied classifiers with 50 randomized tests. (a) Male 1. (b) Male 2. (c) Male 3. (d) Female 1. (e) Female 2.

advancements in the past decade, leading to its widespread industrial use, due to more powerful processors, stronger learning algorithms, and easier access to big amounts of data. The ML models address real-time difficulties across several sectors. Increasing model complexity improves prediction performance. Deep learning helps machines recognize, analyze, and extract complicated data representations for detection and classification. This hierarchy of increasing complexity and utilization of enormous amounts of data to train and develop advanced systems boosts their predictive capacity, but lowers their ability to explain their underlying workings and methods. Therefore, it is harder to understand their activities and interpret their future recommendations.

It is difficult to trust systems whose outcomes are difficult to explain, especially in fields such as healthcare or self-driving vehicles, where moral and justice questions have inevitably emerged. The development of efficient, fair, resilient, and high-performing models for real-world applications has resurrected the subject of XAI. It is gaining popularity at the moment owing to its simplicity of understanding and easy-to-interpret procedures [37], [38]. XAI is a set of tools and frameworks for understanding and interpreting ML model predictions, as well as analyzing and optimizing model performance and assisting others in comprehending your models' interpretability. The interpretability of a model may be evaluated on two levels.

- 1) *Global Interpretation:* It offers a comprehensive description of the operation of AI systems, for example, by outlining the rules of features that govern their outcomes.
- 2) *Local Interpretation:* In XAI, local interpretability does not attempt to describe the complete AI model. It instead trains XAI models to approximate individual prediction. Local interpretability models show how the model's outcomes change as the values of certain attributes change within a given interval.

Local explanation approaches such as LIME and SHAP are two of the most frequently used model-agnostic, local explanation strategies for explaining any given black-box classifier. LIME is an innovative method that uses a locally interpretable process to describe the prediction process of any classification model and gives information about the classifier's prediction strategy and feature correlations. It is based on the idea that any complicated model is linear on a local scale. LIME attempts to fit a basic model around a single observation to simulate how the global model works at that location. The basic model can then be used to explain the more complicated model's predictions locally.

The LIME model should be model-agnostic, to learn the behavior of the underlying model, obfuscate the input, and see how the predictions change. To discover the behavior of the underlying model, disrupt the input and examine how the predictions change using the LIME model, which is model-agnostic. Interpretability benefits from this since input may be perturbed by things that humans comprehend (e.g., words or parts of a picture), even if the model uses significantly more intricate components as features (e.g., a graph) (e.g., word embeddings). It makes use of the following ways to do this:

- 1) permute the sample;
- 2) the distance between permutations and the original data is calculated;
- 3) using a complex model, make predictions on new data;
- 4) the best features of the complex model results from permuted data are chosen;
- 5) fit a simple model to permuted data using features chosen in the previous step and the similarity score as a weight;
- 6) the simplest model's feature weights explain the complex model's local behavior.

The mathematical expression of LIME is presented as

$$\text{explain}(y) = \underset{g \in G}{\text{argmin}} L(h, g, \lambda_y) + P(g). \quad (3)$$

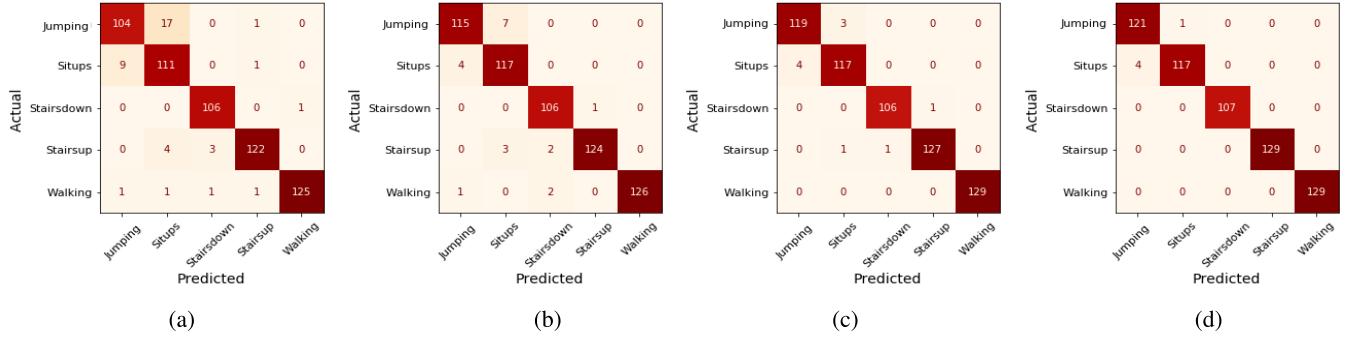


Fig. 4. Confusion matrix for subject M1 with studied classifier. (a) Decision tree. (b) Bagging. (c) Gradient boosting. (d) Extra trees.

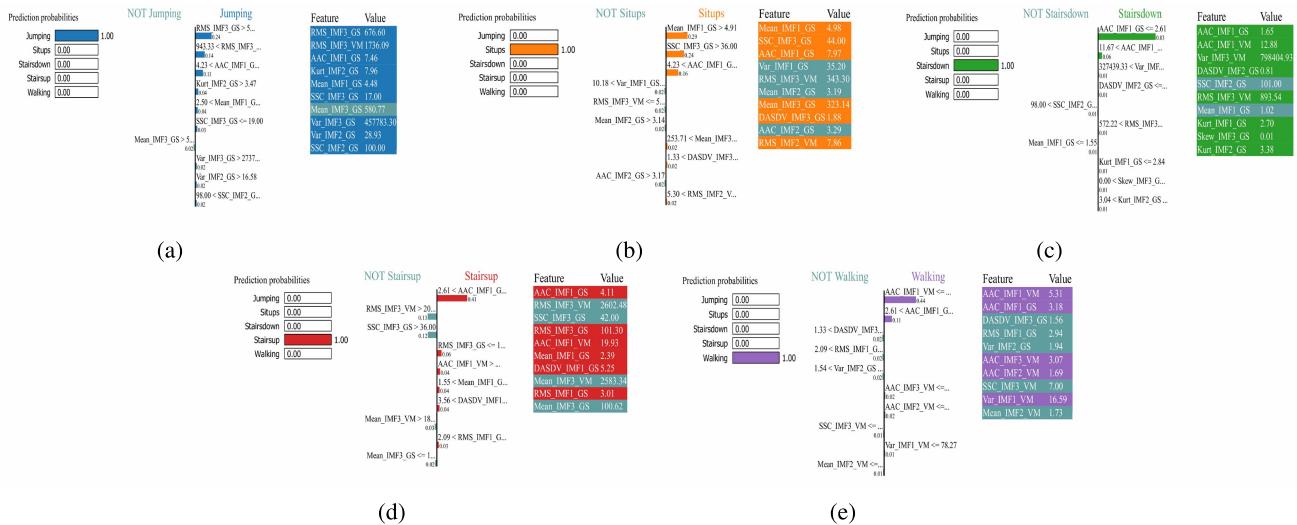


Fig. 5. LIME outcomes for extra tree classifier for distinct lower limb activities of subject M1. (a) Jumping. (b) Situps. (c) Stairs down. (d) Stairs up. (e) Walking.

Here, $\text{explain}(y)$ represents the explanation model for instance y ; $P(g)$ represents the complexity of model; $L(h, g, \lambda_y)$ represents the loss function; h represents the black-box model; g represents the explainer; y is the similarity between data points.

III. RESULTS AND DISCUSSIONS

The problem in this application is a multiclass classification task, and the dataset was partitioned into 70% training dataset and 30% testing sets, since the training set should be more than that of the test dataset to get a higher classification rate. A classification system using ML produces four possible outcomes: true negative (TN), true positive (TP), false negative (FN), and false positive (FP). Here, TP and TN are the correct predictions of the positive and negative classes while the FP and FN are the wrong predictions, respectively. The classification performance indicators for each classifier were calculated based on the number of occurrences of each of the following four potential outcomes:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (4)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (5)$$

$$\text{Recall(Sensitivity)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (6)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (7)$$

$$F1\text{-score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}. \quad (8)$$

Although accuracy may be considered one of the evaluation criteria, additional evaluation metrics such as sensitivity, selectivity, and $F1$ -score are necessary and covered in this study. Table V summarizes the performance metrics for the five computational classifiers evaluated in this study for recognizing five distinct lower limb activities. Across all the subjects, the extra tree classifier has the best accuracy and $F1$ -score value, while the decision tree classifier has the lowest accuracy and $F1$ -score value. The decision tree, bagging, and gradient boosting tree classifiers have a mean accuracy of 89.10%, 93.59%, and 94.99%, respectively, while the extra tree classifier has a mean accuracy of 96.83%. The decision tree, bagging, gradient boosting tree, and extra tree classifiers had mean F -scores of 88.78%, 93.48%, 94.88%, and 96.72%, respectively.

To determine statistical validity, an experiment was done with 50 randomized tests to determine the accuracy of all the

TABLE VI
CLASS-WISE ACCURACY OF EXTRA TREE CLASSIFIER (IN %)

ID	Jumping	Situps	Stairs down	Stairs up	Walking
M1	99.18	96.69	100	100	100
M2	97.32	100	96.52	96	100
M3	99.29	97.45	91.67	98.13	100
F1	97	97.93	96.47	97.91	86
F2	97.76	99.22	91.66	94.57	88.59
Mean	98.11	98.26	95.26	97.32	94.92

studied classifiers and create a box plot to demonstrate the accuracy distributions as shown in Fig. 3. Here, the mean accuracy of the extra tree classifier is higher than that of the other examined classifiers, and the standard deviation is also minimum in all the five subjects for the 50 randomized samples.

The confusion matrix derived by the studied classifier for subject 1 is shown in Fig. 4. The confusion matrix is a table that allows you to visualize the performance of a classification system. It comprises data on the actual and expected labels that a model evaluates. From Fig. 4(a), it can be seen that 104 samples of jogging class, 111 samples of situps class, 106 samples of stairs down class, 122 samples of stairs up class, and 125 samples of walking class are correctly identified, while 18 samples of jogging class, ten samples of situps class, one sample of stairs down class, seven samples of stairs up class and four samples of walking class are wrongly predicted. Similarly, Fig. 4(b)–(d) shows the confusion matrix of the remaining studied ML classifiers.

Table VI shows the class-wise accuracy of extra tree classifier for each subject individually for all the five activities. The average accuracy of activities such as jogging, situps, ladder, ladder, and walking is 98.11%, 98.26%, 95.26%, 97.32%, and 94.92%, respectively.

A total of eight TD handcrafted features are derived from sEMG signals of two leg muscles using three decomposed signals (IMF1, IMF2, and IMF3) obtained using WD-EEMD. Thus, the classification algorithm uses a total of 48 features (two muscles × three IMF × eight handcrafted features = 48 features) to classify the five distinct lower limb actions. However, it is hard to know the classifiers' predictions of the actions. Having a credible explanation for the results of these classifiers would be quite valuable in the long run. As a result, LIME is used to enhance the interpretability of the classification process. Fig. 5 shows the outcomes of the LIME approach for extra tree classifier for distinct lower limb activities of subject M1. The results with the top ten features are presented for five samples of the different activities. In Fig. 5(a), the extra tree classifier predicts the instance with jumping to be a positive response with a chance of 100%. Only the mean value of IMF3 of the GS muscle supports the prediction to be negative, while the rms value of IMF3 of both the muscles is the top two features that support it to be positive. Fig. 5(b)–(e) shows similar findings for situps, stairs down, stairs up, and walking, respectively.

IV. CONCLUSION

The research provides a wireless and portable two-channel data acquisition module for sEMG measurement using a MyoWare Muscle Sensor for the automated identification of lower limb actions. Five subjects (three males and two females) were chosen to acquire EMG signals during the five distinct activities: jumping, situps, stairs down, stairs up, and walking. The WD-EEMD preprocessing technique was first used to denoise the sEMG signal, and then an overlapping windowing approach was used to extract eight TD hand-crafted features from each signal of muscles. A comparative performance analysis of the decision tree and its ensembles is presented. However, it was difficult to determine how the classifiers predicted the actions. As a result, the XAI technique known as LIME is used, which explores how extracted features are anticipated or which features are most responsible for each action. LIME has some limitations, such as label and data shift, explanations that depend on hyperparameter selection, and even the same instances may have various perceptions. In the future, highly interactively XAI models such as SHapley Additive exPlanations (SHAP) and local interpretation-driven abstract Bayesian networks can be used.

There are still some expansions available for the intended work in the future. To begin, the data were collected with a rather small sample. As a future scope, researchers can collect data from more subjects and validate their results and also analyze the performances for intersubject activity classification. In addition, other modern ML algorithms may be used, and feature reduction techniques can be used to minimize the extracted features.

REFERENCES

- [1] S. Ranasinghe, F. Al Machot, and H. C. Mayr, "A review on applications of activity recognition systems with regard to performance and evaluation," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 8, pp. 1–22, Aug. 2016.
- [2] A. Vijayvargiya, P. L. Singh, S. M. Verma, R. Kumar, and S. Bansal, "Performance comparison analysis of different classifier for early detection of knee osteoarthritis," in *Sensors for Health Monitoring*. Academic, Aug. 2019, pp. 243–257.
- [3] R. Gopura, K. Kiguchi, G. Mann, and D. Torricelli, "Robotic prosthetic limbs," *J. Robot.*, vol. 2018, no. 8, pp. 1–2, 2018.
- [4] A. Vijayvargiya, C. Prakash, R. Kumar, S. Bansal, and J. M. R. S. Tavares, "Human knee abnormality detection from imbalanced sEMG data," *Biomed. Signal Process. Control*, vol. 66, Apr. 2021, Art. no. 102406.
- [5] Z. Hussain, M. Sheng, and W. Emma Zhang, "Different approaches for human activity recognition: A survey," 2019, *arXiv:1906.05074*.
- [6] S. Xia, L. Chu, L. Pei, Z. Zhang, W. Yu, and R. C. Qiu, "Learning disentangled representation for mixed-reality human activity recognition with a single IMU sensor," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 2514314.
- [7] Z. Chen, C. Jiang, S. Xiang, J. Ding, M. Wu, and X. Li, "Smartphone sensor-based human activity recognition using feature fusion and maximum full *a posteriori*," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 3992–4001, Jul. 2020.
- [8] A. Vijayvargiya, B. Singh, R. Kumar, and J. M. R. S. Tavares, "Human lower limb activity recognition techniques, databases, challenges and its applications using sEMG signal: An overview," *Biomed. Eng. Lett.*, vol. 188, no. 2, pp. 1–16, Jun. 2022.
- [9] A. Pradhan *et al.*, "Acquisition and classification of EMG using a dual-channel EMG biopotential amplifier for controlling assistive devices," in *Proc. IEEE Annu. India Conf. (INDICON)*, Dec. 2016, pp. 1–5.
- [10] W. Guo, X. Sheng, H. Liu, and X. Zhu, "Development of a multi-channel compact-size wireless hybrid SEMG/NIRS sensor system for prosthetic manipulation," *IEEE Sensors J.*, vol. 16, no. 2, pp. 447–456, Jan. 2016.

- [11] S. Pancholi and A. M. Joshi, "Portable EMG data acquisition module for upper limb prosthesis application," *IEEE Sensors J.*, vol. 18, no. 8, pp. 3436–3443, Apr. 2018.
- [12] M. Ergeneci, K. Gokcesu, E. Ertan, and P. Kosmas, "An embedded, eight channel, noise canceling, wireless, wearable sEMG data acquisition system with adaptive muscle contraction detection," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 68–79, Feb. 2018.
- [13] A. Vijayvargiya, V. Gupta, R. Kumar, N. Dey, and J. M. R. S. Tavares, "A hybrid WD-EEMD sEMG feature extraction technique for lower limb activity recognition," *IEEE Sensors J.*, vol. 21, no. 18, pp. 20431–20439, Sep. 2021.
- [14] W. Gao, L. Zhang, W. Huang, F. Min, J. He, and A. Song, "Deep neural networks for sensor-based human activity recognition using selective kernel convolution," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.
- [15] J. Li *et al.*, "Using body sensor network to measure the effect of rehabilitation therapy on improvement of lower limb motor function in children with spastic diplegia," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 11, pp. 9215–9227, Nov. 2020.
- [16] P. K. Shukla, A. Vijayvargiya, and R. Kumar, "Human activity recognition using accelerometer and gyroscope data from smartphones," in *Proc. Int. Conf. Emerg. Trends Commun., Control Comput. (ICONC)*, Feb. 2020, pp. 1–6.
- [17] A. Gautam, M. Panwar, D. Biswas, and A. Acharyya, "MyoNet: A transfer-learning-based LRCN for lower limb movement recognition and knee joint angle prediction for remote monitoring of rehabilitation progress from sEMG," *IEEE J. Transl. Eng. Health Med.*, vol. 8, pp. 1–10, 2020.
- [18] G. R. Naik *et al.*, "An ICA-EBM-based sEMG classifier for recognizing lower limb movements in individuals with and without knee pathology," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 3, pp. 675–686, Jan. 2018.
- [19] J. Zhang, C. Ling, and S. Li, "EMG signals based human action recognition via deep belief networks," *IFAC-PapersOnLine*, vol. 52, no. 19, pp. 271–276, 2019.
- [20] Y. Luan, Y. Shi, W. Wu, Z. Liu, H. Chang, and J. Cheng, "HAR-sEMG: A dataset for human activity recognition on lower-limb sEMG," *Knowl. Inf. Syst.*, vol. 63, no. 10, pp. 2791–2814, Oct. 2021.
- [21] X. Shi, P. Qin, J. Zhu, M. Zhai, and W. Shi, "Feature extraction and classification of lower limb motion based on sEMG signals," *IEEE Access*, vol. 8, pp. 132882–132892, 2020.
- [22] S. Park, D. Lee, W. K. Chung, and K. Kim, "Hierarchical motion segmentation through sEMG for continuous lower limb motions," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4402–4409, Oct. 2019.
- [23] B. Hu, E. Rouse, and L. Hargrove, "Benchmark datasets for bilateral lower-limb neuromechanical signals from wearable sensors during unassisted locomotion in able-bodied individuals," *Frontiers Robot. AI*, vol. 5, p. 14, Feb. 2018.
- [24] J. Ryu, B.-H. Lee, and D.-H. Kim, "sEMG signal-based lower limb human motion detection using a top and slope feature extraction algorithm," *IEEE Signal Process. Lett.*, vol. 24, no. 7, pp. 929–932, Jul. 2017.
- [25] Y. Yu, L. Fan, S. Kuang, L. Sun, and F. Zhang, "The research of sEMG movement pattern classification based on multiple fused wavelet function," in *Proc. IEEE Int. Conf. Cyber Technol. Autom., Control, Intell. Syst. (CYBER)*, Jun. 2015, pp. 487–491.
- [26] M. Lichman, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, Tech. Rep., 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [27] A. Vijayvargiya, R. Kumar, and N. Dey, "Voting-based 1D CNN model for human lower limb activity recognition using sEMG signal," *Phys. Eng. Sci. Med.*, vol. 44, no. 4, pp. 1297–1309, Nov. 2021.
- [28] Myoware Muscle Sensor SEN 13723—SparkFun Electronics. Accessed: Jan. 7, 2022. [Online]. Available: <https://www.sparkfun.com/products/13723>
- [29] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Wavelet-based denoising algorithm for robust EMG pattern recognition," *Fluctuation Noise Lett.*, vol. 10, no. 2, pp. 157–167, Jun. 2011.
- [30] A. Graps, "An introduction to wavelets," *IEEE Comput. Sci. Eng.*, vol. 2, no. 2, pp. 50–61, Jun. 1995.
- [31] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, Apr. 2014.
- [32] W. M. B. W. Daud, A. B. Yahya, C. S. Horng, M. F. Sulaima, and R. Sudirman, "Features extraction of electromyography signals in time domain on biceps brachii muscle," *Int. J. Model. Optim.*, vol. 3, no. 6, pp. 3–7, Dec. 2013.
- [33] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.
- [34] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
- [35] L. Mason, J. Baxter, P. L. Bartlett, and M. R. Frean, "Boosting algorithms as gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 512–518.
- [36] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Apr. 2006.
- [37] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?" Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1135–1144.
- [38] M. S. Kamal, A. Northcote, L. Chowdhury, N. Dey, R. G. Crespo, and E. Herrera-Viedma, "Alzheimer's patient analysis using image and gene expression data and explainable-AI to present associated genes," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 2513107.

Ankit Vijayvargiya (Senior Member, IEEE) received the B.Tech. degree (Hons.) in electrical engineering from Rajasthan Technical University (RTU), Kota, India, in 2010, the M.Tech. degree in instrumentation from Devi Ahilya University (DAVV), Indore, India, in 2014, and the Ph.D. degree in electrical engineering from the Malaviya National Institute of Technology (MNIT), Jaipur, India, in 2022.

His research interests include biomedical signal processing, pattern recognition, and machine learning.

Puneet Singh is currently pursuing the B.Tech. degree in electronics and communication engineering with the Malaviya National Institute of Technology (MNIT), Jaipur, Rajasthan, India.

His research interests include embedded systems and machine learning.

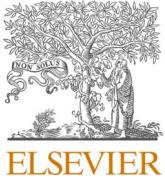
Rajesh Kumar (Senior Member, IEEE) received the B.Tech. degree (Hons.) in electrical engineering from the National Institute of Technology (NIT), Kurukshetra, India, in 1994, the M.E. degree (Hons.) in power systems from the Malaviya National Institute of Technology (MNIT), Jaipur, Rajasthan, India, in 1997, and the Ph.D. degree in intelligent systems from the University of Rajasthan, Jaipur, in 2005.

His research interests include theory and practice of intelligent systems, computational intelligence and analytics, electric transportation, smart grid, and robotics.

Niljanan Dey (Senior Member, IEEE) received the B.Tech. and M.Tech. degrees in information technology from the West Bengal University of Technology, Kolkata, India, in 2005 and 2011, respectively, and the Ph.D. degree in digital image processing from Jadavpur University, Kolkata, in 2015.

His research topics are signal processing, machine learning, and information security.

D. Dey is an Associate Editor of *IET Image Processing*, the Editor-in-Chief of the *International Journal of Ambient Computing and Intelligence* (IJACI), and a Series Co-editor of *Springer Tracts in Nature-Inspired Computing* (STNIC).



Human knee abnormality detection from imbalanced sEMG data

Ankit Vijayvargiya ^{a,*}, Chandra Prakash ^b, Rajesh Kumar ^a, Sanjeev Bansal ^c, João Manuel R. S. Tavares ^d

^a Department of Electrical Engineering, Malaviya National Institute of Technology, Jaipur, India

^b Department of Computer Science and Engineering, NIT, Delhi, India

^c Department of Physical Medicine and Rehabilitation, Sawai Man Singh Hospital, Jaipur, India

^d Instituto de Ciéncia e Inovação em Engenharia Mecânica e Engenharia Industrial, Departamento de Engenharia Mecânica, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal



ARTICLE INFO

Keywords:

Surface electromyography
Wavelet denoising
Oversampling techniques
Imbalanced data
Machine learning

ABSTRACT

The classification of imbalanced datasets, especially in medicine, is a major problem in data mining. Such a problem is evident in analyzing normal and abnormal subjects about knee from data collected during walking. In this work, surface electromyography (sEMG) data were collected during walking from the lower limb of 22 individuals (11 with and 11 without knee abnormality). Subjects with a knee abnormality take longer to complete the walking task than healthy subjects. Therefore, the SEMG signal length of unhealthy subjects is longer than that of healthy subjects, resulting in a problem of imbalance in the collected sEMG signal data. Thus, the development of a classification model for such datasets is challenging due to the bias towards the majority class in the data. The collected sEMG signals are challenging due to the contribution of multiple motor units at a time and their dependency on neuromuscular activity, physiological and anatomical properties of the involved muscles. Hence, automated analysis of such sEMG signals is an arduous task. A multi-step classification scheme is proposed in this research to overcome this limitation. The wavelet denoising (WD) scheme is used to denoise the collected sEMG signals, followed by the extraction of eleven time-domain features. The oversampling techniques are then used to balance the data under analysis by increasing the training minority class. The competency of the proposed scheme was assessed using various computational classifiers with 10 fold cross-validation. It was found that the oversampling techniques improve the performance of all studied classifiers when applied to the studied imbalanced sEMG data.

1. Introduction

Knee pain is a common complaint that affects the autonomy of individuals of all ages. As per the survey [1], one in every three people have arthritis or joint symptoms in the 18–64 age group due to an injury or underlying condition such as knee osteoarthritis. The knee joint is a synovial joint that acts as a shock absorber and provides stability to the body. The knees join the femur (thigh bone) and come to the tibia (shin bone). The patella (knee cap) and fibula are the other bones that form the knee joint. The bones of the knee are attached to the muscles along with the tendon. The articular cartilage is a thin cartilage between the femur and tibia and provides a smooth movement to these bones [2].

Neuromusculoskeletal disorders such as cerebral palsy and osteoarthritis are other infirmities that reduce the quality of life of people [3,4].

Clinically, such diseases are diagnosed using X-Ray [5] or Magnetic Resonance Imaging (MRI) techniques [6]. X-ray technique is primary used for gathering and evaluation of the bone status while MRI provides the detailed information of knee structure such as cartilage, ligaments and tendons. MRI is an efficient way for diagnosing but at a higher cost. Knee abnormalities can be diagnosed through wearable sensors such as electromyography (EMG), accelerometer and gyrometer, or by visual sensors such as imaging cameras. A comprehensive review of available gait approach and analysis of applications based on gait data was discussed by chandra prakash and others [7]. EMG sensors allow the recognition of movements in advance [8] while providing faster detection of signal variations [9], and are therefore superior in the investigation of neuromusculoskeletal disorders.

The signal collected by EMG sensors is a biomedical signal that

* Corresponding author.

E-mail addresses: ankityvijay@skit.ac.in (A. Vijayvargiya), cprakash@nitdelhi.ac.in (C. Prakash), rkumar.ee@mnit.ac.in (R. Kumar), sanjivbansal1899@gmail.com (S. Bansal), tavares@fe.up.pt (J.M. R.S. Tavares).

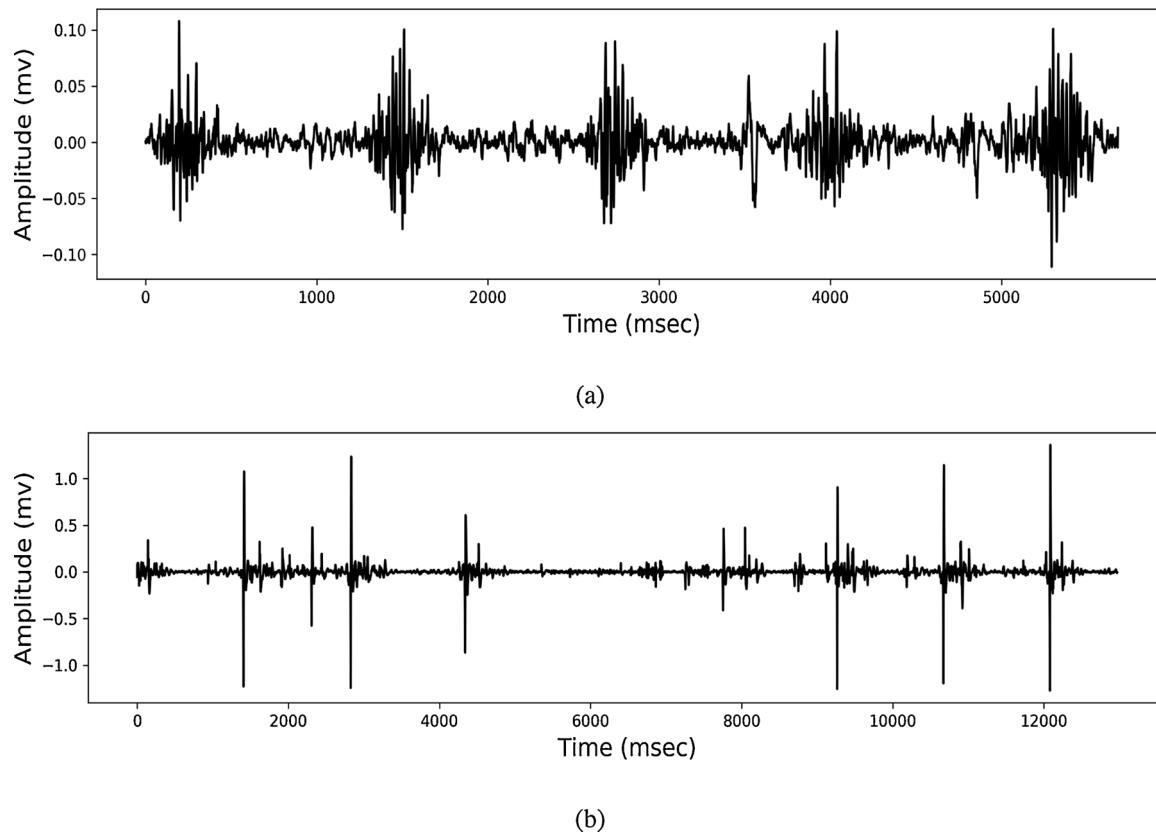


Fig. 1. sEMG signals acquired during gait from: (a) a healthy subject and (b) an abnormal knee subject.

quantifies the electrical activity produced by skeletal muscles. Surface electromyography (sEMG) and intramuscular EMG (iEMG) techniques are popular approaches used to acquire EMG signals [10]. sEMG presents an advantage over iEMG, as the electrodes can be applied without pain or medical monitoring, and additionally the possibility of infection is negligible. Long-term control is often simpler with surface electrodes than using iEMG needles. The placement of the sEMG sensors has a significant influence on the acquired signal and their consequent analysis and recognition by the computational algorithm as confirmed in [11]. sEMG sensors can collect data during daily human activities such as sitting, standing, walking and climbing, which can be used to detect anomalies, actions recognition, etc [12].

sEMG signals are being employed in different applications such as automatic control of lower and upper limb prosthesis or exoskeletons [13], diagnosis of neuromuscular disorders [14] and exercise, fitness monitoring [15]. Khimraj et al. classified six movements of lower limb and compared the results of the computational classifiers [16]. Hudgins et al. proposed a pattern recognition based approach with time-domain features and a multilayer perceptron neural network for the classification of four types of limb motions [17] using sEMG signals. Huang et al. estimated from sEMG signals the human arm joint torque using a back-propagation neural network (BPNN) and auto encoders [18]. Silva et al. studied the spinal cord injury using the sEMG signal collected during activities of the upper limb [19]. Sudarsan et al. designed and developed an artificial limb controlled by sEMG signals [20]. Tuncer et al. proposed the ternary pattern and discrete wavelet based feature extraction for hand movement recognition from sEMG signal for amputee people [21]. The Support Vector Machine (SVM) based classification of different upper limb movements performed by five healthy subjects was investigated by Cai et al. [22]. A linear time series based prediction models were proposed by chandra prakash and others. These models can be used for efficient control of robotic assistive devices for lower limb for a smooth movement [23].

In the last decade, researchers have focused on the classification of sEMG signals from the upper limb. The sEMG signals acquired from the lower limb are more challenging as it got influence from overlapping muscles [24,25]. The classification of sEMG data from the lower limb has been studied using different signal processing techniques [26,27]. The human lower limb flexion/extension joint angles were estimated from sEMG signal using a deep belief network by Chen et al. [28]. Bonato et al. investigated the fatigue of quadriceps and hamstring muscles based on sEMG signal [29]. Swaroop et al. used sEMG signal for myopathy and neuropathy classification using a neural network based approach [30]. Kugler et al. used a SVM based solution for the recognition of Parkinson's disease from sEMG data [31]. Morbidoni et al. classified the gait phase while walking from sEMG data with the help of a deep learning based approach [32]. Characterization of walking patterns and using computational techniques are classified into normal and gait pathology [33].

Class imbalance is a challenging issue in medical data. Class imbalance occurs in the case where a varying quantity, i.e., different number, of total samples is present in different data classes. So, if imbalanced data appears in the classification, the result will be swayed toward the majority class [23]. Therefore, for improving the diagnosis success, it is required to balance the data either by increasing the minority class (oversampling) or reducing the majority class (under sampling). Rajesh et al. classified five groups of heartbeat with imbalanced ECG beat using the AdaBoost ensemble classifier [34]. Past studies show that oversampling techniques can overcome the class imbalance problem [35,36]. Taft et al. applied the Synthetic Minority Oversampling Technique (SMOTE) to enhance the performance of a classification model for the identification of adverse drug events in women admitted for labor and delivery [36].

During walking, the imbalance problem of the acquired sEMG data may occur because of two reasons:

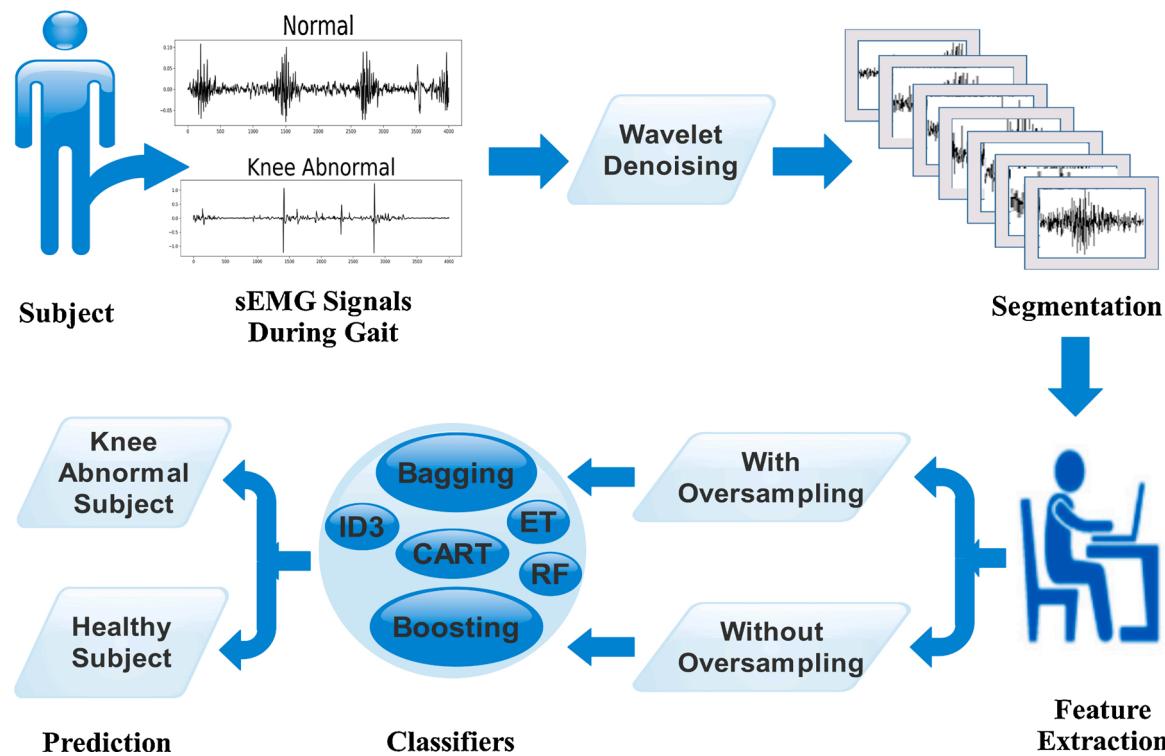


Fig. 2. Block diagram of the proposed methodology for abnormality knee detection from sEMG signal collected during walking.

1. The number of abnormal subjects is inferior to the number of healthy subjects;
2. the length of the data collected from abnormal subjects is different from the one obtained from normal subjects.

A person with knee abnormality takes longer to complete the movement task resulting in larger signal length in contrast to a healthy (or control) subject. Hence, the large length of the sEMG signal of the abnormal subject leads to a class imbalance problem. In the present study, two classes with the same number of subjects have been taken; however, the signal length of each class is different. Fig. 1(a) and (b) exhibit the sEMG data acquired from one normal subject and one abnormal subject respectively while performing the same gait task, respectively. It can be observed that the lengths of the acquired sEMG signals have varying lengths.

As per the author's knowledge, there is no literature/study that resolves the problem of identifying knee abnormality from imbalanced sEMG gait data. Therefore, this paper presents for the first time the problem of identifying a knee abnormality from class imbalanced sEMG data and evaluated the performance parameters of various machine learning classifiers when applied to a balanced and imbalanced sEMG data. The data considered here include (1) original imbalanced data and (2) balanced data that have been obtained through the application of oversampling techniques on the original data. The major contributions of this work are:

1. Identification of knee abnormality from imbalanced sEMG dataset, where sEMG signal length differ between normal and abnormal subjects.
2. Evaluation of metrics (MSE, MAE, SNR and PSNR) for selecting optimal mother wavelet and decomposition level of DWT wavelet;
3. Extraction of eleven discrete wavelet transform (DWT) based features by splitting the sEMG signal into various frequency bands.
4. Evaluating the impact of oversampling techniques on the performance indices of classification models.

This article is organized as follows. A description of the dataset data that has been utilized in the presented work is given in Section 2. Section 3 presents the proposed methodology. Results and their discussion are presented in Sections 4 and 5, respectively. Under Section 6, conclusions and perspectives of future work have been elaborated.

2. sEMG dataset

The sEMG signal data used in this study has been publicly made available to the UCI machine learning repository by Sanchez et al [37]. The data consist of sEMG signals of the lower limbs of 22 subjects over 18 years of age, where 11 subjects are healthy and 11 subjects have known knee abnormalities. The healthy subjects do not have previous record of knee injury or pain. One abnormal knee subject suffered from sciatic nerve injury, six abnormal subjects had anterior cruciate ligament (ACL) injury and the remaining four suffered from a meniscus injury. The data was collected using a DataLog MWX8 from Biometrics Ltd. and a goniometer when the subjects were performing one of three different tasks: walking, flexion of the leg up and leg extension from sitting position. The sEMG data were recorded for the four muscles: biceps femoris (BF), vastus medialis (VM), rectus femoris (RF) and semitendinosus(ST), with the goniometer attached to the external side of the knee joint. The affected limb of the subject with abnormal knee and the left leg of the healthy subjects were chosen for acquiring the sEMG signal. The data was acquired according to a sampling frequency of 1000 Hz and 14-bit of resolution. The sEMG signals have already been filtered using a band pass filter with a pass band frequency of 20 Hz to 460 Hz. The recorded data does not contain any signal corresponding to the transition states, i.e., standing to sitting, sitting to walking, walking to standing, etc. The data was transferred directly from the MWX8 device to the computer by bluetooth in real-time. Only the sEMG signals acquired during the walking task were used the experiments performed for this study.

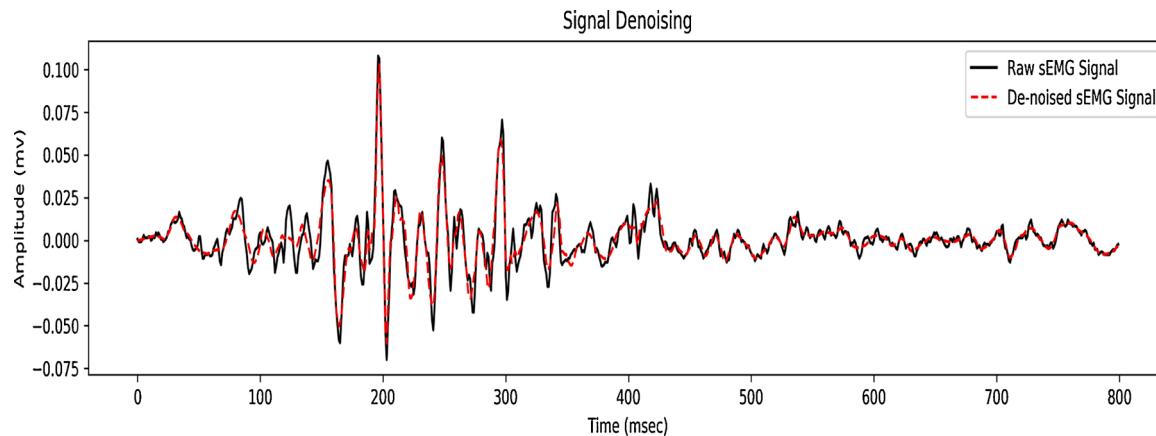


Fig. 3. A raw sEMG Signal and the corresponding denoised sEMG signal obtained by wavelet decomposition.

3. Proposed methodology

This section presents the methodology proposed for the identification of knee abnormality from imbalanced sEMG signal data collected during walking. Fig. 2 presents the general flow of the proposed methodology.

Feature extraction from the collected sEMG signals requires signal pre-processing. The dataset is already preprocessed with band pass filter with a pass band frequency of 20 to 460 Hz. To remove the random noise discrete wavelet denoising technique has been used. The mother wavelet and level of decomposition of the wavelet denoising based on DWT were selected from the mean squared error (MSE), mean absolute error (MAE), signal to noise ratio (SNR) and peak signal to noise ratio (PSNR) values. Thereafter, eleven time-domain features were extracted by the use of an overlapping windowing technique with a window length of 256 ms and an overlapping of 25% and then the samples of the features were normalized in between 0 (zero) to 1 (one) by the use of min-max normalization method. Since the data length of healthy and unhealthy subjects are not the same, therefore the number of samples of the extracted features for both conditions, i.e., classes, are also not equal. Hence, in order to balance the samples of the used features, different over-sampling techniques have been used.

The main sections of the proposed methodology presented in Fig. 2 have been explained in the following subsection:

3.1. Wavelet denoising

Four types of noises are usually introduced while recording sEMG signals [38]: (1) Ambient noise generated by electromagnetic appliances, (2) Inherent noise resulting from electronic devices, (3) Motion artifacts produced due to the movement of the sEMG electrodes, and (4) Inherent noise instability due to the firing rate of the involved motor units. Therefore, signal denoising is an essential task that should be performed before using the signals for classification purposes. Conventional filtering techniques like High Pass, Low Pass or Band Pass, can be used to minimize the noise which is not within the range of the active sEMG signal spectrum band. The sEMG signals have been passed through the band pass filter (20 to 460 Hz). Recently, novel methods such as wavelet denoising, Independent Component Analysis (ICA) and Empirical Mode Decomposition (EMD), have been successfully used in recent studies to minimize noise from sEMG signals [39–41].

Over the past few years, the use of wavelet denoising has been proven successful on sEMG signals acquired from the upper and lower limbs. Phinyomark et al. presented the idea to denoise the sEMG signal using the wavelet denoising algorithm [42]. Random noises like White Gaussian noise in sEMG signals are difficult to minimize by signal filtering; however, it can be effectively removed with the help of wavelet

denoising. The white Gaussian noise $Y(n)$ can be expressed as:

$$Y(n) = X(n) + H(n), \quad (1)$$

where $X(n)$ is the original signal and $H(n)$ is the noise signal.

The steps of the discrete wavelet transform based wavelet denoising are:

- Decompose the signal $Y(n)$ by using the DWT;
- Select the threshold for each wavelet transform;
- Use inverse wavelet transform with threshold function to reconstruct the denoised signal.

When wavelet denoising is applied to signals, different wavelet coefficients are generated by passing the signal through High-Pass and Low-Pass filters. The approximation and detail coefficients are obtained after the decomposition of the signal by using the wavelet denoising technique. The number of coefficients depends on the level of decomposition. In this research, the wavelet denoising technique is used with *sym4* from the Symlet family to the first level of decomposition. After this, the thresholding can be performed. There are two traditional methods for thresholding the wavelet coefficient results: Hard and Soft thresholding [43]. Hard thresholding can keep only partial information of the original signal while soft thresholding has a constant deviation relative to the original signal. In this study, Universal thresholding is applied to the detailed coefficients. The universal thresholding selection rule [44] is defined as:

$$\lambda = \sigma \sqrt{2 \ln(N)}, \quad (2)$$

where $\sigma = (\text{MAD})/0.6745$, with MAD referring to the Median Absolute Deviation of the wavelet coefficient and N is the signal length.

As an example, Fig. 3 shows a raw sEMG signal and the corresponding denoised signal after applying the wavelet denoising technique.

3.2. Segmentation

The nature of a sEMG signal is stochastic, so segmented sEMG signal is more appropriate rather than the full sEMG signal. Different length of sEMG data affects the accuracy of the classification model. Two types of windowing techniques, called Adjacent and Overlapping, are usually used for sEMG signal segmentation [45]. Here, the overlapping windowing technique with a 256 ms time windowing and 25% of overlapping is used [46].

3.3. Feature extraction

In a raw sEMG signal, it has been seen that various noises and artifacts are present, which degrades its analysis accuracy because the required information remains as an amalgam in the raw sEMG signal. Therefore, to enhance the classification accuracy, the sEMG signal is denoised first, and then, features are extracted to be used as the input of a computational classifier. Three types of feature extraction techniques are available in the literature: time-domain (TD), frequency domain (FD) and time-frequency domain (TFD) feature. In this study, eleven time-domain features are used for knee abnormality identification from sEMG signals collected during walking [38].

- **Mean Absolute Value (MAV)** It is the average of N absolute values of a time series sample of sEMG signal (x_i) within a given time interval:

$$\text{MAV} = \frac{1}{N} \sum_{i=1}^N |x_i|. \quad (3)$$

- **Root Mean Square (RMS)**: It is calculated by taking the square root of the arithmetic mean of the squared sample amplitude:

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N |x_i|^2}. \quad (4)$$

- **Zero Crossing (ZC)**: It gives information about how many times the signal crosses the zero amplitude level:

$$\text{ZC} = \sum_{i=1}^{N-1} f(x_i \cdot x_{i+1}), \quad (5)$$

where:

$$f(x) = \begin{cases} 1, & \text{if, } x < 0, \\ 0, & \text{otherwise.} \end{cases}$$

- **Slope Sign Change (SSC)**: Similar to the zero crossing, SSC also gives information of frequency in terms of time domain features. Thus, it indicates how many times positive to negative or negative to positive slope transitions have taken place:

$$\text{SSC} = \sum_{i=2}^{N-1} f(x_i), \quad (6)$$

where:

$$f(x) = \begin{cases} 1, & \text{if, } (x_i > x_{i-1} \text{ and } x_i > x_{i+1}) \text{ or } (x_i < x_{i-1} \text{ and } x_i < x_{i+1}), \\ 0, & \text{otherwise.} \end{cases}$$

- **Variance (VAR)**: It gives information about the deviation of the signal from its mean value:

$$\text{VAR} = \frac{1}{N-1} \sum_{i=1}^N x_i^2. \quad (7)$$

- **Wilson Amplitude (WAMP)**: It indicates the amount of stages resulting from amplitude change between two adjoining segments that exceeds a pre-defined threshold in the sEMG signal:

$$\text{WAMP} = \sum_{i=1}^{N-1} f(|(x_{i+1} - x_i)|), \quad (8)$$

where:

$$f(x) = \begin{cases} 1, & \text{if, } (x \geq \text{Threshold}), \\ 0, & \text{otherwise.} \end{cases}$$

- **Myopulse Percentage Rate (MYOP)**: It is the average value of absolute values of the sEMG signal exceeding a pre-defined threshold:

$$\text{MYOP} = \frac{1}{N} \sum_{i=1}^N f(x_i), \quad (9)$$

where:

$$f(x) = \begin{cases} 1, & \text{if, } (x \geq \text{Threshold}), \\ 0, & \text{otherwise.} \end{cases}$$

- **Difference Absolute Standard Deviation Value (DASDV)**: It is the standard deviation value of the wavelength:

$$\text{DASDV} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} (x_{i+1} - x_i)^2}. \quad (10)$$

- **Average Amplitude Change (AAC)**: It is the average cumulative length of the sEMG signal over the time segment:

$$\text{AAC} = \frac{1}{N} \sum_{i=1}^{N-1} |x_{i+1} - x_i| \quad (11)$$

- **Skewness (Skew)**: It is a measure related to the lack of symmetry or the asymmetry of the signal data:

$$\text{Skew} = \frac{E[(x - \mu)^3]}{\sigma^3}, \quad (12)$$

where σ is the standard deviation of the data, μ is the mean of the data, and E is the expected value estimator of the signal.

- **Kurtosis (Kurt)**: It determines if the signal has a peak or is rather flat at its mean point:

$$\text{Kurt} = \frac{E[(x - \mu)^4]}{\sigma^4}. \quad (13)$$

Here, eleven time domain features for four different muscles, thus, 44 features were extracted. The length of the signal has been observed to be different for the two classes under classification; therefore, the number of samples of extracted features are different for the abnormal and healthy knee subjects.

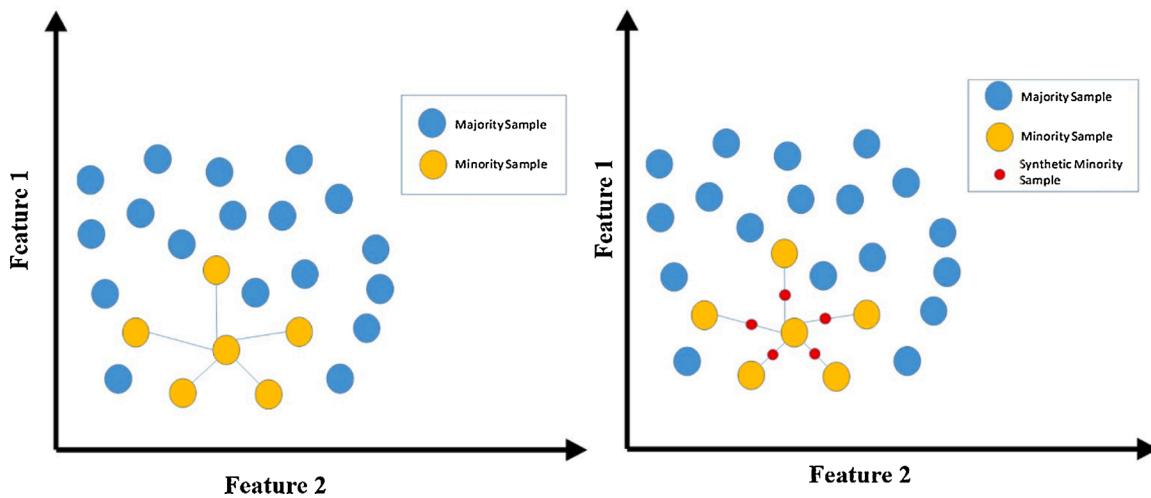


Fig. 4. Illustration of the SMOTE oversampling approach.

3.4. Normalization

Normalization or Min-Max scaling is a re-scaling technique that shifts the range of the features to scale the range in between 0 (zero) to 1 (one) according to:

$$Y_{\text{new}} = \frac{Y_{\text{Fold}} - Y_{\text{Foldmin}}}{Y_{\text{Foldmax}} - Y_{\text{Foldmin}}} \quad (14)$$

where Y_{new} is the normalized EMG feature, Y_{Fold} is the actual EMG feature, Y_{Foldmax} is the maximum value of the actual EMG feature, and Y_{Foldmin} is the minimum value of the actual EMG feature.

3.5. Oversampling techniques

As discussed previously, the length of the recorded sEMG data of abnormal knee subjects is different from the one of the normal subjects. An oversampling technique has then been used to balance the imbalanced class, i.e., to increase the number of samples of the minority class. On the one hand, if imbalanced data is used for training a classification model, then the data of majority categories would have dominated. On the other hand, if the data of minority classes are oversampled by duplicity, then an over fitting problem may occur due to repeated samples. Data level methods, algorithm level methods and hybrid methods are three approaches that have been used for handling the class imbalance problem.

Chawla et al. [47] proposed SMOTE for increasing the minority class. In the SMOTE approach, the minority class can be oversampled by creating synthetic cases in the feature space formed by the instance and its K -nearest neighbors as shown in Fig. 4. The steps of the SMOTE oversampling technique are:

1. Choose K nearest neighbor from the minority samples ($X_i \in S_{\text{min}}$) according to the requirement of oversampling.
2. Randomly select a neighbor X_j ($X_j \in S_{\text{min}}$) from the K nearest neighbor.
3. A new synthetic sample is generated according to:

$$X_{\text{new}} = X_i + |X_j - X_i| * \delta, \text{ where } \delta \in [0, 1] \text{ is the random variable.}$$

Fig. 5 shows the pair plots for three extracted features: Mean Absolute value, Zero Crossing and Kurtosis of the Rectus Femoris muscle for two cases, without oversampling and after the SMOTE oversampling. In these plots, blue circles indicate samples of healthy subjects and red circles indicate the samples of abnormal knee subjects. For all the three features, Fig. 5 (a) shows that the samples of abnormal knee subjects (red circles) are in higher number than those of the normal subjects (blue circles) and Fig. 5 (b) shows that the number of the samples of the

minority class (blue circles) have increased after the SMOTE oversampling.

Another oversampling approach, Adaptive Synthetic oversampling (ADAYSN) has been proposed by Haibo et al. [48]. According to the data distribution, new data points of minority class can be generated using ADAYSN. In this technique, one can shift the decision boundary to focus on those difficult to learn samples and also reduce the learning bias that is introduced by the original imbalanced dataset. SVM SMOTE is also a variant of SMOTE oversampling where the borderline is defined based on the SVM hyper plane methodology [49].

3.6. Computational classifiers

Once the features are extracted, and the oversampling is performed, computational techniques can be used to classify the data as from healthy subjects or abnormal knee subjects. In this study, Iterative Dichotomiser 3 (ID3), Classification and Regression Trees (CART), Bagging, Gradient Booster, Random Forest and Extra Tree based classifiers have been studied.

Iterative Dichotomiser 3 (ID3) is used to generate a decision tree [50]. It is a supervised learning algorithm for data classification that can be used for both categorical and numerical variables. It has no guarantee to give the optimal solution since it may converge upon local minima. Therefore, for splitting the input data, Greedy techniques are used to locally select the best attribute on each iteration. An over fitting problem may occur for the high depth of decision tree.

The steps of the Iterative Dichotomiser 3 classifier are:

- Step 1: Calculate the entropy of every attribute used in the data:

$$H(S) = - \sum_{y \in Y} p(y) \log_2 p(y), \quad (15)$$

where S is the data where the entropy is calculated, Y is the set of classes in S , and $p(y)$ is the proportion of the elements in class y to the number of elements in S .

- Step 2: Split set S into subsets using the attribute where the information gain is maximum:

$$\text{IG}(S, A) = H(S) - H(S|A). \quad (16)$$

- Step 3: If the entropy is 0 (zero), then it is a leaf node and no further splitting is performed; however, if entropy is higher than 0 (zero),

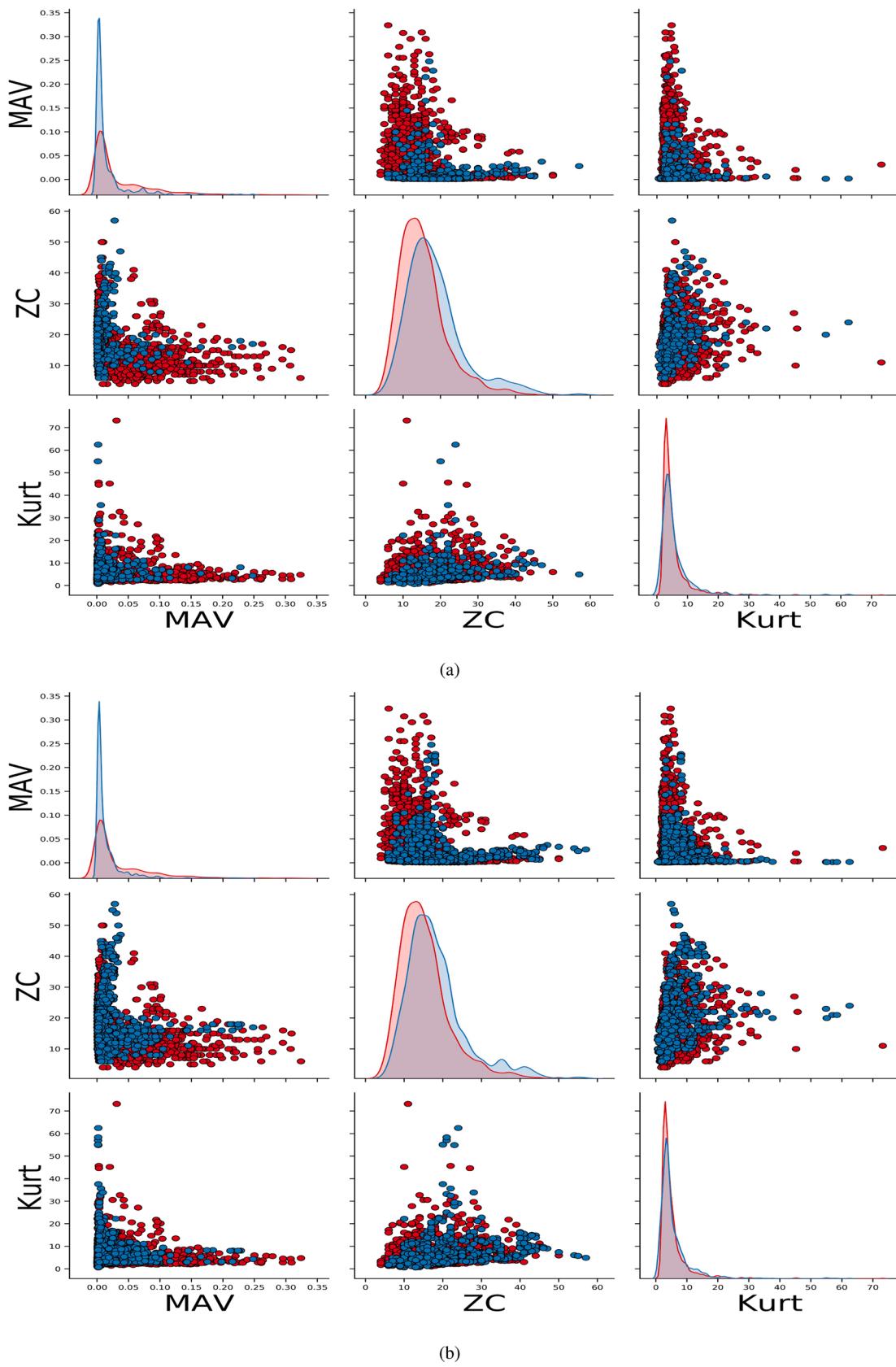


Fig. 5. Pair plots for three extracted features: (a) without oversampling and (b) after the SMOTE oversampling.

Table 1
Confusion Matrix.

		Predictive class	
		Positive class	Negative class
Actual class	Positive Class	True Positive (TP)	False Negative (FN)
	Negative Class	False Positive (FP)	True Negative (TN)

further splitting is needed and it should be performed as listed in step 2.

Classification and Regression Tree (CART) [51] is also a type of decision tree algorithm that can be used for classification and regression. It is similar to ID3 but the only difference is that ID3 uses information gain while CART uses the Gini index for splitting the data. Accordingly, the feature with a lower Gini index value is selected for splitting:

$$\text{Gini} = 1 - \sum_{i=1}^n (p_i)^2, \quad (17)$$

where p_i is the probability of a feature in a particular class.

Bagging Classifier [52] is an ensemble method that is used to build an accurate prediction model by combining the results of multiple machine learning algorithms. This classifier selects bootstrap (random) samples from the input data. It constructs n classification trees and calculates the prediction of individual classification trees, and then accumulates them to form the final prediction based on averaging or voting of the obtained individual outcomes. It is also called Bootstrap Algorithm as it can be used to reduce the high variance of a black-box estimator (e.g., a decision tree), by integrating randomization into its construction process and then making an ensemble out of it.

Gradient Boosting Classifier [53] is based on the idea of applying a weak learning algorithm repeatedly while improving the results from the previous ones sequentially and finally build the classification as in a robust classifier. The Decision Tree is usually used as a weak learner. The performance is commonly measured by using the loss function at each iteration and optimized with the help of the Gradient Descent algorithm.

Random Forest Classifier [54] is an extension of the Decision Tree algorithm where bootstrap techniques are integrated. This classifier develops many classification trees rather than a single tree. It creates several bootstrap samples from the sample data and then develops a decision tree for each of the bootstrap samples. Random features are selected, and the features which give the best split are considered for splitting the node. The average number of votes from different trees are calculated, which provides the output of the overall prediction.

Extra Trees Classifier [55] is also called Extremely Randomized Decision Tree. It is also based on the Decision Tree algorithm and involves an ensemble decision tree like in the Random Forest algorithm. It differs from the Random Forest algorithm in two ways: (i) the training of each tree uses the entire training set instead of a random subset and (ii) the random splits are chosen instead of computing the locally best splits for each feature which makes this algorithm highly efficient.

Table 2
Mean values of MSE, MAE, SNR and PSNR calculated from the sEMG signals of all 22 subjects using different mother wavelets (best values found are in bold).

Wavelet function	MSE	MAE	SNR	PSNR
db1	0.00057	0.01081	7.94073	27.61071
db4	0.00042	0.00971	8.55542	28.22540
db7	0.00046	0.00994	8.30930	27.97928
db9	0.00048	0.01005	8.16712	27.83709
sym2	0.00045	0.00998	8.43299	28.10296
sym4	0.00039	0.00941	8.82642	28.49640
sym 5	0.00040	0.00947	8.77723	28.44721
sym7	0.00040	0.00953	8.71923	28.38921
bior1.1	0.00057	0.01081	7.94073	27.61071

Support Vector Machine Classifier [56] is a supervised machine learning algorithm where each data sample is plotted in a n-dimensional space and a hyper plane is constructed to separate the different classes optimally. This optimal decision surface can be constructed by maximizing the margin width between the nearest members of the classes.

Multilayer Perceptron Neural Network [57] is also a supervised learning algorithm. It is an artificial neural network that consists of three or more layers in a feed forward architecture with two layers dedicated to input and output while others are hidden layers. MLP uses a nonlinear activation function in the neurons and each layer is fully connected to the next layer.

3.7. Performance Measures

In this work, the problem under study is a binary classification problem which means either the subjects have knee abnormalities or not. In binary classification, the classifier gives four possible outcomes, which are: True Positive (TP), i.e., correct positive prediction, True Negative (TN), i.e., correct negative prediction, False Positive (FP), i.e., incorrect positive prediction, and False Negative (FN), i.e., incorrect negative prediction. A confusion matrix may be built from these four outcomes, as illustrated in Table 1, and different performance measures may be calculated:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (18)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (19)$$

$$\text{Recall(Sensitivity)} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (20)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \quad (21)$$

$$F1 - \text{Score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}. \quad (22)$$

For balanced data, accuracy can be considered as one of the evaluation metrics. In the case of imbalanced data, there is a need for other evaluation metrics such as sensitivity, specificity and F-score, which have also been considered in this study [58].

4. Results

This section presents the results obtained using classifiers without and with oversampling techniques for the problem of knee abnormality detection from imbalance sEMG walking data. The obtained results seem to support the hypothesis that oversampling techniques improve the performance of the used classification model. The performance of the studied classifiers applied on original data (imbalanced data), and balanced data, obtained using oversampling techniques on the original data, was assessed using:

1. Selection of the optimal mother wavelet and decomposition level of the DWT denoising technique;
2. Performance analysis using the k-fold cross-validation technique;
3. Performance analysis according to a 95% confidence interval.

4.1. Selection of the optimal DWT denoising mother wavelet and level of decomposition

The selection of the mother wavelet and decomposition level in wavelet transform denoising plays an important role. In this study, haar (db1), Daubechies (db4, db7, db9), symlet (sym2, sym4, sym5, sym7) and biorthogonal (bior1.1) have been used from the wavelet transform

Table 3

Mean values of MSE, MAE, SNR and PSNR calculated from the sEMG signals of all 22 subjects using different levels of wavelet decomposition (best values found are in bold).

Decomposition level	MSE	MAE	SNR	PSNR
D1	0.00001	0.00142	24.31335	43.98333
D2	0.00005	0.00318	17.68627	37.35625
D3	0.00015	0.00592	12.42797	32.09795
D4	0.00039	0.00941	8.82642	28.49640
D5	0.00100	0.01368	6.65022	26.32019
D6	0.00157	0.01650	5.63505	25.30503
D7	0.00170	0.01703	5.42477	25.09475
D8	0.00171	0.01707	5.40338	25.07336
D9	0.00171	0.01706	5.40112	25.07110
D10	0.00172	0.01709	5.39753	25.06750

family with 1 to 10 levels of decomposition. here, the mother wavelet and decomposition level have been selected based on mean squared error (MSE), mean absolute error (MAE), signal to noise ratio (SNR), and peak signal to noise ratio (PSNR). These four metrics are commonly used to assess the performance of denoising methods. The MSE, MAE, SNR, and PSNR values were calculated from the sEMG signals of all 22 subjects using different levels of wavelet decomposition and then the mean value of these metrics was considered. A lower value of MSE and MAE and a higher value of SNR and PSNR indicate that the signal is accurately denoised. These four parameters were calculated for 9 mother wavelet functions with the fourth level of decomposition as shown in Table 2. It was also computed the same four metrics for the 1 to 10 levels of the wavelet decomposition with symlet4 mother wavelet as indicated in Table 3.

4.2. Performance analysis using k-fold cross validation

K-fold cross-validation is a method of re-sampling using constrained

data to validate the performance of machine learning models. In this approach, the input data is randomly split into k groups of equal size. Then, the model is trained using $k - 1$ groups of data and validated with the k^{th} group. Here, the performance of the studied classification models was assessed using 10-fold cross-validation [59] in four sEMG walking knee datasets:

1. The original data (imbalanced data);
2. The balanced data obtained by SMOTE oversampling;
3. The balanced data obtained by ADASYN oversampling;
4. The balanced data obtained by SVM SMOTE oversampling.

The obtained results are presented in Table 4.

4.3. Performance analysis according to a 95% confidence interval

For testing the statistical significance, an experiment was conducted for 100 randomized tests for measurement of accuracy, sensitivity, specificity and F-score performance metrics were calculated according to a 95% confidence interval. The obtained results are summarized in Table 5.

5. Discussion

Fig. 6 shows the number of samples used in training and testing the classifiers with and without oversampling. In the case of Without oversampling, the classification models were trained with 342 and 1533 samples of the extracted features from healthy and abnormal knee subjects, respectively, and tested with 38 and 170 samples. On the other hand, with oversampling, the number of the training samples of normal subjects was increased from 342 to 1533, and then the classification models were trained with 1533 samples of the extracted features from both healthy and knee abnormal subjects, and the testing was performed

Table 4

Performance obtained by each classifier when applied on the original data and on the oversampled data (best values found in bold.)

Oversampling	Classifier	Accuracy	Sensitivity	Specificity	F-Score				
Original	ID3	83.2	1.5	0.935	0.016	0.379	0.059	0.536	0.059
	CART	82.8	1.5	0.951	0.03	0.276	0.137	0.405	0.194
	Bagging	89.5	1.5	0.979	0.011	0.518	0.067	0.675	0.058
	Gradient Boosting	89.4	2.2	0.972	0.017	0.545	0.086	0.694	0.071
	Random Forest	90.9	1.1	0.981	0.01	0.587	0.047	0.733	0.037
	SVM	86.5	1.5	989	0.012	0.308	0.053	0.467	0.06
	MLP	84.9	2.2	0.965	0.021	0.329	0.141	0.471	0.181
	Extra Tree	91.9	1.1	0.984	0.008	0.629	0.065	0.765	0.048
SMOTE	ID3	76.2	3.8	0.787	0.56	0.647	0.123	0.700	0.07
	CART	75.6	4.3	0.771	0.061	0.687	0.109	0.718	0.052
	Bagging	89.2	1.8	0.937	0.016	0.689	0.067	0.793	0.045
	Gradient Boosting	87.5	1.9	0.893	0.022	0.792	0.066	0.838	0.036
	Random Forest	90.7	2	0.937	0.02	0.776	0.063	0.848	0.038
	SVM	81.4	2	0.812	0.024	0.824	0.065	0.816	0.03
	MLP	85.5	2.8	0.878	0.035	0.75	0.063	0.807	0.036
	Extra Tree	93.1	1.1	0.969	0.01	0.763	0.054	0.853	0.033
ADASYN	ID3	63.5	6.2	0.604	0.089	0.774	0.088	0.668	0.047
	CART	62.8	6.4	0.596	0.094	0.771	0.109	0.659	0.046
	Bagging	87.8	2.4	919	0.018	0.695	0.09	0.788	0.06
	Gradient Boosting	86.4	1.5	0.878	0.019	0.8	0.054	0.836	0.027
	Random Forest	90.3	2.4	0.93	0.023	0.782	0.063	0.847	0.03
	SVM	78.1	2.7	0.758	0.034	0.887	0.054	0.816	0.027
	MLP	83.2	3.1	0.835	0.033	0.821	0.085	0.825	0.047
	Extra Tree	92.2	1.8	0.96	0.017	0.753	0.067	0.842	0.043
SVM SMOTE	ID3	71.7	4.6	0.713	0.063	0.734	0.08	0.718	0.04
	CART	73	5.3	0.726	73	0.75	0.102	0.73	0.049
	Bagging	90.2	2.4	0.937	0.024	0.745	0.084	0.827	0.053
	Gradient Boosting	88.1	2.2	0.897	0.022	0.808	0.086	0.847	0.048
	Random Forest	91	2.4	0.941	0.021	0.774	0.083	0.847	0.053
	SVM	81.8	2.3	0.824	0.03	0.789	0.078	0.803	0.04
	MLP	85.3	2.2	0.873	0.027	0.766	0.078	0.813	0.044
	Extra Tree	93.2	1.8	0.97	0.014	0.761	0.07	0.851	0.044

Table 5

Performance obtained by each classifier when applied on the original data and on the balanced data obtained by oversampling according to a 95% confidence Interval (best values found in bold).

Oversampling	Classifier	Accuracy		Sensitivity		Specificity		F-Score	
		Lower	Upper	Lower	Upper	Lower	Upper	Lower	Upper
Original	ID3	82.93	83.04	0.9319	0.933	0.366	0.3732	0.5198	0.5276
	CART	82.78	82.92	0.9499	0.9522	0.2718	0.2855	0.3955	0.4141
	Bagging	89.54	89.68	0.9762	0.9773	0.5317	0.5379	0.6843	0.6896
	Gradient Boosting	89.48	89.6	0.9751	0.976	0.5339	0.5391	0.6863	0.6907
	Random Forest	90.87	91	0.9807	0.9816	0.5845	0.5903	0.7293	0.734
	SVM	86.46	86.52	0.9891	0.9894	0.3053	0.3081	0.4623	0.4658
	MLP	85.41	85.65	0.9647	0.9668	0.3516	0.369	0.4968	0.5177
SMOTE	Extra Tree	91.63	91.75	0.9833	0.9841	0.615	0.6207	0.7539	0.7583
	ID3	73.36	73.82	0.7461	0.7525	0.6721	0.6795	0.7029	0.7071
	CART	72.44	72.99	0.7348	0.7426	0.6715	0.6796	0.6964	0.7002
	Bagging	88.89	89.06	0.9358	0.9374	0.6768	0.6826	0.7829	0.7869
	Gradient Boosting	87.12	87.27	0.8944	0.8958	0.7662	0.7705	0.8237	0.8263
	Random Forest	90.78	90.91	0.9392	0.9406	0.765	0.7696	0.8416	0.8445
	SVM	81.46	81.59	0.8147	0.8161	0.8128	0.8159	0.8123	0.8141
ADASYN	MLP	82.05	82.35	0.8289	0.8328	0.7789	0.7853	0.8013	0.8049
	Extra Tree	92.5	92.64	0.9661	0.9671	0.7399	0.7454	0.836	0.8397
	ID3	63.64	64.28	0.6033	0.6121	0.7788	0.787	0.6747	0.6794
	CART	63.94	64.6	0.609	0.6183	0.7684	0.7777	0.6742	0.6788
	Bagging	88.26	88.44	0.9269	0.9286	0.6823	0.6867	0.7834	0.7879
	Gradient Boosting	85.7	86.89	0.8744	0.8763	0.7773	0.7825	0.8215	0.8247
	Random Forest	90.3	90.45	0.9307	0.9323	0.777	0.7818	0.8456	0.8486
SVM SMOTE	SVM	77.4	77.52	0.7506	0.752	0.8782	0.8807	0.8085	0.8098
	MLP	78.28	78.8	0.7747	0.7802	0.8206	0.8268	0.7947	0.7992
	Extra Tree	92.43	92.56	0.9614	0.9627	0.7562	0.761	0.845	0.8479
	ID3	73.53	73.91	0.7412	0.7469	0.7025	0.7106	0.7162	0.7197
	CART	72.95	73.43	0.7306	0.738	0.7165	0.726	0.7176	0.7215
	Bagging	89.41	89.56	0.935	0.9365	0.7086	0.7147	0.8038	0.8078
	Gradient Boosting	87.36	87.5	0.8939	0.8954	0.7809	0.7856	0.8321	0.8349
SVM SMOTE	Random Forest	91	91.14	0.9384	0.9398	0.7812	0.7864	0.8513	0.8543
	SVM	82.07	82.28	0.8253	0.8284	0.7969	0.8016	0.8095	0.8113
	MLP	82.88	83.19	0.8419	0.846	0.7657	0.7728	0.8001	0.8038
	Extra Tree	92.5	92.62	0.9653	0.9663	0.7425	0.748	0.8375	0.8411

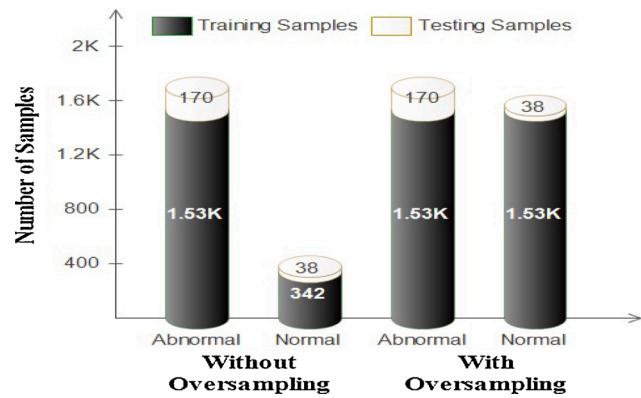


Fig. 6. Number of samples used in training and testing the classifiers with and without data oversampling.

using the same number of samples of the without oversampling case.

Table 2 presents a comparative analysis in terms of MSE, MAE, SNR and PSNR values that were obtained using different mother wavelet functions for denoising the input signal. The results show that symlet4 led to the lowest MAE and MSE values along with the highest SNR and PSNR values. Therefore, symlet4 was employed as the mother wavelet function. Table 3 presents the results as to MSE, MAE, SNR and PSNR metrics obtained using different wavelet decomposition levels to denoising the input signal. The results show that the first level of decomposition led to the lowest MAE and MSE values along with the highest SNR and PSNR values. Therefore, the decomposition level 1 (D1) was considered.

Table 4 allows the performance comparison of the different classifiers when used in combination with different oversampling methods. The results support the hypothesis that the oversampling improves all performance metrics. The Extra Tree classifier obtained the highest accuracy and F-score, and was followed by SVM, MLP, random forest, gradient boosting, bagging, CART and ID3.

In the case of the original data without oversampling, i.e., the imbalanced data, the Extra Tree classifier obtained an accuracy of 91.9% while for the MLP, SVM, Random Forest, Gradient Boosting, Bagging, CART and ID3 classifiers, it was of 84.9%, 86.5%, 90.9%, 89.4%, 89.5%, 82.8%, and 83.2%, respectively, as indicated in Table 4. Similarly, F-score was of 76.5%, 47.1%, 46.7%, 73.3%, 69.4%, 67.5%, 40.5% and 53.6% for the Extra Tree, MLP, SVM, Random Forest, Gradient Boosting, Bagging, CART and ID3 classifiers, respectively. In this case, the accuracy and sensitivity values were observed to be good, but the values obtained for the other performance metrics, mainly for F-score and specificity, were not so interesting.

After the use of the different oversampling techniques, it was found out that the classification accuracy could not be improved, but the other performance measures, mainly F-Score and sensitivity, had better values than the ones obtained from the original data. In the case of oversampling by SMOTE, the accuracies were of 93.1%, 85.5%, 81.4%, 90.7%, 87.5%, 89.2%, 75.6% and 76.2%, and as to F-score were of 85.3%, 80.7%, 81.6%, 84.3%, 83.8%, 79.3%, 71.8% and 70.0% for the extra tree, SVM, MLP, random forest, gradient boosting, bagging, CART and ID3 classifiers, respectively, as indicated in Table 4.

A receiver operating characteristic (ROC) curve is a tool commonly used to analyze the performance of classification models, mainly in two class classification problems. It is a graphical representation between the true positive rate (sensitivity) and false positive rate (1-specificity) [60]. Fig. 7 shows the ROC curve obtained for the studied classifiers when

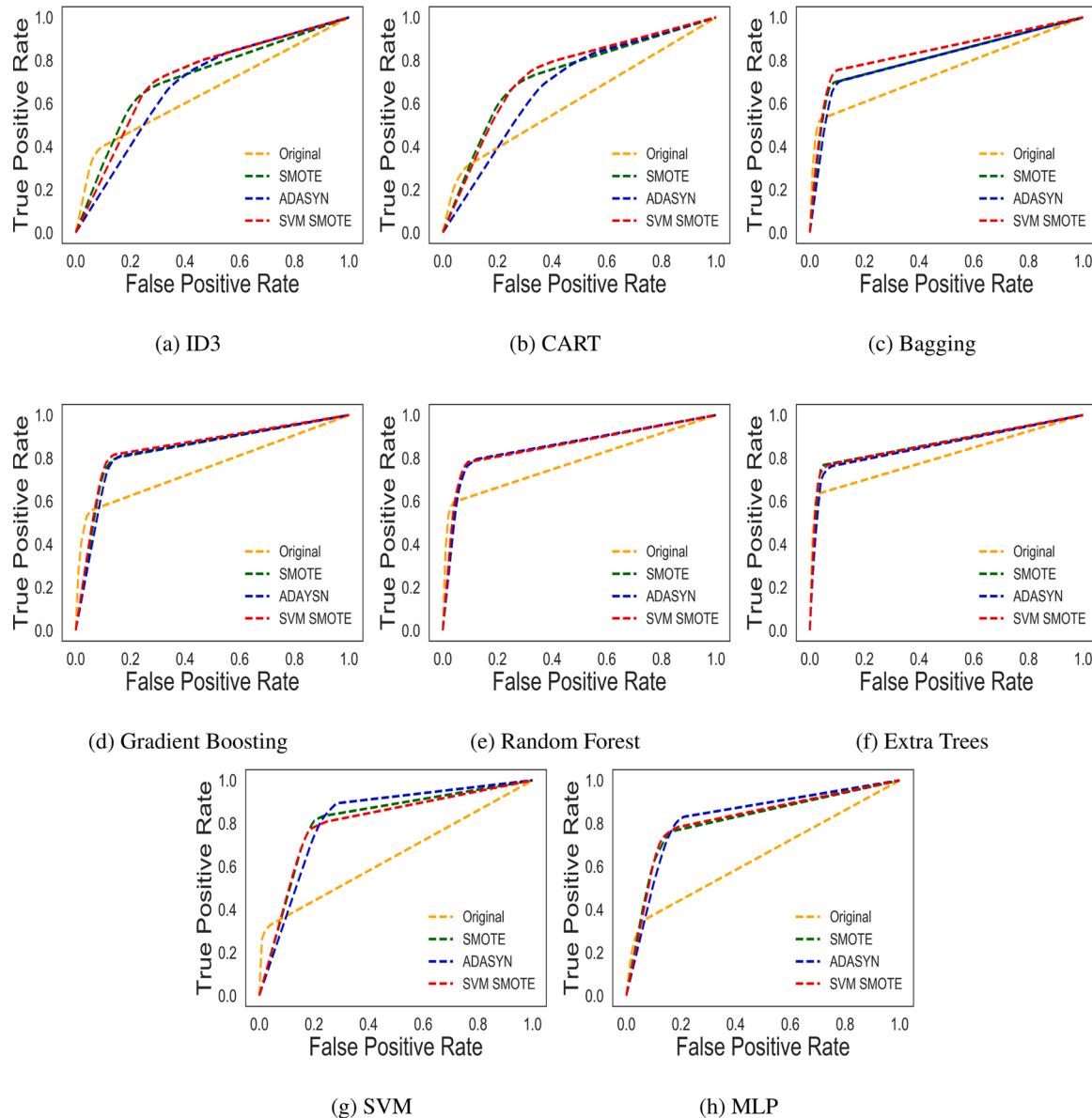


Fig. 7. ROC curve of classifier with and without oversampling techniques.

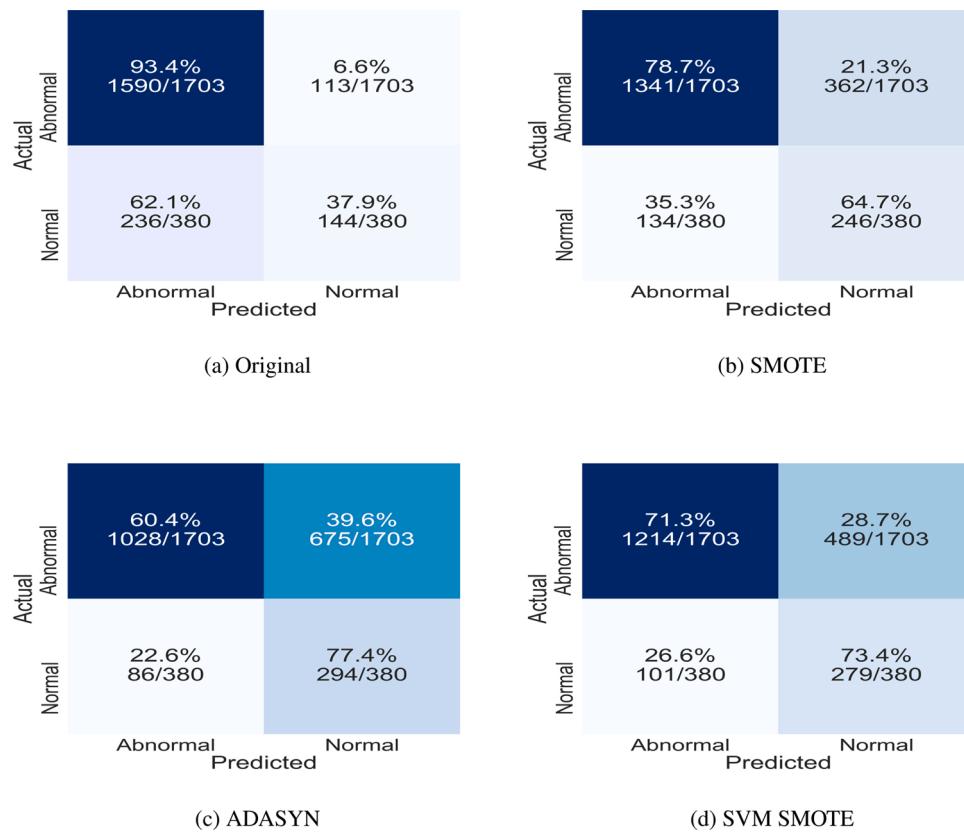


Fig. 8. Confusion Matrix of the ID3 classifier with and without oversampling.

applied to the original data and oversampled data. For all classifiers, the ROC curve built as to the oversampled data is far better than the one built from the as to the original data.

Fig. 8 shows the confusion matrices obtained for the ID3 and Extra Tree classifiers without data oversampling. The confusion matrix allows conceptualizing the performance of a classification algorithm in a tabular form. It contains information about the true and predicted labels evaluated by a model. In Fig. 8 (a), it can be seen that 62.1% of the samples of normal subjects were wrongly predicted as of abnormal subjects and 93.4% of the samples of abnormal subjects were correctly predicted as of abnormal subjects. Fig. 8 (b-d) show the confusion matrix of ID3 classifier obtained with different oversampling methods. From these confusion matrices, it is concluded that the used classification models were biased towards the majority class (abnormal subjects) with the original dataset but when minority class data is increased by using oversampling techniques then the classification models were not found as biased on a single class.

Data distribution can be analyzed with the help of box plots. Fig. 9 shows the box plot distributions for F-score with oversampling techniques of all studied classifiers according to the 10-fold cross-validation technique. For all the classifiers, the F-score obtained with oversampling was found to be far better than the one obtained without oversampling.

Table 5 shows the performance metrics obtained for each studied classifier according to a 95% confidence interval for 100 randomized tests. From the presented results, it may be observed that the range of values for all the metrics obtained for the different classifiers are narrow, for without and with oversampling. This allows for the conclusion that their respective distributions are not heterogeneous.

Furthermore, no significant difference was found between the performance metrics of the different oversampling techniques that have been implemented. However, if the results are compared in terms of with and without oversampling, then one can conclude that the oversampling can lead to promising knee abnormality detection even from

imbalanced sEMG data, independent of the classification model that is used.

6. Conclusion

Imbalanced data is a major problem in the classification of medical data, which may arise either due to the high differences in the number of healthy and unhealthy subjects or due to the length of the collected data be different for normal and abnormal subjects. In this work, the input raw sEMG signal was denoised by the wavelet denoising technique. After removing the noise, the sEMG signal was segmented and eleven-time domain features were extracted. After obtaining the features, the number of minority class samples was increased by using oversampling techniques, and then the performance of different classifiers on the imbalanced data and the oversampled, i.e., balanced, data was assessed. The obtained results suggest that oversampling techniques improve the performance of the classifiers in cases of imbalanced data. For that, four performance metrics (Accuracy, Sensitivity, Selectivity, and F-Score) were used to assess the classification accuracy of the sEMG data acquired from abnormal knee and healthy subjects during walking in terms of knee abnormality detection during walking.

In this study, one offline sEMG dataset was used to test the proposed approach. In the future, the approach can be validated using a large dataset acquired in real-time in a clinical scenario, which can validate its clinical use. Further, the classification can be performed without handcrafted feature extraction, and one may also try to reduce the extracted features space by using feature reduction techniques, which may decrease the computational time and improve the classification accuracy.

Authors' contributions

Ankit Vijayvargiya: Methodology, Software, Visualization,

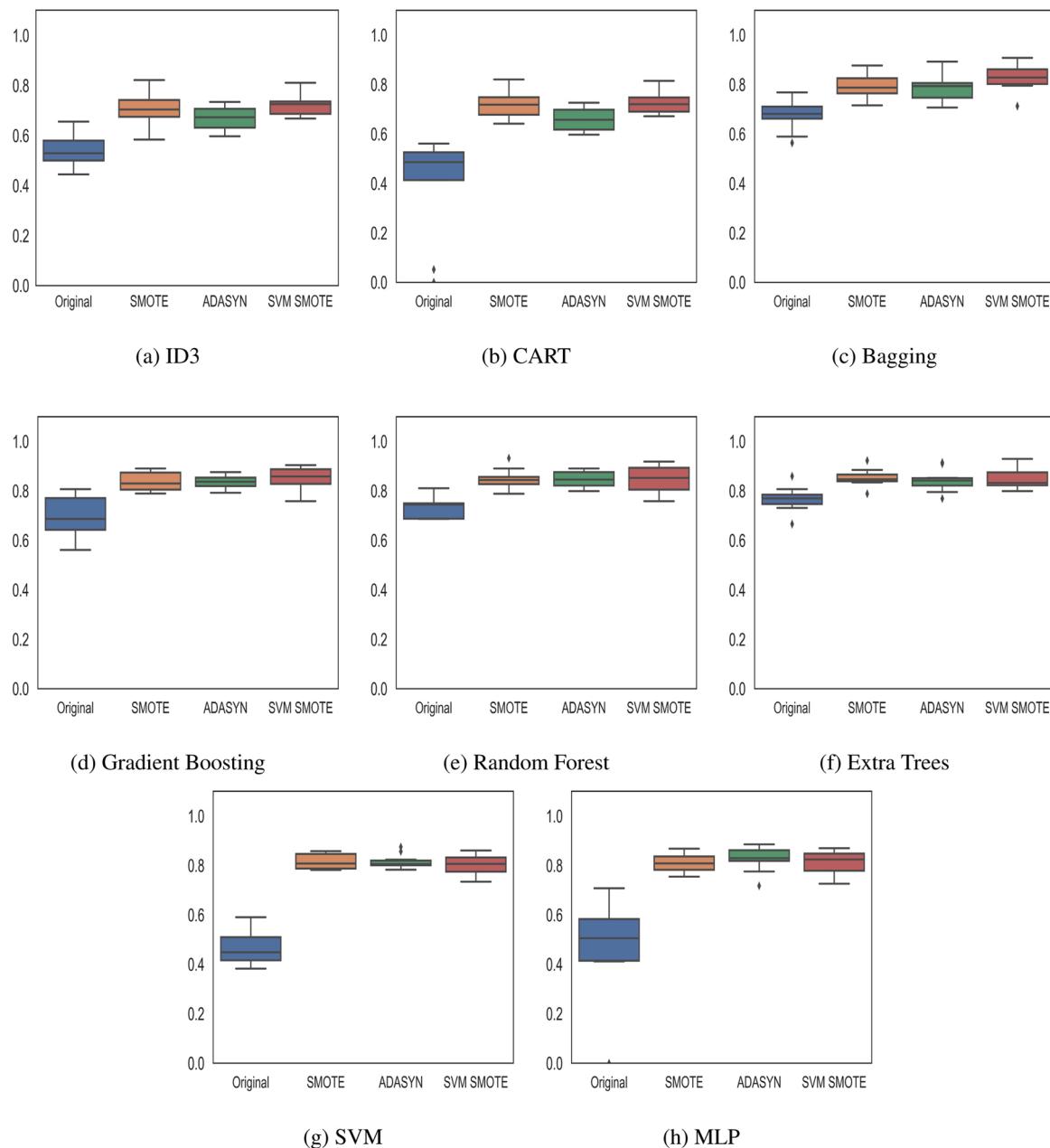


Fig. 9. Box plot for each classifier with applied oversampling techniques according to 10-fold cross validation.

Writing–Original draft preparation

Dr. Chandra Prakash: Writing–Original draft preparation, Validation

Dr. Rajesh Kumar: Conceptualization, Supervision, Writing–Reviewing and Editing

Dr. Sanjeev Bansal: Investigation, Validation

Dr. João Manuel R. S. Tavares: Writing–Reviewing and Editing

Compliance with ethical standards

Ethical approval This article does not contain any study with human participants performed by any of the authors.

Acknowledgement

This publication is supported by Visvesvaraya PhD Scheme, Meity, Govt. of India, MEITY-PHD-2942.

Declaration of Competing Interest

The authors report no declarations of interest.

References

- [1] A. Foundation, Arthritis by the Numbers, 2017.
- [2] J.L. Baldwin, The anatomy of the medial patellofemoral ligament, Am. J. Sports Med. 37 (12) (2009) 2355–2361.
- [3] A. Praemer, Musculoskeletal conditions in the united states, Am. Acad. Orthop Surg. 22 (1976) 1–199.
- [4] J. Hemavathi, T.S. Kumar, R.K. Prasad, Knee vibratography: arthritis diagnosis through non-invasive cloud based artificial intelligence, in: 2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), IEEE, 2017, pp. 1–6.
- [5] J. Bedson, K. Jordan, P. Croft, How do gps use x rays to manage chronic knee pain in the elderly?, a case study, Ann. Rheum. Dis. 62 (5) (2003) 450–454.
- [6] S. Kubakaddi, K. Ravikumar, D. Harini, Measurement of cartilage thickness for early detection of knee osteoarthritis (koa), in: 2013 IEEE Point-of-Care Healthcare Technologies (PHT), IEEE, 2013, pp. 208–211.

- [7] Chandra Prakash, Rajesh Kumar, Namita Mittal, Recent developments in human gait research: parameters, approaches, applications, machine learning techniques, datasets and challenges, *Artif. Intell. Rev.* 49 (1) (2018) 1–40.
- [8] B.-S. Yang, S.-T. Liao, Fall detecting using inertial and electromyographic sensors, Proceedings of the 36th Annual Meeting of the American Society of Biomechanics, Gainsville, FL, USA (2012) 15–18.
- [9] J. Cheng, X. Chen, M. Shen, A framework for daily activity monitoring and fall detection based on surface electromyography and accelerometer signals, *IEEE J. Biomed. Health Inform.* 17 (1) (2012) 38–45.
- [10] R. Merletti, C.J. De Luca, New Techniques in Surface Electromyography, 1989.
- [11] Suguru Kanoga, Atsunori Kanemura, Hideki Asoh, Are armband semg devices dense enough for long-term use?-sensor placement shifts cause significant reduction in recognition accuracy, *Biomed. Signal Process. Control* 60 (2020) 101981.
- [12] J. Yin, Q. Yang, J.J. Pan, Sensor-based abnormal human-activity detection, *IEEE Trans. Knowl. Data Eng.* 20 (8) (2008) 1082–1090.
- [13] S.K. Au, P. Bonato, H. Herr, An emg-position controlled system for an active ankle-foot prosthesis: an initial experimental study, in: 9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005, IEEE, 2005, pp. 375–379.
- [14] C.S. Pattichis, C.N. Schizas, Genetics-based machine learning for the assessment of certain neuromuscular disorders, *IEEE Trans. Neural Netw.* 7 (2) (1996) 427–439.
- [15] L. Burkow-Heikkinen, Non-invasive physiological monitoring of exercise and fitness, *Neurol. Res.* 33 (1) (2011) 3–17.
- [16] P.K. Shukla, A. Vijayvargiya, R. Kumar, et al., Human activity recognition using accelerometer and gyroscope data from smartphones, in: 2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3), IEEE, 2020, pp. 1–6.
- [17] Bernard Hudgins, Philip Parker, Robert N. Scott, A new strategy for multifunction myoelectric control, *IEEE Trans. Biomed. Eng.* 40 (1) (1993) 82–94.
- [18] Yanjiang Huang, Kaibin Chen, Xianmin Zhang, Kai Wang, Jun Ota, Joint torque estimation for the human arm from semg using backpropagation neural networks and autoencoders, *Biomed. Signal Process. Control* 62 (2020) 102051.
- [19] U.S.L.G. Da Silva, H.A. Villagra, L.L. Oliva, N.F. Marconi, Emg activity of upper limb on spinal cord injury individuals during whole-body vibration, *Physiol. Int. (Acta Physiol. Hung.)* 103 (3) (2016) 361–367.
- [20] S. Sudarsan, I. Student Member, E.C. Sekaran, Design and development of emg controlled prosthetics limb, *Procedia Eng.* 38 (2012) 3547–3551.
- [21] T. Tuncer, S. Dogan, A. Subasi, Surface emg signal classification using ternary pattern and discrete wavelet transform based feature extraction for hand movement recognition, *Biomed. Signal Process. Control* 58 (2020) 101872.
- [22] S. Cai, Y. Chen, S. Huang, Y. Wu, H. Zheng, X. Li, L. Xie, Svm-based classification of semg signals for upper-limb self-rehabilitation training, *Front Neurorobot.* 13 (2019) 31.
- [23] Krawczyk, Bartosz, Learning from imbalanced data: open challenges and future directions, *Prog. Artif. Intell.* 5 (4) (2016) 221–232.
- [24] A. Rainoldi, G. Melchiorri, I. Caruso, A method for positioning electrodes during surface emg recordings in lower limb muscles, *J. Neurosci. Methods* 134 (1) (2004) 37–43.
- [25] G.S. Murley, H.B. Menz, K.B. Landorf, A.R. Bird, Reliability of lower limb electromyography during overground walking: a comparison of maximal-and submaximal normalisation techniques, *J. Biomech.* 43 (4) (2010) 749–756.
- [26] H. Huang, F. Zhang, L.J. Hargrove, Z. Dou, D.R. Rogers, K.B. Englehart, Continuous locomotion-mode identification for prosthetic legs based on neuromuscular-mechanical fusion, *IEEE Trans. Biomed. Eng.* 58 (10) (2011) 2867–2875.
- [27] S. Mulroy, J. Gronley, W. Weiss, C. Newsam, J. Perry, Use of cluster analysis for gait pattern classification of patients in the early and late recovery phases following stroke, *Gait Posture* 18 (1) (2003) 114–125.
- [28] J. Chen, X. Zhang, Y. Cheng, N. Xi, Surface emg based continuous estimation of human lower limb joint angles by using deep belief networks, *Biomed. Signal Process. Control* 40 (2018) 335–342.
- [29] P. Bonato, M. Heng, J. Gonzalez-Cueto, A. Leardini, J. O'Connor, S.H. Roy, Emg-based measures of fatigue during a repetitive squat exercise, *IEEE Eng. Med. Biol. Mag.* 20 (6) (2001) 133–143.
- [30] R. Swaroop, M. Kaur, P. Suresh, P.K. Sadhu, Classification of myopathy and neuropathy emg signals using neural network, in: 2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT), IEEE, 2017, pp. 1–5.
- [31] P. Kugler, C. Jaremenko, J. Schlachetzki, J. Winkler, J. Klucken, B. Eskofier, Automatic recognition of parkinson's disease using surface electromyography during standardized gait tests, in: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2013, pp. 5781–5784.
- [32] Christian Morbidoni, Alessandro Cucchiarelli, Sandro Fioretti, Francesco Di Nardo, A deep learning approach to emg-based classification of gait phases during level ground walking, *Electronics* 8 (8) (2019) 894.
- [33] Gourav Takhar, Chandra Prakash, Namita Mittal, Rajesh Kumar, Vision-based gender recognition using hybrid background subtraction technique, in: International Conference on Next Generation Computing Technologies, Springer, 2017, pp. 651–662.
- [34] K.N. Rajesh, R. Dhuli, Classification of imbalanced ecg beats using re-sampling techniques and adaboost ensemble classifier, *Biomed. Signal Process. Control* 41 (2018) 242–254.
- [35] J. Nahar, T. Imam, K.S. Tickle, A.S. Ali, Y.-P.P. Chen, Computational intelligence for microarray data and biomedical image analysis for the early diagnosis of breast cancer, *Expert Syst. Appl.* 39 (16) (2012) 12371–12377.
- [36] L. Taft, R.S. Evans, C. Shyu, M. Egger, N. Chawla, J. Mitchell, S.N. Thornton, B. Bray, M. Varner, Countering imbalanced datasets to improve adverse drug event predictive models in labor and delivery, *J. Biomed. Inform.* 42 (2) (2009) 356–364.
- [37] O. Sanchez, J. Sotelo, M. Gonzales, G. Hernandez, Emg Dataset in Lower Limb Data Set, UCI Machine Learning Repository, 2014, 2014-02.
- [38] R. Chowdhury, M. Reaz, M. Ali, A. Bakar, K. Chellappan, T. Chang, Surface electromyography signal processing and classification techniques, *Sensors* 13 (9) (2013) 12431–12466.
- [39] C.-F. Jiang, S.-L. Kuo, A comparative study of wavelet denoising of surface electromyographic signals, in: 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, 2007, pp. 1868–1871.
- [40] A.O. Andrade, S. Nasuto, P. Kyberd, C.M. Sweeney-Reed, F. Van Kanijn, Emg signal filtering based on empirical mode decomposition, *Biomed. Signal Process. Control* 1 (1) (2006) 44–55.
- [41] Gokhan Altan, Yakup Kutlu, Novruz Allahverdi, Deep belief networks based brain activity classification using eeg from slow cortical potentials in stroke, *Int J. Appl. Math. Electr. Comput.* 4 (2016) 205–210.
- [42] Angkoon Phinyomark, Pornchai Phukpattaranont, Chusak Limsakul, Wavelet-based denoising algorithm for robust emg pattern recognition, *Fluct. Noise Lett.* 10 (02) (2011) 157–167.
- [43] A. Graps, An introduction to wavelets, *IEEE Comput. Sci. Eng.* 2 (2) (1995) 50–61.
- [44] C. He, J. Xing, J. Li, Q. Yang, R. Wang, A new wavelet threshold determination method considering interscale correlation in signal denoising, *Math. Probl. Eng.* (2015).
- [45] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, I. Rojas, Window size impact in human activity recognition, *Sensors* 14 (4) (2014) 6474–6499.
- [46] G.R. Naik, S.E. Selvan, S.P. Arjunan, A. Acharyya, D.K. Kumar, A. Ramanujam, H. T. Nguyen, An ica-ebm-based semg classifier for recognizing lower limb movements in individuals with and without knee pathology, *IEEE Trans. Neural Syst. Rehabil. Eng.* 26 (3) (2018) 675–686.
- [47] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, *J. Artif. Intell. Res.* 16 (2002) 321–357.
- [48] H. He, Y. Bai, E. Garcia, S. Li, Adaptive synthetic sampling approach for imbalanced learning. ieee international joint conference on neural networks, 2008 (IEEE World Congress On Computational Intelligence) (2008).
- [49] Q. Wang, Z. Luo, J. Huang, Y. Feng, Z. Liu, A novel ensemble method for imbalanced data learning: bagging of extrapolation-smote svm, *Comput. Intell. Neurosci.* (2017).
- [50] J. Ross Quinlan, Induction of decision trees, *Mach. Learn.* 1 (1) (1986) 81–106.
- [51] D.H. Moore, Classification and regression trees, by leo breiman, jerome h. friedman, richard a. olshen, and charles j. stone. brooks/cole publishing, monterey, 1984, 358 pages, *Cytometry* 8 (5) (1987) 534–535.
- [52] L. Breiman, Bagging predictors, *Mach. Learn.* 24 (2) (1996) 123–140.
- [53] L. Mason, J. Baxter, P.L. Bartlett, M.R. Frean, Boosting algorithms as gradient descent. *Advances in Neural Information Processing Systems*, 2000, pp. 512–518.
- [54] L. Breiman, Random forests, *Mach. Learn.* 45 (1) (2001) 5–32.
- [55] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, *Mach. Learn.* 63 (1) (2006) 3–42.
- [56] Zhang Xuegong, Introduction to statistical learning theory and support vector machines, *Acta Automatica Sinica.* 26 (1) (2000) 32–42.
- [57] X. Zhai, A.A.S. Ali, A. Amira, F. Bensaali, Mlp neural network based gas classification system on zynq soc, *IEEE Access* 4 (2016) 8138–8146.
- [58] G. Altan, Y. Kutlu, A.Ö. Pekmezci, S. Nural, Deep learning with 3d-second order difference plot on respiratory sounds, *Biomed. Signal Process. Control* 45 (2018) 58–69.
- [59] Ron Kohavi, et al., A study of cross-validation and bootstrap for accuracy estimation and model selection, *Ijcai*, Vol. 14, Montreal, Canada (1995) 1137–1145.
- [60] N. Nazmi, A. Rahman, M. Azizi, S.-I. Yamamoto, S.A. Ahmad, H. Zamzuri, S. A. Mazlan, A review of classification techniques of emg signals during isotonic and isometric contractions, *Sensors* 16 (8) (2016) 1304.

Research Article

Hybrid Deep Learning Approaches for sEMG Signal-Based Lower Limb Activity Recognition

Ankit Vijayvargiya ,^{1,2} Bharat Singh ,¹ Rajesh Kumar ,¹ Usha Desai ,³ and Jude Hemanth ,⁴

¹Malaviya National Institute of Technology, Jaipur, India

²Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur, India

³Department of Electronics and Communication Engineering, SR University, Warangal, Telangana, India

⁴Department of Electronics and Communication Engineering, Karunya University, Coimbatore, Tamilnadu, India

Correspondence should be addressed to Jude Hemanth; judehemanth@karunya.edu

Received 6 July 2022; Revised 16 October 2022; Accepted 28 October 2022; Published 26 November 2022

Academic Editor: Paolo Crippa

Copyright © 2022 Ankit Vijayvargiya et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Lower limb activity recognition utilizing body sensor data has attracted researchers due to its practical applications, such as neuromuscular disease detection and kinesiological investigations. The employment of wearable sensors including accelerometers, gyroscopes, and surface electromyography has grown due to their low cost and broad applicability. Electromyography (EMG) sensors are preferable for automated control of a lower limb exoskeleton or prosthesis since they detect the signal beforehand and allow faster movement detection. The study presents hybrid deep learning models for lower limb activity recognition. Noise is suppressed using discrete wavelet transform, and then the signal is segmented using overlapping windowing. Convolutional neural network is used for temporal learning, whereas long short-term memory or gated recurrent unit is used for sequence learning. After that, performance indices of the models such as accuracy, sensitivity, specificity, and F-score are calculated. The findings indicate that the suggested hybrid model outperforms the individual models.

1. Introduction

Lower limb activity recognition (LLAR) has increased in popularity due to its ability to monitor or identify daily lower limb human actions in a range of applications such as elderly health monitoring, surveillance and security systems, human fall detection, and so on [1, 2]. The two methodologies utilized for acquiring human activity data are visual and wearable sensors [3]. Wearable sensors such as inertial measurement units, goniometers, and sEMG electrodes are placed on the subject's body for data collection [4]. The vision-based approach has limited capability in terms of applicability, security and complexity [5]. Wearable sensors have seen significant technical advancements in recent times. It results in a lower overall cost, making it more accessible. Popular sensors used in wearable research for activity recognition include inertial measurement units,

accelerometers, gyroscopes, electromyography, and barometers [6]. EMG sensors are better than others since they can predict movement in advance in a very short amount of time. Out of these sensors, the EMG sensors are superior because they can anticipate movement in advance in a very short amount of time [7, 8]. Neuromuscular activity generates the biological signal known as the EMG signal. It can be detected by the electrical currents in muscles during the muscle contraction. Surface (non-invasive) EMG and intramuscular (invasive) EMG are the two approaches that are employed for the recording of the EMG signal [9]. Intramuscular electromyogram (iEMG) signals are captured by placing the wire electrodes within the muscles, whereas surface electromyogram (sEMG) signals are captured by placing the surface electrodes just above the muscle's surface. The following are the advantages of sEMG over the iEMG [10]:

- (1) There is no requirement of medical supervision for placing the electrodes; also, there is no discomfort.
- (2) Infection risk is mitigated.

Surface EMG signals are used in a wide range of healthcare applications, which include the control of prostheses or exoskeletons, neuromuscular disease assessment, activity monitoring, and many more [11–13]. According to Kiguchi et al., sEMG signal-based neuro-fuzzy approach can be used to control an upper limb robotic exoskeleton [14]. It is found that sEMG signals can be used for multiple applications. Krasin et al. [15] proposed the low-cost elbow joint powered exoskeleton. The major goal of this sEMG signal-based exoskeleton is for strengthening the biceps brachii. Sharmila et al. [16] presented a low-cost sEMG-controlled prosthetic arm for upper limb amputees. Sensors are employed for the recording of the sEMG signals from the muscles during various activities to control prostheses autonomously. Then, the actuators can be controlled using artificial intelligence methodology. Pancholi and others [17] developed hardware for amputees to recognize real-time arm gestures. Vijayvargiya et al. developed low-cost sEMG data acquisition system for collecting the sEMG signals [18]. Cai et al. [19] identified the upper limb motion pattern for controlling a rehabilitation robot using sEMG data and a support vector machine approach.

Compared to lower limb sEMG signals, much emphasis has been dedicated to the classification and pattern recognition of upper limb sEMG signals in the recent decade. Classification of lower limb sEMG signals has shown to be more complicated than the classification of upper limb signals. It is because of the complexity induced due to the inherent coupling of lower limb sEMG signals. Souit et al. [20] presented the control approach using sEMG for a lower limb exoskeleton. The exoskeleton can be operated autonomously utilizing artificial intelligence approaches by analyzing the sEMG signal produced by muscles during various activities. Khimraj et al. [21] investigated classification between six lower limb activities and evaluated the performance of various machine learning classifiers for the same. Silva et al. [22] carried out research investigation on spinal cord damage based on the EMG signal that has been captured while the upper limb movements were being performed. Vijayvargiya et al. [23] investigated the detection of knee abnormalities using unbalanced sEMG data for walking activity. Here, the authors demonstrated the impact of an imbalanced signal on the model performance for the detection of a knee issue and evaluated the performance of several oversampling methods to improve machine learning model performance. Ertugrul et al. [24] proposed an adaptive local binary pattern (ALBP) approach for the retrieval of the characteristics and classification of healthy and abnormal knee participants with an accuracy of 85%. Handcrafted characteristics from signals are required for machine learning models, which may be extracted using statistical methods. Choosing the proper feature set manually is a tedious process. According to the available literature, deep learning models such as CNN, LSTM, GRU, and other techniques have been employed to solve the problem

[25]. The features are retrieved first by the algorithm in deep learning techniques, and then the classification procedure is carried out. According to earlier studies, these deep learning models have been used for various applications and have demonstrated very high performance [26–30].

This study aims to apply hybrid deep learning algorithms to identify lower limb activity. The authors present the models that integrate the advantage of convolutional neural network (CNN), long short-term memory (LSTM), and gated recurrent unit (GRU). CNN architecture has multiple layers such as the input layer, output layer, dense layer, convolutional layer, rectified linear unit layers, and dropout layers. One of the significant issues with CNN is its inability to analyze the characteristics of the time-series data such as the previous or temporal data. Therefore, the LSTM or GRU can be employed for the analysis of data which efficiently collects temporal information present in the data. As a result, a combination of CNN with LSTM (CNN-LSTM) or CNN with GRU (CNN-GRU) is proposed as the better strategy for processing EMG data. In the suggested hybrid models, a convolutional neural network provides temporal learning, whereas the LSTM or GRU is used for capturing sequence-to-sequence learning. The major contributions of this study are as follows:

- (1) Surface electromyography (sEMG) data acquired from leg muscles are used in this study to examine lower limb movements in healthy and knee deformity individuals by using the hybrid deep learning framework.
- (2) The preprocessing technique known as wavelet denoising is applied in order to eliminate noises of sEMG signal.
- (3) Hybrid deep learning models, CNN-LSTM and CNN-GRU, are proposed for recognition of lower limb activities. Here, a convolutional neural network (CNN) is used for temporal learning, while long short-term memory (LSTM) or gated recurrent unit (GRU) is used for sequence learning.
- (4) The proposed hybrid CNN-GRU model has high performance compared to the existing models.

2. Dataset

The authors used the publicly accessible sEMG signal dataset from the UCI machine learning repository by Sanchez et al. in their study [24, 31]. The data comprise sEMG signals from the lower extremities of 22 individuals over the age of 18, 11 of which have been healthy and 11 of which have known knee injuries. There is no history of knee injury or discomfort in the healthy participants. The sciatic nerve was injured in one abnormal knee subject, the anterior cruciate ligament (ACL) was injured in six abnormal knee subjects, and the meniscus was injured in the remaining four abnormal knee subjects. The sEMG signals were acquired using a Biometrics Ltd. DataLog MWX8 and a goniometer while the participants performed one of three tasks: walking, sitting, and standing. The biceps femoris (BF), vastus

medialis (VM), rectus femoris (RF), and semitendinosus (ST) muscles had their sEMG data taken with goniometer affixed towards the outer surface of the knee. The sEMG signal was recorded on the damaged limb of the person with a defective knee and the left leg of the healthy individuals. The data were collected at a sample rate of 1 kHz with a resolution of 14 bits. A band-pass filter with a passband frequency of 20 Hz to 460 Hz has already been applied to the sEMG signals. The sEMG signals collected throughout each activity by the healthy and abnormal knee subjects are shown in Figure 1.

3. Proposed Methodology

This section explains the methods used for lower limb activity recognition (wavelet denoising, segmentation, and deep learning frameworks). Figure 2 depicts the proposed deep learning-based method for the identification of the lower limb activity based on the sEMG signals. First, noise is removed from the raw sEMG signal with the help of discrete wavelet transform, and then the signal is segmented using the overlapping windowing technique. After that, deep learning models CNN, CNN-LSTM, and CNN-GRU are applied to identify the lower limb activities in healthy and abnormal knee individuals. In these hybrid models, a convolutional neural network (CNN) is used for temporal learning, while long short-term memory (LSTM) or gated recurrent unit (GRU) is used for sequence learning.

3.1. Wavelet Denoising. Various types of noises have been interlaced with the sEMG signals during the recording. The most prominent noises are as follows [32]:

- (i) Electronic devices and electromagnetic interference generate inherent and ambient noises, respectively.
- (ii) The subject's walking leads to electrodes' movement, which induces the artifacts.
- (iii) The firing rate of motor units can lead to the inherent instability noise.

These noises can badly impact the performance of the classifier. Therefore, it is required to filter the noise from the sEMG signal. There are various topologies proposed by various researchers in the literature. Traditional approaches such as low pass, high pass, and band pass can be employed for the filtering of noises that do not fall within the frequency band of the sEMG signal, which is 20 to 460 Hz. However, these methods have failed to filter noise in the active spectrum of sEMG signals. In recent times, several researchers have successfully employed empirical mode decomposition, independent component analysis, and wavelet decomposition for the filtering of noises from the sEMG signals [33].

The use of wavelet decomposition [34, 35] has seen a rising trend in sEMG signal denoising for both the upper and lower limbs. This is because it can effectively eliminate the white Gaussian noise from the signal. In the wavelet method, firstly, the mother wavelet function is selected. Then, the frequency and temporal analysis can be performed

by the low and high-frequency versions of the wavelet, respectively. Wavelet decomposition includes the following steps:

- (i) The signal is decomposed using the discrete wavelet transform.
- (ii) A threshold is chosen.
- (iii) The signal is reconstructed using the inverse wavelet transform.

Decomposition generates the approximation and detail coefficients. The level of decomposition decides the number of the coefficients. After this, thresholding is applied. Thresholding forced all coefficients to zero, which is below a certain threshold. Afterward, the signal is reconstructed again. By scaling (s) and translating (τ) a single basic wavelet $\Psi_{s,\tau}$, many mother wavelets are created. The mathematical expression of basic wavelet is presented in the following equation:

$$\Psi_{s,\tau} = \frac{1}{\sqrt{s}} \Psi\left(\frac{t - \tau}{s}\right). \quad (1)$$

As per literature, multiple thresholding approaches are employed, such as hard, soft, and universal. In this work, the authors have employed a db4 mother wavelet from the Daubechies family with four decomposition levels and applied garotte thresholding on second detail coefficients.

3.2. Segmentation. Since the sEMG signal's characteristic is stochastic, it is suggested by Vijayvargiya et al. to segment the sEMG signal into smaller portions. However, different lengths of sEMG signal impact the classifier accuracy. Therefore, commonly used windowing topologies such as adjacent and overlapping are employed on the signal segmentation. Here, the authors have used the overlapping-based windowing technique with a window size of 256 ms and 25% overlapping [36].

3.3. Deep Learning Framework

3.3.1. Convolutional Neural Network (CNN). Generally, the artificial neural network has a fully connected network of neurons, which means every neuron is connected via weights to the next layer completely. This connectivity can lead to overfitting of the network. To address this problem, several researchers in the literature have presented regularization techniques, which involve the magnitude of the weight-to-loss ratio. Similarly, CNN has been adopted for regularization because of its ability to progressively exploit the dataset's structure by gathering the simple structures. It uses the principle of convolution instead of matrix pointwise multiplication. An additional advantage is that these networks have fewer parameters as compared to other fully connected neural networks. Therefore, the training of these networks is fast. A CNN includes a sequence of multiple layers such as input, hidden, and output layers. Hidden layers constitute the three different types of layers which have convolutional, max-pooling, and fully connected

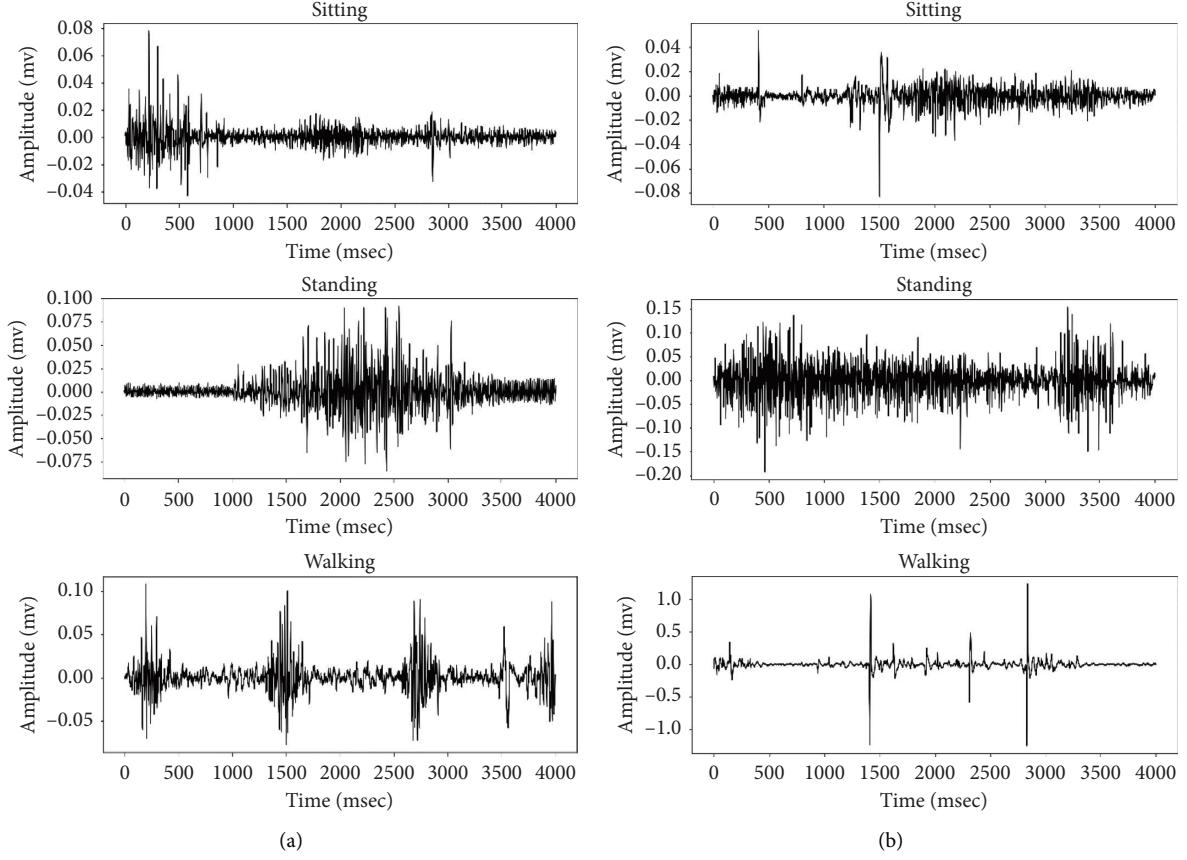


FIGURE 1: sEMG signals acquired during walking, standing, and sitting: (a) normal subject and (b) abnormal subject.

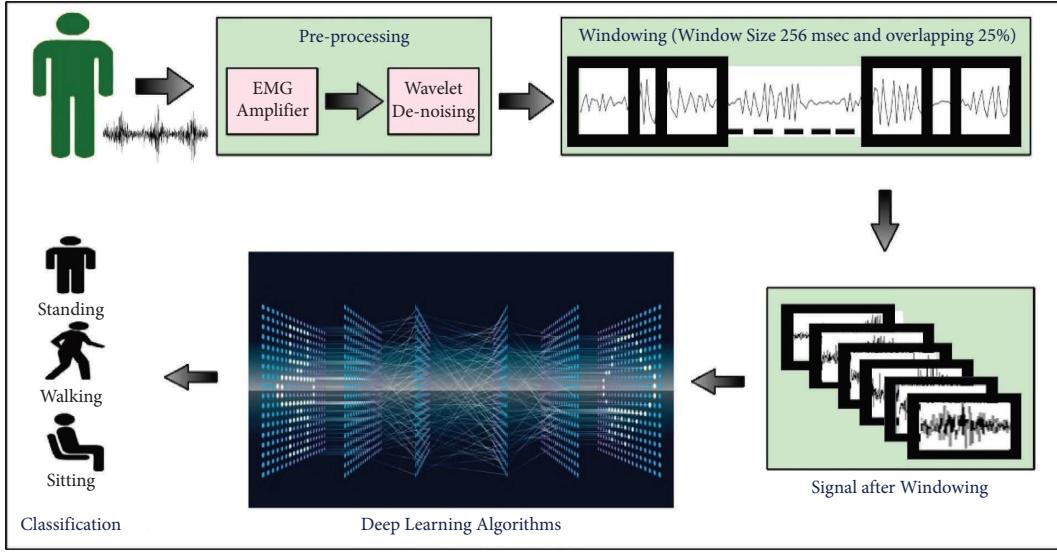


FIGURE 2: Block diagram of the proposed methodology for lower limb activity.

layers. The convolutional layer convolves with multiplication. Generally, ReLU is actively employed as the activation function in the neuron.

Mainly, CNN is employed on 2D datasets such as video or images. Therefore, CNN is named as the 2D CNN. As per

the literature, some authors have modified 2D CNN to 1D CNN [37].

Mathematical expressions of layers of CNN are described in equations (2) to (9).

(1) *Convolution Layer.*

(i) Forward propagation:

$$a_{ij}^{(k)} = \sum_{s=0}^{m-1} \sum_{t=0}^{n-1} W_{st}^{(k)} x_{(i+s)(j+t)} + b^{(k)}. \quad (2)$$

(ii) Backpropagation to update the weight:

$$\begin{aligned} \frac{\delta E}{\delta W_{st}^{(k)}} &= \sum_{i=0}^{M-m} \sum_{j=0}^{N-n} \frac{\delta E}{\delta a_{ij}^{(k)}} \frac{\delta a_{ij}^{(k)}}{\delta W_{st}^{(k)}} \\ &= \sum_{i=0}^{M-m} \sum_{j=0}^{N-n} \frac{\delta E}{\delta a_{ij}^{(k)}} x_{(i+s)(j+t)}, \end{aligned} \quad (3)$$

$$\begin{aligned} \frac{\delta E}{\delta b^{(k)}} &= \sum_{i=0}^{M-m} \sum_{j=0}^{N-n} \frac{\delta E}{\delta a_{ij}^{(k)}} \frac{\delta a_{ij}^{(k)}}{\delta b^{(k)}} \\ &= \sum_{i=0}^{M-m} \sum_{j=0}^{N-n} \frac{\delta E}{\delta a_{ij}^{(k)}}. \end{aligned} \quad (4)$$

(iii) Backpropagation to previous layer:

$$\begin{aligned} \frac{\delta E}{\delta x_{ij}} &= \sum_{s=0}^{m-1} \sum_{t=0}^{n-1} \frac{\delta E}{\delta a_{(i-s)(j-t)}^{(k)}} \frac{\delta a_{(i-s)(j-t)}^{(k)}}{\delta x_{ij}} \\ &= \sum_{s=0}^{m-1} \sum_{t=0}^{n-1} \frac{\delta E}{\delta a_{(i-s)(j-t)}^{(k)}} W_{st}^{(k)}. \end{aligned} \quad (5)$$

(2) Max-Pooling Layer.

(i) Forward propagation:

$$a_{ij} = \max(0, x_{(i+s)(j+t)}). \quad (6)$$

(ii) Backpropagation:

$$\begin{aligned} \frac{\delta E}{\delta x_{(i+s)(j+t)}} &= \frac{\delta E}{\delta a_{ij}^{(k)}}, \\ \frac{\delta a_{ij}^{(k)}}{\delta x_{(i+s)(j+t)}} &= \begin{cases} \frac{\delta E}{\delta a_{ij}^{(k)}}, & \text{if } a_{ij}^k = x_{(i+s)(j+t)}, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (7)$$

(3) Fully Connected Layer.

(i) Forward propagation of ReLU activation function:

$$a_{ij} = \max(0, x_{ij}). \quad (8)$$

(ii) Backpropagation of ReLU activation function:

$$\frac{\delta E}{\delta x_{ij}} = \frac{\delta E}{\delta a_{ij}^{(k)}} \frac{\delta a_{ij}^{(k)}}{\delta x_{ij}} = \begin{cases} \frac{\delta E}{\delta a_{ij}^{(k)}}, & \text{if } a_{ij}^k \geq 0, \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

where x is the input, a^k denotes the output after convolution layer k , k denotes the layer index, W denotes the kernel (filter), $m * n$ denotes the filter size, $M * N$ denotes the input size, b denotes the bias, and E denotes the cost function.

3.3.2. Long Short-Term Memory (LSTM). Conventional neural networks failed to achieve the desired accuracy for sequential time-series datasets because they do not have any memory element to store the previous state. This means it only relies on the present state. Therefore, the recurrent neural network allows a feedback mechanism to learn the previous step state. This structure of recurrent provides better handling of the time-series data. However, it can often encounter a problem when the network becomes deeper. Vector-Jacobian gradient vector tends to shrink due to its repetitive nature. This can lead to training being slower and less effective. This phenomenon is known to be a gradient vanishing problem. Thus, these recurrent neural networks perform badly in terms of long-term dependency. Thus, the long short-term memory (LSTM) [38] can effectively deal with the gradient vanishing problem by tackling the long-term dependencies.

The architecture of LSTM consists of an LSTM cell. Each cell contains a well-designed vectorized pointwise multiplication between the new and the previous states. In addition, mathematical functions such as the hyperbolic tan and sigmoid functions are employed to control the flow of information instead of only a single layer as in simple recurrent layers. The LSTM cell consists of the three operational gates with memory element via feedback, and each gate has its bias and weight vectors. Three gates are forget gate, input/update gate, and output gate. Firstly, the current and previous activation state information flows through the sigmoid function, and this layer is known to be the forget layer. This function squashed the information into the range of 0 and 1. This operation quantifies how much information is required for the next step prediction. Secondly, the processed information from the forget gate and input vector flows through the hyperbolic tangent function. This layer is known as the input/update layer because it uses the previous state and present state to generate the new information for the cell. Finally, the linear vector-matrix addition is performed. Then, the information flowed through the hyperbolic tangent. Afterward, the output gate scaled the resultant final values from the previous gate. Overall, the sigmoid function controls the flow such as which information is required or what is needed to forget from the previous cell. It

significantly increases the accuracy. The mathematical expression is given in equations (10) to (15).

Forget gate:

$$F_t = \sigma(W_{fx}X_t + W_{fh}H_{t-1} + B_f). \quad (10)$$

Input gate:

$$I_t = \sigma(W_{ix}X_t + W_{ih}H_{t-1} + B_i). \quad (11)$$

Output gate:

$$O_t = \sigma(W_{ox}X_t + W_{oh}H_{t-1} + B_o). \quad (12)$$

Intermediate state:

$$\check{C}_t = \tanh(W_c X_t + W_c H_{t-1} + B_c). \quad (13)$$

Final state:

$$C_t = F_t \odot C_{t-1} + I_t \odot \check{C}_t. \quad (14)$$

New state:

$$H_t = O_t \odot \tanh(C_t), \quad (15)$$

where W and B are the layer weight and bias vectors, respectively, and the input vector is denoted by X .

3.3.3. Gated Recurrent Unit (GRU). The gated recurrent unit [39] is a variant of the LSTM. Therefore, it has similar properties as the LSTM, and it also solves the vanishing gradient problem. This overall improves the learning long-term capability of the neural network. It also has the sigmoid and hyperbolic tangent functions. It does not have a separate forget and input gate like LSTM, which means not having a separate memory element. Therefore, it has fewer parameters as compared to the LSTM cell. It yields efficiently compared to the former. The update gate consists of both sigmoid and hyperbolic tangent functions. In addition, a reset gate is also designed in the GRU cell. All process flow is the same as the LSTM cell. The mathematical expression is given in equations (16) to (19).

Update gate:

$$U_t = \sigma(W_{ux}X_t + W_{uh}H_{t-1} + B_u). \quad (16)$$

Reset gate:

$$R_t = \sigma(W_{rx}X_t + W_{rh}H_{t-1} + B_r). \quad (17)$$

Internal activation:

$$\check{H}_t = \tanh(W_{hx}X_t + W_{hh}(R_t \odot H_{t-1}) + B_h). \quad (18)$$

Output activation:

$$H_t = U_t \odot \check{H}_t + (1 - U_t) \odot H_{t-1}, \quad (19)$$

where W and B represent the respective layer weight and bias vector and X represents the input vector.

3.3.4. Proposed Hybrid Models. One of the most serious shortcomings of CNN is its inability to analyze historical or

temporal data contained within time-series signal, as it lacks a memory element for storing the previous state. This means that it is entirely dependent on the current situation. Thus, a feedback mechanism enables the recurrent neural network to learn the previous step state. This recurrent structure enables more efficient handling of time-series data. However, it frequently encounters a gradient vanishing challenge and performs badly in long-term dependency. By addressing long-term dependencies, long short-term memory or gated recurrent unit, the variants of the recurrent neural network, can successfully address the gradient vanishing problem.

In this study, the authors proposed a hybrid of a convolutional neural network with long short-term memory and gated recurrent unit to recognize lower limb activities. CNN is employed for capturing the temporal relationship present in the dataset, whereas the LSTM or GRU is employed for sequence-to-sequence learning. The authors have used the dataset collected from the four channels collected at a sampling frequency of 1 kHz. Then, the signal is segmented as 256 ms. Thus, the total window length is equal to 256 samples. Therefore, the four channels are connected serially. It results in an array size of 1×1024 , which can be applied as the input for the 1D CNN. Firstly, the input features are normalized in range of 0 to 1. Then, the normalized signal is passed through the two convolutional layers which consist of the convolutional and non-linear layers (ReLU). Afterward, the processed data are passed through the max-pooling layer and then followed through two LSTM or GRU layers. After that, the signal is passed through the two fully connected layers. Feature dimension is reduced by integrating the pooling layer after the convolutional layer. The parameters of studied deep learning models are shown in Table 1 which are found out using the trial-and-error approach.

4. Results and Discussion

We show our findings for several situations involving lower limb activity recognition in this section. The suggested model was trained and evaluated using TPU Google Colab, a cloud-based system to detect a human's lower limb actions. The model's code within Colab notebook executes on a Google cloud server. Python's Keras modules have been used to recognize the various 1D CNN models, and the outcomes were computed. The Adam optimizer integrates the classic backpropagation approach with a cross-entropy loss function and employs the stochastic gradient descent strategy. The hyperparameters of the optimizer have the following values: learning rate (0.001), epsilon (0.00000001), beta 1 (0.9), beta 2 (0.999), and locking (false).

In this investigation, total of 22 subjects consisting of normal persons and knee difficulty persons in equal numbers were examined. The performance indicators were evaluated on (1) healthy adults and (2) individuals with a knee anomaly. The authors have considered starting 70% of the signal for every subject as a training dataset and the

TABLE 1: Parameters considered in the proposed study using deep learning models.

Model parameters	CNN	LSTM	GRU	CNN-LSTM	CNN-GRU
Convolution layer 1	Number of filters	16	—	16	16
	Kernel size	3	—	3	3
Convolution layer 1	Number of filters	8	—	8	8
	Kernel size	3	—	3	3
Pooling layer	Type of pooling	Max	—	Max	Max
	Kernel size	2	—	2	2
LSTM	Unit size	—	16	16	—
LSTM	Unit size	—	8	8	—
GRU	Unit size	—	—	16	16
GRU	Unit size	—	—	8	8
Fully connected	Unit size	16	16	16	16
Fully connected	Unit size	3	3	3	3

remaining 30% as a testing dataset. It helps in the reduction of the temporal dependencies and arbitrarily tests set selection. The authors have considered five performance metrics such as accuracy, precision, sensitivity, specificity, and F-score like the previous study of Vijayvargiya et al. [40].

Table 2 shows the summary indices of the studied deep learning models in percentage for the three activities such as walking, sitting, and standing, under research obtained from sEMG data gathered from healthy people, whereas Table 3 shows the performance for knee abnormality subjects. The data in these tables allow the individual models to be compared in terms of participants with and without knee abnormalities, confirming that the suggested hybrid models outperform the individual model. In the case of the healthy subjects, the CNN-GRU model obtained an accuracy of 99.86% while for CNN, LSTM, GRU, and CNN-LSTM, it was 98.88%, 95.38%, 98.32%, and 99.02%, respectively, as indicated in Table 2. Similarly, F-score value of CNN-GRU model was 99.79% while for CNN, LSTM, GRU, and CNN-LSTM, it was 98.31%, 93.82%, 97.51%, and 98.50%, respectively. In the case of the abnormal knee subjects, the CNN-GRU model obtained an accuracy of 98.69% while for CNN, LSTM, GRU, and CNN-LSTM, it was 92.62%, 54.54%, 96.69%, and 97.62%, respectively, as indicated in Table 3. Similarly, F-score value of the CNN-GRU model was 98.61% while for CNN, LSTM, GRU, and CNN-LSTM, it was 92.22%, 47.64%, 96.55%, and 97.42%, respectively.

Table 4 presents the time taken to complete an epoch of studied deep learning models. It indicates that when the CNN layer is used, the time taken for completion of an epoch is less required than the individual LSTM or GRU model. In healthy participants, the proposed hybrid CNN-GRU and CNN-LSTM deep learning models needed 4.26 s and 4.34 s computing time to execute an epoch, respectively. In contrast, the CNN, LSTM, and GRU deep learning models required 2.26 s, 94.06 s, and 117.60 s. In abnormal knee subjects, the proposed hybrid CNN-GRU and CNN-LSTM deep learning models needed 4.58 s and 6.15 s computing time to execute an epoch, respectively, whereas the CNN, LSTM, and GRU deep learning models required 5.19 s, 98.87 s, and 106.67 s. When the proposed hybrid models were examined in terms of computational

TABLE 2: Performance parameters for healthy subjects (in %).

Approach/Parameters	CNN	LSTM	GRU	CNN-LSTM	CNN-GRU
Accuracy	98.88	95.38	98.32	99.02	99.86
Precision	98.49	93.81	97.49	99.20	99.88
Recall/sensitivity	98.13	93.86	97.53	97.88	99.70
Specificity	99.45	97.73	99.20	99.46	99.92
F-score	98.31	93.82	97.51	98.50	99.79

TABLE 3: Performance parameters for abnormal knee subjects (in %).

Approach/Parameters	CNN	LSTM	GRU	CNN-LSTM	CNN-GRU
Accuracy	92.62	54.54	96.69	97.62	98.69
Precision	92.67	79.63	96.50	97.35	98.61
Recall/sensitivity	91.92	51.60	96.62	97.54	98.62
Specificity	96.22	75.31	98.37	98.87	99.36
F-score	92.22	47.64	96.55	97.42	98.61

time, a significant difference was observed for the hybrid models (CNN-LSTM and CNN-GRU) and individual models LSTM and GRU, and a very small variance was observed between the hybrid models and CNN, but the accuracy and F-score values are relatively high in hybrid models than in individual models.

Figure 3 presents the confusion matrices for the CNN, LSTM, GRU, CNN-LSTM, and CNN-GRU models for healthy patients, whereas Figure 4 depicts the confusion matrices obtained for abnormal knee subjects. The confusion matrix provides a tabular representation of the performance of a classification method. It comprises the data on the real and predicted labels from the model. As illustrated in Figure 3(a), 105, 282, and 320 samples of walking, sitting, and standing activities are accurately predicted, but five samples of walking activity are incorrectly predicted as sitting and three samples of sitting activity are incorrectly predicted as walking.

Figure 5 depicts the variation between the loss vs. epoch for the CNN, LSTM, GRU, CNN-LSTM, and CNN-GRU models for healthy patients, whereas Figure 6 depicts the training plot between loss vs. epoch for abnormal knee

TABLE 4: Time taken per epoch (in seconds).

Approach/subjects	CNN	LSTM	GRU	CNN-LSTM	CNN-GRU
Healthy subjects	2.26	94.06	117.60	4.34	4.26
Abnormal knee subjects	5.19	98.87	106.67	6.15	4.58

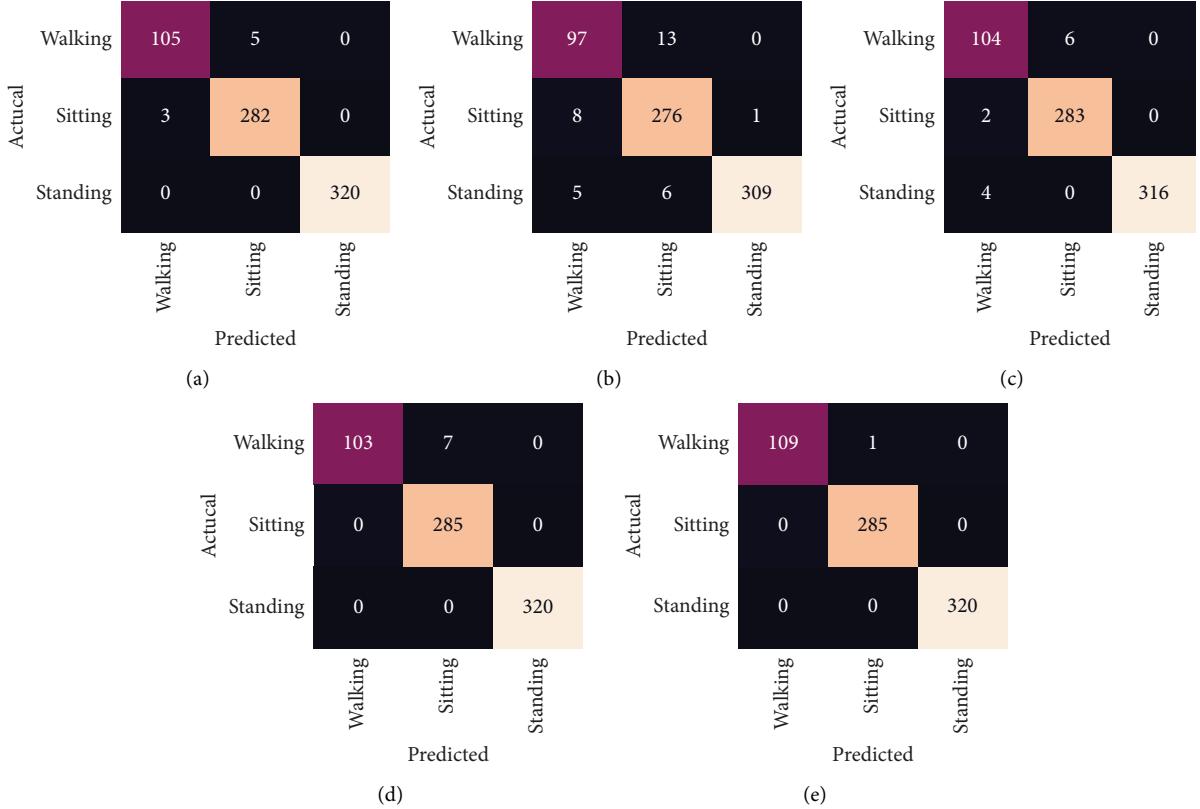


FIGURE 3: Confusion metrics for healthy subjects. (a) CNN. (b) LSTM. (c) GRU. (d) CNN-LSTM. (e) CNN-GRU.

subjects. These plots demonstrate that the value of the loss function decreases as the number of epoch increases, until it approaches a steady state, which indicates that the overfitting problem is resolved. The proposed hybrid models achieve a steady state as they approach the tenth epoch, which is significantly better than the other examined models.

Numerous approaches for recognizing lower limb activity have been presented using similar datasets. Table 5 presents a comparison assessment of the proposed model's performance vs. prior findings, allowing us to conclude that the proposed CNN-GRU model performed well for recognizing lower limb activity in healthy and abnormal knee individuals.

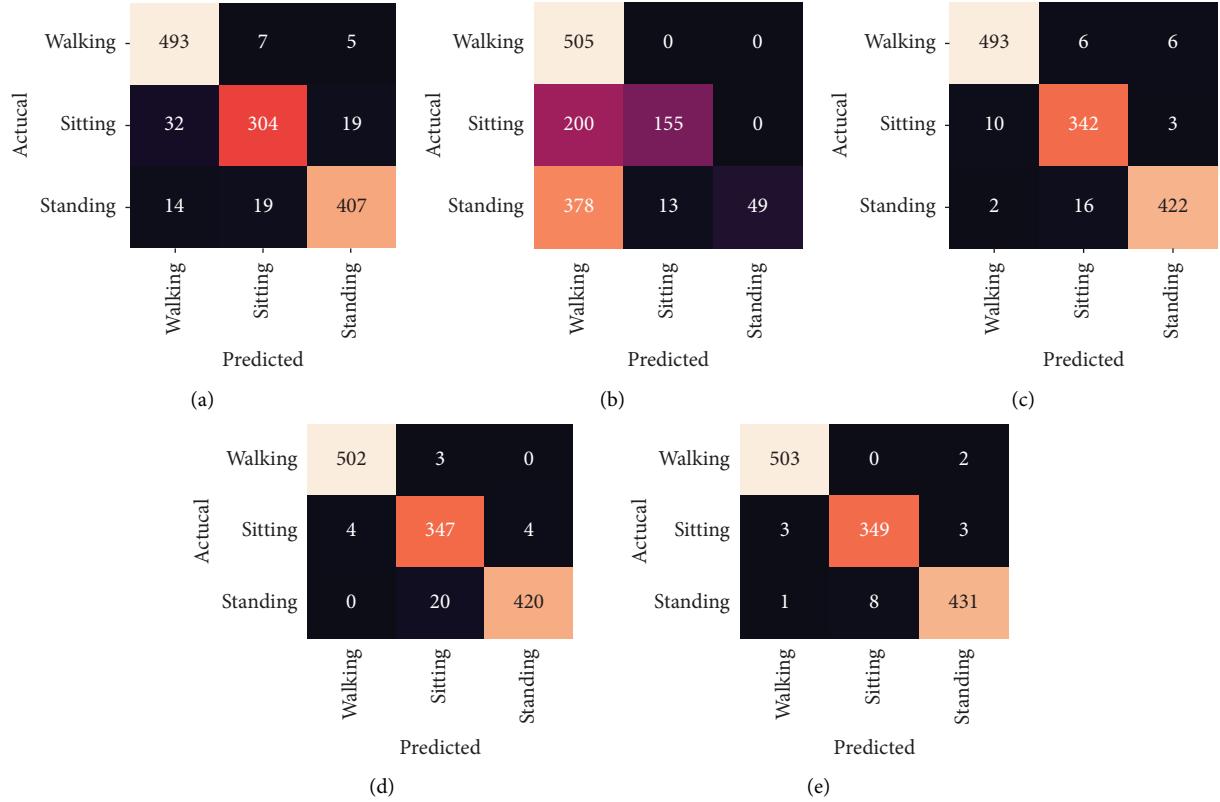


FIGURE 4: Confusion metrics for abnormal knee subjects. (a) CNN. (b) LSTM. (c) GRU. (d) CNN-LSTM. (e) CNN-GRU.

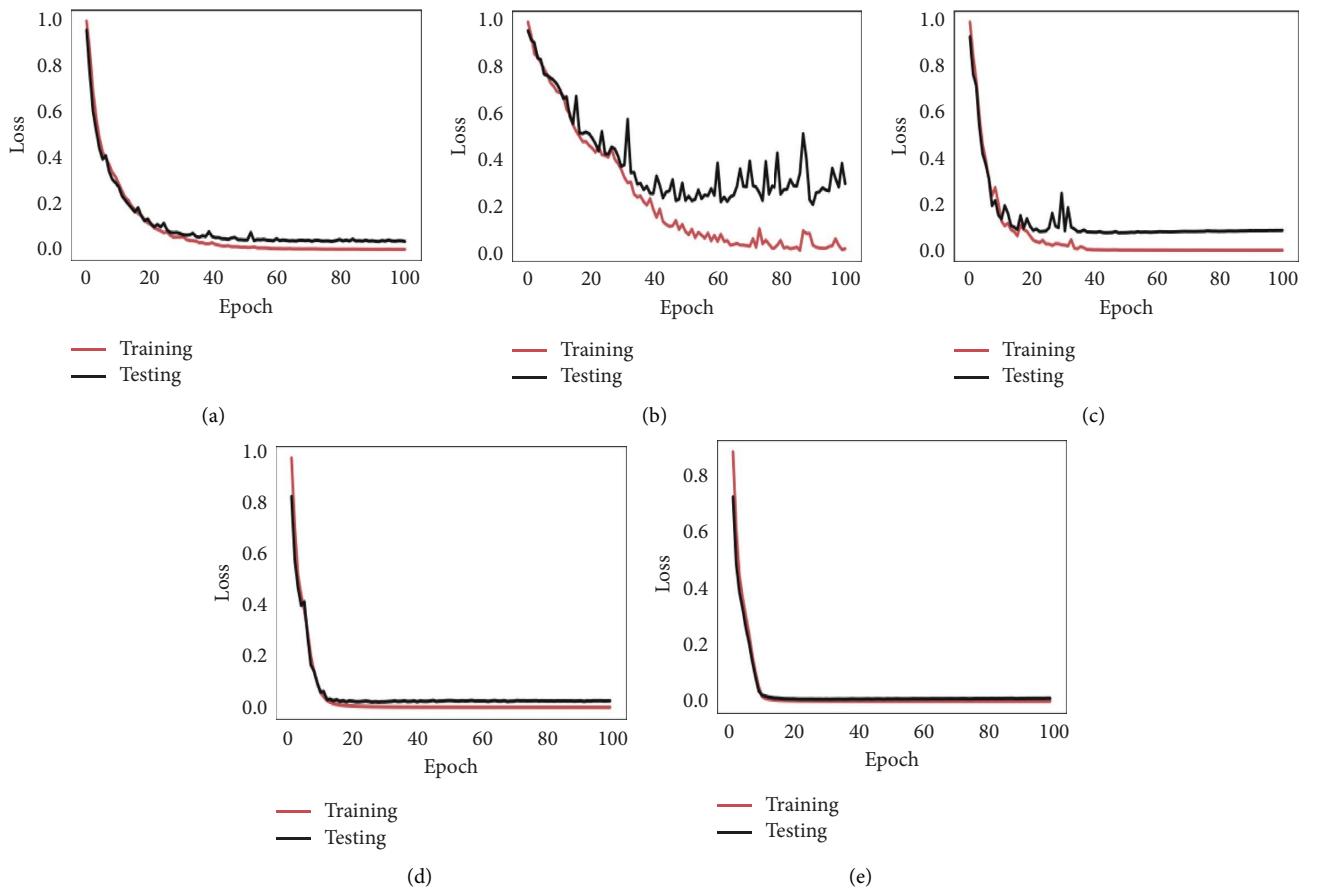


FIGURE 5: Loss vs. epoch curve for healthy subjects. (a) CNN. (b) LSTM. (c) GRU. (d) CNN-LSTM. (e) CNN-GRU.

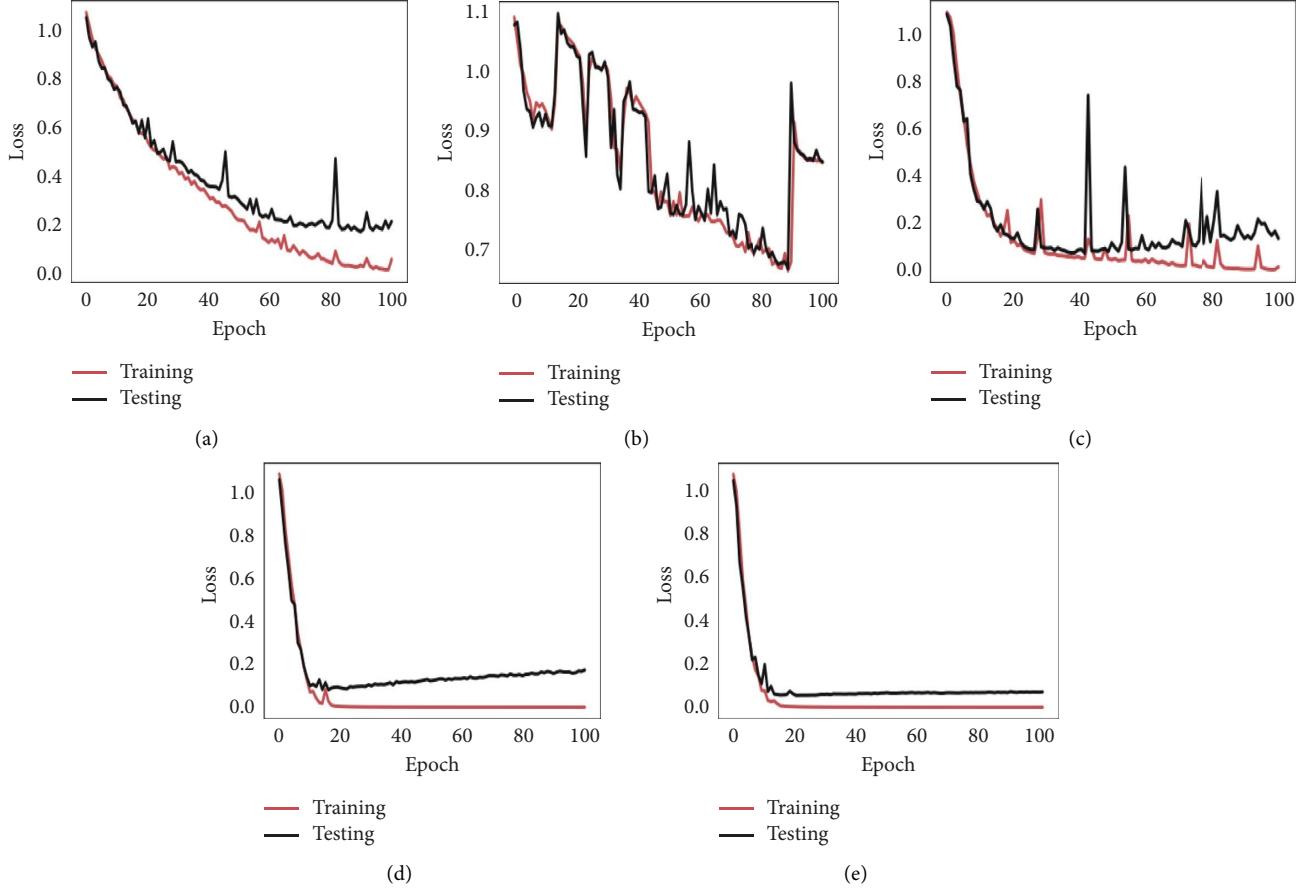


FIGURE 6: Loss vs. epoch curve for abnormal knee subjects.

TABLE 5: Comparison of the proposed method's performance with studies using same dataset (in %).

Approach	Subject	Accuracy		
		Walking	Sitting	Standing
EMD [41]	Healthy	64	67	69
MEMD [41]	Healthy	73	79	82
NA-EMD [41]	Healthy	79	83	83
MP-ANN [42]	Knee deformity	88	94	92
LRCN [43]	Healthy	98.2	97.7	98.4
	Knee deformity	92.8	92.3	92.2
ICA-EBM [36]	Healthy	96.0	96.2	96.2
	Knee deformity	86.6	86.4	85.5
WD-EEMD [34]	Healthy	85.11	88.70	93.50
	Knee deformity	98.86	96.38	96.77
V-1D-CNN [37]	Healthy	96.86	99.57	99.98
	Knee deformity	99.06	96.04	97.24
CNN-GRU (proposed)	Healthy	99.09	100	100
	Knee deformity	99.60	98.30	97.95

5. Conclusion and Future Scope

The research proposes the use of hybrid deep learning models CNN-LSTM and CNN-GRU to analyze sEMG data to detect lower limb activity in individuals with and without knee abnormalities in which CNN is used for temporal learning, and LSTM or GRU is used for sequence learning. To begin, the authors have used discrete wavelet denoising to denoise the original sEMG signal and then introduced overlapping windowing approaches for data segmentation to mitigate the issue of a small dataset. After that, deep learning models CNN, LSTM, GRU, CNN-LSTM, and CNN-GRU are implemented. In the proposed hybrid models, CNN is used for temporal learning, and LSTM or GRU is used for sequence learning. The proposed hybrid CNN-GRU model achieves classification accuracy of 99.86 and 98.69% and computational time of an epoch of 4.26 and 5.58 s for healthy and abnormal knee subjects, respectively. The results were compared with those obtained by individual models with the hybrid approach proving superior to them.

The sEMG dataset used in this study is limited to 22 individuals to evaluate the suggested approach. Thus, the strategy could be validated in the future by utilizing a large real-time dataset. The suggested methodology was evaluated

using an offline dataset; hence, future research could focus on clinical validation using a real-time dataset.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] S. Ranasinghe, F. Al Machot, and H. C. Mayr, "A review on applications of activity recognition systems with regard to performance and evaluation," *International Journal of Distributed Sensor Networks*, vol. 12, no. 8, Article ID 155014771666552, 2016.
- [2] V. B. Semwal, N. Gaud, P. Lalwani, V. Bijalwan, and A. K. Alok, "Pattern identification of different human joints for different human walking styles using inertial measurement unit (IMU) sensor," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 1149–1169, 2022.
- [3] X. Liu, L. Liu, S. J. Simske, and J. Liu, "Human daily activity recognition for healthcare using wearable and visual sensing data," in *Proceedings of the IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 24–31, IEEE, Chicago, IL, USA, 2016, October.
- [4] V. B. Semwal, A. Gupta, and P. Lalwani, "An optimized hybrid deep learning model using ensemble learning approach for human walking activities recognition," *The Journal of Supercomputing*, vol. 77, no. 11, Article ID 12256, 2021.
- [5] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.
- [6] A. Vijayvargiya, B. Singh, R. Kumar, and J. M. R. S. Tavares, "Human lower limb activity recognition techniques, databases, challenges and its applications using sEMG signal: an overview," *Biomedical Engineering Letters*, vol. 12, no. 4, pp. 343–358, 2022.
- [7] B. S. Yang and S. T. Liao, "Fall detecting using inertial and electromyographic sensors," in *Proceedings of the 36th Annual Meeting of the American Society of Biomechanics*, pp. 15–18, Gainesville, FL, USA, 2012, August.
- [8] J. Cheng, X. Chen, and M. Shen, "A framework for daily activity monitoring and fall detection based on surface electromyography and accelerometer signals," *IEEE journal of biomedical and health informatics*, vol. 17, no. 1, pp. 38–45, 2012.
- [9] D. Farina and F. Negro, "Accessing the neural drive to muscle and translation to neurorehabilitation technologies," *IEEE Reviews in biomedical engineering*, vol. 5, pp. 3–14, 2012.
- [10] N. Nazmi, M. A. Abdul Rahman, S. I. Yamamoto, S. A. Ahmad, H. Zamzuri, and S. A. Mazlan, "A review of classification techniques of EMG signals during isotonic and isometric contractions," *Sensors*, vol. 16, no. 8, p. 1304, 2016.
- [11] S. K. Au, P. Bonato, and H. Herr, "An EMG-position controlled system for an active ankle-foot prosthesis: an initial experimental study," in *Proceedings of the 9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005*, pp. 375–379, IEEE, Chicago, IL, USA, 2005, June.
- [12] A. Vijayvargiya, P. L. Singh, S. M. Verma, R. Kumar, and S. Bansal, "Performance comparison analysis of different classifier for early detection of knee osteoarthritis," in *Sensors for Health Monitoring* Academic Press, Massachusetts, MA, USA, 2019.
- [13] L. Burkow-Heikkinen, "Non-invasive physiological monitoring of exercise and fitness," *Neurological Research*, vol. 33, no. 1, pp. 3–17, 2011.
- [14] K. Kiguchi, T. Tanaka, and T. Fukuda, "Neuro-fuzzy control of a robotic exoskeleton with EMG signals," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, pp. 481–490, 2004.
- [15] V. Krasin, V. Gandhi, Z. Yang, and M. Karamanoglu, "EMG based elbow joint powered exoskeleton for biceps brachii strength augmentation," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, IEEE, Killarney, Ireland, 2015, July.
- [16] K. Sharmila, T. V. Sarath, and K. I. Ramachandran, "EMG controlled low cost prosthetic arm," in *Proceedings of the IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, pp. 169–172, IEEE, Mangalore, India, 2016, August.
- [17] S. Pancholi and A. M. Joshi, "Electromyography-based hand gesture recognition system for upper limb amputees," *IEEE Sensors Letters*, vol. 3, no. 3, pp. 1–4, 2019.
- [18] A. Vijayvargiya, P. Singh, R. Kumar, and N. Dey, "Hardware implementation for lower limb surface EMG measurement and analysis using explainable AI for activity recognition," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–9, 2022.
- [19] S. Cai, Y. Chen, S. Huang et al., "SVM-based classification of sEMG signals for upper-limb self-rehabilitation training," *Frontiers in Neurorobotics*, vol. 13, p. 31, 2019.
- [20] C. Souit, D. S. Coelho, M. Szylit, F. Camargo-Junior, M. P. C. Junior, and A. Forner-Cordero, "Design of a lower limb exoskeleton for experimental research on gait control," in *Proceedings of the 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pp. 1098–1103, IEEE, Singapore, 2016, June.
- [21] P. K. Shukla, A. Vijayvargiya, and R. Kumar, "Human activity recognition using accelerometer and gyroscope data from smartphones," in *Proceedings of the International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*, pp. 1–6, IEEE, Lakshmangarh, India, 2020, February.
- [22] U. S. L. G. Da Silva, H. A. Villagra, L. L. Oliva, and N. F. Marconi, "EMG activity of upper limb on spinal cord injury individuals during whole-body vibration," *Physiology International (Acta Physiologica Hungarica)*, vol. 103, no. 3, pp. 361–367, 2016.
- [23] A. Vijayvargiya, C. Prakash, R. Kumar, S. Bansal, and J. M. R. S. Tavares, "Human knee abnormality detection from imbalanced sEMG data," *Biomedical Signal Processing and Control*, vol. 66, Article ID 102406, 2021.
- [24] Ö. F. Ertuğrul, Y. Kaya, and R. Tekin, "A novel approach for SEMG signal classification with adaptive local binary patterns," *Medical, & Biological Engineering & Computing*, vol. 54, no. 7, pp. 1137–1146, 2016.
- [25] V. P. Gurupur, S. A. Kulkarni, X. Liu, U. Desai, and A. Nasir, "Analysing the power of deep learning techniques over the traditional methods using medicare utilisation and provider data," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 31, no. 1, pp. 99–115, 2019.
- [26] N. Dua, S. N. Singh, V. B. Semwal, and S. K. Challa, "Inception inspired CNN-GRU hybrid network for human activity recognition," *Multimedia Tools and Applications*, pp. 1–35, 2022.

- [27] R. Jain, V. B. Semwal, and P. Kaushik, "Deep ensemble learning approach for lower extremity activities recognition using wearable sensors," *Expert Systems*, vol. 39, no. 6, Article ID e12743, 2022.
- [28] N. Dua, S. N. Singh, and V. B. Semwal, "Multi-input CNN-GRU based human activity recognition using wearable sensors," *Computing*, vol. 103, no. 7, pp. 1461–1478, 2021.
- [29] S. K. Challa, A. Kumar, and V. B. Semwal, "A multibranch CNN-BiLSTM model for human activity recognition using wearable sensor data," *The Visual Computer*, pp. 1–15, 2021.
- [30] A. Vijayvargiya, B. Singh, N. Kumari, and R. Kumar, "sEMG-based deep learning framework for the automatic detection of knee abnormality," *Signal, Image and Video Processing*, pp. 1–9, 2022.
- [31] O. F. A. Sanchez, J. L. R. Sotelo, M. H. Gonzales, and G. A. M. Hernandez, "EMG dataset in lower limb data set," *UCI machine learning repository*, vol. 2, 2014.
- [32] R. H. Chowdhury, M. B. Reaz, M. A. B. M. Ali, A. A. Bakar, K. Chellappan, and T. G. Chang, "Surface electromyography signal processing and classification techniques," *Sensors*, vol. 13, no. 9, pp. 12431–12466, 2013.
- [33] U. Desai, R. J. Martis, C. G. Nayak, K. Sarika, and G. Seshikala, "Machine intelligent diagnosis of ECG for arrhythmia classification using DWT, ICA and SVM techniques," in *Proceedings of the Annual IEEE India Conference (INDICON)*, pp. 1–4, IEEE, New Delhi, India, 2015, December.
- [34] A. Vijayvargiya, V. Gupta, R. Kumar, N. Dey, and J. M. R. S. Tavares, "A hybrid WD-EEMD sEMG feature extraction technique for lower limb activity recognition," *IEEE Sensors Journal*, vol. 21, no. 18, Article ID 20431, 2021.
- [35] U. Desai, C. G. Nayak, and R. J. Seshikala, "Automated diagnosis of Coronary Artery Disease using pattern recognition approach," in *Proceedings of the 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 434–437, IEEE, Jeju, Korea (South), 2017 July.
- [36] G. R. Naik, S. E. Selvan, S. P. Arjunan et al., "An ICA-EBM-based sEMG classifier for recognizing lower limb movements in individuals with and without knee pathology," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 3, pp. 675–686, 2018.
- [37] A. Vijayvargiya, R. Kumar, R. Kumar, and N. Dey, "Voting-based 1D CNN model for human lower limb activity recognition using sEMG signal," *Physical and Engineering Sciences in Medicine*, vol. 44, no. 4, pp. 1297–1309, 2021.
- [38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [39] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.
- [40] A. Vijayvargiya, R. Kumar, N. Dey, and J. M. R. Tavares, "Comparative analysis of machine learning techniques for the classification of knee abnormality," in *Proceedings of the IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, pp. 1–6, IEEE, Greater Noida, India, 2020 October.
- [41] Y. Zhang, P. Xu, P. Li et al., "Noise-assisted multivariate empirical mode decomposition for multichannel EMG signals," *BioMedical Engineering Online*, vol. 16, no. 1, p. 107, 2017.
- [42] M. Herrera-González, G. A. Martínez-Hernández, J. L. Rodríguez-Sotelo, and O. Avilés-Sánchez, "Knee functional state classification using surface electromyographic and goniometric signals by means of artificial neural networks," *Ingeniería y Universidad*, vol. 19, no. 1, pp. 51–66, 2015.
- [43] A. Gautam, M. Panwar, D. Biswas, and A. Acharyya, "Myonet: a transfer-learning-based Ircn for lower limb movement recognition and knee joint angle prediction for remote monitoring of rehabilitation progress from sEMG," *IEEE J Transl Eng Health Med*, vol. 8, pp. 1–10, 2020.

Communicated by Ronald Williams

Long Short-Term Memory

Sepp Hochreiter

Fakultät für Informatik, Technische Universität München, 80290 München, Germany

Jürgen Schmidhuber

IDSIA, Corso Elvezia 36, 6900 Lugano, Switzerland

Learning to store information over extended time intervals by recurrent backpropagation takes a very long time, mostly because of insufficient, decaying error backflow. We briefly review Hochreiter's (1991) analysis of this problem, then address it by introducing a novel, efficient, gradient-based method called long short-term memory (LSTM). Truncating the gradient where this does not do harm, LSTM can learn to bridge minimal time lags in excess of 1000 discrete-time steps by enforcing constant error flow through constant error carousels within special units. Multiplicative gate units learn to open and close access to the constant error flow. LSTM is local in space and time; its computational complexity per time step and weight is $O(1)$. Our experiments with artificial data involve local, distributed, real-valued, and noisy pattern representations. In comparisons with real-time recurrent learning, back propagation through time, recurrent cascade correlation, Elman nets, and neural sequence chunking, LSTM leads to many more successful runs, and learns much faster. LSTM also solves complex, artificial long-time-lag tasks that have never been solved by previous recurrent network algorithms.

1 Introduction

In principle, recurrent networks can use their feedback connections to store representations of recent input events in the form of activations (short-term memory, as opposed to long-term memory embodied by slowly changing weights). This is potentially significant for many applications, including speech processing, non-Markovian control, and music composition (Mozer, 1992). The most widely used algorithms for learning what to put in short-term memory, however, take too much time or do not work well at all, especially when minimal time lags between inputs and corresponding teacher signals are long. Although theoretically fascinating, existing methods do not provide clear practical advantages over, say, backpropagation in feed-forward nets with limited time windows. This article reviews an analysis of the problem and suggests a remedy.

The problem. With conventional backpropagation through time (BPTT; Williams & Zipser, 1992; Werbos, 1988) or real-time recurrent learning (RTRL; Robinson & Fallside, 1987), error signals flowing backward in time tend to (1) blow up or (2) vanish; the temporal evolution of the backpropagated error exponentially depends on the size of the weights (Hochreiter, 1991). Case 1 may lead to oscillating weights; in case 2, learning to bridge long time lags takes a prohibitive amount of time or does not work at all (see section 3).

This article presents long short-term memory (LSTM), a novel recurrent network architecture in conjunction with an appropriate gradient-based learning algorithm. LSTM is designed to overcome these error backflow problems. It can learn to bridge time intervals in excess of 1000 steps even in case of noisy, incompressible input sequences, without loss of short-time-lag capabilities. This is achieved by an efficient, gradient-based algorithm for an architecture enforcing constant (thus, neither exploding nor vanishing) error flow through internal states of special units (provided the gradient computation is truncated at certain architecture-specific points; this does not affect long-term error flow, though).

Section 2 briefly reviews previous work. Section 3 begins with an outline of the detailed analysis of vanishing errors due to Hochreiter (1991). It then introduces a naive approach to constant error backpropagation for didactic purposes and highlights its problems concerning information storage and retrieval. These problems lead to the LSTM architecture described in section 4. Section 5 presents numerous experiments and comparisons with competing methods. LSTM outperforms them and also learns to solve complex, artificial tasks no other recurrent net algorithm has solved. Section 6 discusses LSTM's limitations and advantages. The appendix contains a detailed description of the algorithm (A.1) and explicit error flow formulas (A.2).

2 Previous Work

This section focuses on recurrent nets with time-varying inputs (as opposed to nets with stationary inputs and fixed-point-based gradient calculations; e.g., Almeida, 1987; Pineda, 1987).

2.1 Gradient-Descent Variants. The approaches of Elman (1988), Fahlman (1991), Williams (1989), Schmidhuber (1992a), Pearlmutter (1989), and many of the related algorithms in Pearlmutter's comprehensive overview (1995) suffer from the same problems as BPTT and RTRL (see sections 1 and 3).

2.2 Time Delays. Other methods that seem practical for short time lags only are time-delay neural networks (Lang, Waibel, & Hinton, 1990) and Plate's method (Plate, 1993), which updates unit activations based on a

weighted sum of old activations (see also de Vries & Principe, 1991). Lin et al. (1996) propose variants of time-delay networks called NARX networks.

2.3 Time Constants. To deal with long time lags, Mozer (1992) uses time constants influencing changes of unit activations (deVries and Principe's 1991 approach may in fact be viewed as a mixture of time-delay neural networks and time constants). For long time lags, however, the time constants need external fine tuning (Mozer, 1992). Sun, Chen, and Lee's alternative approach (1993) updates the activation of a recurrent unit by adding the old activation and the (scaled) current net input. The net input, however, tends to perturb the stored information, which makes long-term storage impractical.

2.4 Ring's Approach. Ring (1993) also proposed a method for bridging long time lags. Whenever a unit in his network receives conflicting error signals, he adds a higher-order unit influencing appropriate connections. Although his approach can sometimes be extremely fast, to bridge a time lag involving 100 steps may require the addition of 100 units. Also, Ring's net does not generalize to unseen lag durations.

2.5 Bengio et al.'s Approach. Bengio, Simard, and Frasconi (1994) investigate methods such as simulated annealing, multigrid random search, time-weighted pseudo-Newton optimization, and discrete error propagation. Their "latch" and "two-sequence" problems are very similar to problem 3a in this article with minimal time lag 100 (see Experiment 3). Bengio and Frasconi (1994) also propose an expectation-maximization approach for propagating targets. With n so-called state networks, at a given time, their system can be in one of only n different states. (See also the beginning of section 5.) But to solve continuous problems such as the adding problem (section 5.4), their system would require an unacceptable number of states (i.e., state networks).

2.6 Kalman Filters. Puskorius and Feldkamp (1994) use Kalman filter techniques to improve recurrent net performance. Since they use "a derivative discount factor imposed to decay exponentially the effects of past dynamic derivatives," there is no reason to believe that their Kalman filter-trained recurrent networks will be useful for very long minimal time lags.

2.7 Second Order Nets. We will see that LSTM uses multiplicative units (MUs) to protect error flow from unwanted perturbations. It is not the first recurrent net method using MUs, though. For instance, Watrous and Kuhn (1992) use MUs in second-order nets. There are some differences from LSTM: (1) Watrous and Kuhn's architecture does not enforce constant error flow and is not designed to solve long-time-lag problems; (2) it has fully connected second-order sigma-pi units, while the LSTM architecture's MUs

are used only to gate access to constant error flow; and (3) Watrous and Kuhn's algorithm costs $O(W^2)$ operations per time step, ours only $O(W)$, where W is the number of weights. See also Miller and Giles (1993) for additional work on MUs.

2.8 Simple Weight Guessing. To avoid long-time-lag problems of gradient-based approaches, we may simply randomly initialize all network weights until the resulting net happens to classify all training sequences correctly. In fact, recently we discovered (Schmidhuber & Hochreiter, 1996; Hochreiter & Schmidhuber, 1996, 1997) that simple weight guessing solves many of the problems in Bengio et al. (1994), Bengio and Frasconi (1994), Miller and Giles (1993), and Lin et al. (1996) faster than the algorithms these authors proposed. This does not mean that weight guessing is a good algorithm. It just means that the problems are very simple. More realistic tasks require either many free parameters (e.g., input weights) or high weight precision (e.g., for continuous-valued parameters), such that guessing becomes completely infeasible.

2.9 Adaptive Sequence Chunkers. Schmidhuber's hierarchical chunker systems (1992b, 1993) do have a capability to bridge arbitrary time lags, but only if there is local predictability across the subsequences causing the time lags (see also Mozer, 1992). For instance, in his postdoctoral thesis, Schmidhuber (1993) uses hierarchical recurrent nets to solve rapidly certain grammar learning tasks involving minimal time lags in excess of 1000 steps. The performance of chunker systems, however, deteriorates as the noise level increases and the input sequences become less compressible. LSTM does not suffer from this problem.

3 Constant Error Backpropagation

3.1 Exponentially Decaying Error

3.1.1 Conventional BPTT (e.g., Williams & Zipser, 1992). Output unit k 's target at time t is denoted by $d_k(t)$. Using mean squared error, k 's error signal is

$$\vartheta_k(t) = f'_k(\text{net}_k(t))(d_k(t) - y^k(t)),$$

where

$$y^i(t) = f_i(\text{net}_i(t))$$

is the activation of a noninput unit i with differentiable activation function f_i ,

$$\text{net}_i(t) = \sum_j w_{ij} y^j(t-1)$$

is unit i 's current net input, and w_{ij} is the weight on the connection from unit j to i . Some nonoutput unit j 's backpropagated error signal is

$$\vartheta_j(t) = f'_j(\text{net}_j(t)) \sum_i w_{ij} \vartheta_i(t+1).$$

The corresponding contribution to w_{jl} 's total weight update is $\alpha \vartheta_j(t) y^l(t-1)$, where α is the learning rate and l stands for an arbitrary unit connected to unit j .

3.1.2 Outline of Hochreiter's Analysis (1991, pp. 19–21). Suppose we have a fully connected net whose noninput unit indices range from 1 to n . Let us focus on local error flow from unit u to unit v (later we will see that the analysis immediately extends to global error flow). The error occurring at an arbitrary unit u at time step t is propagated back into time for q time steps, to an arbitrary unit v . This will scale the error by the following factor:

$$\frac{\partial \vartheta_v(t-q)}{\partial \vartheta_u(t)} = \begin{cases} f'_v(\text{net}_v(t-1)) w_{uv} & q=1 \\ f'_v(\text{net}_v(t-q)) \sum_{l=1}^n \frac{\partial \vartheta_l(t-q+1)}{\partial \vartheta_u(t)} w_{lv} & q>1 \end{cases}. \quad (3.1)$$

With $l_q = v$ and $l_0 = u$, we obtain:

$$\frac{\partial \vartheta_v(t-q)}{\partial \vartheta_u(t)} = \sum_{l_1=1}^n \dots \sum_{l_{q-1}=1}^n \prod_{m=1}^q f'_{l_m}(\text{net}_{l_m}(t-m)) w_{l_m l_{m-1}} \quad (3.2)$$

(proof by induction). The sum of the n^{q-1} terms $\prod_{m=1}^q f'_{l_m}(\text{net}_{l_m}(t-m)) w_{l_m l_{m-1}}$ determines the total error backflow (note that since the summation terms may have different signs, increasing the number of units n does not necessarily increase error flow).

3.1.3 Intuitive Explanation of Equation 3.2.

If

$$|f'_{l_m}(\text{net}_{l_m}(t-m)) w_{l_m l_{m-1}}| > 1.0$$

for all m (as can happen, e.g., with linear f_{l_m}), then the largest product increases exponentially with q . That is, the error blows up, and conflicting error signals arriving at unit v can lead to oscillating weights and unstable learning (for error blowups or bifurcations, see also Pineda, 1988; Baldi & Pineda, 1991; Doya, 1992). On the other hand, if

$$|f'_{l_m}(\text{net}_{l_m}(t-m)) w_{l_m l_{m-1}}| < 1.0$$

for all m , then the largest product decreases exponentially with q . That is, the error vanishes, and nothing can be learned in acceptable time.

If f_{l_m} is the logistic sigmoid function, then the maximal value of f'_{l_m} is 0.25. If $y^{l_{m-1}}$ is constant and not equal to zero, then $|f'_{l_m}(\text{net}_{l_m})w_{l_m l_{m-1}}|$ takes on maximal values where

$$w_{l_m l_{m-1}} = \frac{1}{y^{l_{m-1}}} \coth\left(\frac{1}{2}\text{net}_{l_m}\right),$$

goes to zero for $|w_{l_m l_{m-1}}| \rightarrow \infty$, and is less than 1.0 for $|w_{l_m l_{m-1}}| < 4.0$ (e.g., if the absolute maximal weight value w_{\max} is smaller than 4.0). Hence with conventional logistic sigmoid activation functions, the error flow tends to vanish as long as the weights have absolute values below 4.0, especially in the beginning of the training phase. In general, the use of larger initial weights will not help, though, as seen above, for $|w_{l_m l_{m-1}}| \rightarrow \infty$ the relevant derivative goes to zero “faster” than the absolute weight can grow (also, some weights will have to change their signs by crossing zero). Likewise, increasing the learning rate does not help either; it will not change the ratio of long-range error flow and short-range error flow. BPTT is too sensitive to recent distractions. (A very similar, more recent analysis was presented by Bengio et al., 1994.)

3.1.4 Global Error Flow. The local error flow analysis above immediately shows that global error flow vanishes too. To see this, compute

$$\sum_{u: u \text{ output unit}} \frac{\partial \vartheta_v(t-q)}{\partial \vartheta_u(t)}.$$

3.1.5 Weak Upper Bound for Scaling Factor. The following, slightly extended vanishing error analysis also takes n , the number of units, into account. For $q > 1$, equation 3.2 can be rewritten as

$$(W_{u^T})^T F(t-1) \prod_{m=2}^{q-1} (WF(t-m)) W_v f'_v(\text{net}_v(t-q)),$$

where the weight matrix W is defined by $[W]_{ij} := w_{ji}$, v 's outgoing weight vector W_v is defined by $[W_v]_i := [W]_{iv} = w_{iv}$, u 's incoming weight vector W_{u^T} is defined by $[W_{u^T}]_i := [W]_{ui} = w_{ui}$, and for $m = 1, \dots, q$, $F(t-m)$ is the diagonal matrix of first-order derivatives defined as $[F(t-m)]_{ij} := 0$ if $i \neq j$, and $[F(t-m)]_{ij} := f'_i(\text{net}_i(t-m))$ otherwise. Here T is the transposition operator, $[A]_{ij}$ is the element in the i th column and j th row of matrix A , and $[x]_i$ is the i th component of vector x .

Using a matrix norm $\|\cdot\|_A$ compatible with vector norm $\|\cdot\|_x$, we define

$$f'_{\max} := \max_{m=1, \dots, q} \{\|F(t-m)\|_A\}.$$

For $\max_{i=1, \dots, n} \{|x_i|\} \leq \|x\|_x$ we get $|x^T y| \leq n \|x\|_x \|y\|_x$. Since

$$|f'_v(\text{net}_v(t-q))| \leq \|F(t-q)\|_A \leq f'_{\max},$$

we obtain the following inequality:

$$\left| \frac{\partial \vartheta_v(t-q)}{\partial \vartheta_u(t)} \right| \leq n (f'_{\max})^q \|W_v\|_x \|W_{u^T}\|_x \|W\|_A^{q-2} \leq n (f'_{\max} \|W\|_A)^q.$$

This inequality results from

$$\|W_v\|_x = \|We_v\|_x \leq \|W\|_A \|e_v\|_x \leq \|W\|_A$$

and

$$\|W_{u^T}\|_x = \|W^T e_u\|_x \leq \|W\|_A \|e_u\|_x \leq \|W\|_A,$$

where e_k is the unit vector whose components are 0 except for the k th component, which is 1. Note that this is a weak, extreme case upper bound; it will be reached only if all $\|F(t-m)\|_A$ take on maximal values, and if the contributions of all paths across which error flows back from unit u to unit v have the same sign. Large $\|W\|_A$, however, typically result in small values of $\|F(t-m)\|_A$, as confirmed by experiments (see, e.g., Hochreiter, 1991).

For example, with norms

$$\|W\|_A := \max_r \sum_s |w_{rs}|$$

and

$$\|x\|_x := \max_r |x_r|,$$

we have $f'_{\max} = 0.25$ for the logistic sigmoid. We observe that if

$$|w_{ij}| \leq w_{\max} < \frac{4.0}{n} \quad \forall i, j,$$

then $\|W\|_A \leq nw_{\max} < 4.0$ will result in exponential decay. By setting $\tau := (\frac{nw_{\max}}{4.0}) < 1.0$, we obtain

$$\left| \frac{\partial \vartheta_v(t-q)}{\partial \vartheta_u(t)} \right| \leq n(\tau)^q.$$

We refer to Hochreiter (1991) for additional results.

3.2 Constant Error Flow: Naive Approach.

3.2.1 A Single Unit. To avoid vanishing error signals, how can we achieve constant error flow through a single unit j with a single connection to itself? According to the rules above, at time t , j 's local error backflow is $\vartheta_j(t) = f'_j(\text{net}_j(t))\vartheta_j(t+1)w_{jj}$. To enforce constant error flow through j , we require

$$f'_j(\text{net}_j(t))w_{jj} = 1.0.$$

Note the similarity to Mozer's fixed time constant system (1992)—a time constant of 1.0 is appropriate for potentially infinite time lags.¹

3.2.2 The Constant Error Carousel. Integrating the differential equation above, we obtain

$$f_j(\text{net}_j(t)) = \frac{\text{net}_j(t)}{w_{jj}}$$

for arbitrary $\text{net}_j(t)$. This means f_j has to be linear, and unit j 's activation has to remain constant:

$$y_j(t+1) = f_j(\text{net}_j(t+1)) = f_j(w_{jj}y_j(t)) = y_j(t).$$

In the experiments, this will be ensured by using the identity function $f_j : f_j(x) = x, \forall x$, and by setting $w_{jj} = 1.0$. We refer to this as the constant error carousel (CEC). CEC will be LSTM's central feature (see section 4).

Of course, unit j will not only be connected to itself but also to other units. This invokes two obvious, related problems (also inherent in all other gradient-based approaches):

1. Input weight conflict: For simplicity, let us focus on a single additional input weight w_{ji} . Assume that the total error can be reduced by switching on unit j in response to a certain input and keeping it active for a long time (until it helps to compute a desired output). Provided i is nonzero, since the same incoming weight has to be used for both storing certain inputs and ignoring others, w_{ji} will often receive conflicting weight update signals during this time (recall that j is linear). These signals will attempt to make w_{ji} participate in (1) storing the input (by switching on j) and (2) protecting the input (by preventing j from being switched off by irrelevant later inputs). This conflict makes learning difficult and calls for a more context-sensitive mechanism for controlling write operations through input weights.
2. Output weight conflict: Assume j is switched on and currently stores some previous input. For simplicity, let us focus on a single additional outgoing weight w_{kj} . The same w_{kj} has to be used for both retrieving j 's content at certain times and preventing j from disturbing k at other times. As long as unit j is nonzero, w_{kj} will attract conflicting weight update signals generated during sequence processing. These signals will attempt to make w_{kj} participate in accessing the information stored in j and—at different times—protecting unit k from being perturbed by j . For instance, with many tasks there are certain short-time-lag errors that can be reduced in early training stages. However,

¹ We do not use the expression "time constant" in the differential sense, as Pearlmutter (1995) does.

at later training stages, j may suddenly start to cause avoidable errors in situations that already seemed under control by attempting to participate in reducing more difficult long-time-lag errors. Again, this conflict makes learning difficult and calls for a more context-sensitive mechanism for controlling read operations through output weights.

Of course, input and output weight conflicts are not specific for long time lags; they occur for short time lags as well. Their effects, however, become particularly pronounced in the long-time-lag case. As the time lag increases, stored information must be protected against perturbation for longer and longer periods, and, especially in advanced stages of learning, more and more already correct outputs also require protection against perturbation.

Due to the problems set out, the naive approach does not work well except in the case of certain simple problems involving local input-output representations and nonrepeating input patterns (see Hochreiter, 1991; Silva, Amarel, Langlois, & Almeida, 1996). The next section shows how to do it right.

4 The Concept of Long Short-Term Memory

4.1 Memory Cells and Gate Units. To construct an architecture that allows for constant error flow through special, self-connected units without the disadvantages of the naive approach, we extend the CEC embodied by the self-connected, linear unit j from section 3.2 by introducing additional features. A multiplicative input gate unit is introduced to protect the memory contents stored in j from perturbation by irrelevant inputs, and a multiplicative output gate unit is introduced to protect other units from perturbation by currently irrelevant memory contents stored in j .

The resulting, more complex unit is called a memory cell (see Figure 1). The j th memory cell is denoted c_j . Each memory cell is built around a central linear unit with a fixed self-connection (the CEC). In addition to net_{c_j} , c_j gets input from a multiplicative unit out_j (the output gate), and from another multiplicative unit in_j (the input gate). in_j 's activation at time t is denoted by $y^{in_j}(t)$, out_j 's by $y^{out_j}(t)$. We have

$$y^{out_j}(t) = f_{out_j}(\text{net}_{out_j}(t)); \quad y^{in_j}(t) = f_{in_j}(\text{net}_{in_j}(t));$$

where

$$\text{net}_{out_j}(t) = \sum_u w_{out_j u} y^u(t-1),$$

and

$$\text{net}_{in_j}(t) = \sum_u w_{in_j u} y^u(t-1).$$

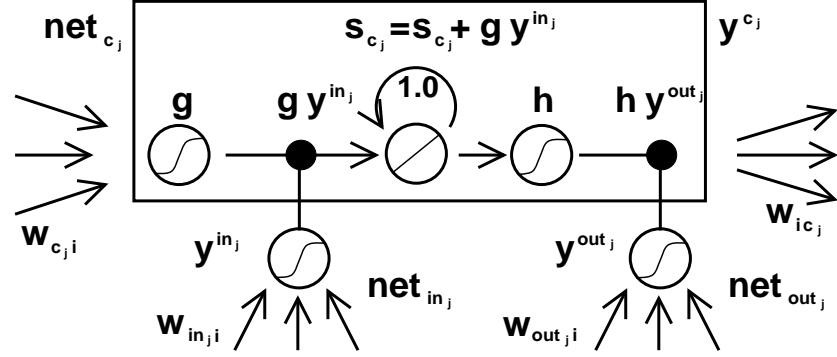


Figure 1: Architecture of memory cell c_j (the box) and its gate units in_j , out_j . The self-recurrent connection (with weight 1.0) indicates feedback with a delay of one time step. It builds the basis of the CEC. The gate units open and close access to CEC. See text and appendix A.1 for details.

We also have

$$net_{c_j}(t) = \sum_u w_{c_j u} y^u(t-1).$$

The summation indices u may stand for input units, gate units, memory cells, or even conventional hidden units if there are any (see section 4.3). All these different types of units may convey useful information about the current state of the net. For instance, an input gate (output gate) may use inputs from other memory cells to decide whether to store (access) certain information in its memory cell. There even may be recurrent self-connections like $w_{c_j c_j}$. It is up to the user to define the network topology. See Figure 2 for an example.

At time t , c_j 's output $y^{c_j}(t)$ is computed as

$$y^{c_j}(t) = y^{out_j}(t)h(s_{c_j}(t)),$$

where the internal state $s_{c_j}(t)$ is

$$s_{c_j}(0) = 0, s_{c_j}(t) = s_{c_j}(t-1) + y^{in_j}(t)g(net_{c_j}(t)) \text{ for } t > 0.$$

The differentiable function g squashes net_{c_j} ; the differentiable function h scales memory cell outputs computed from the internal state s_{c_j} .

4.2 Why Gate Units? To avoid input weight conflicts, in_j controls the error flow to memory cell c_j 's input connections $w_{c_j i}$. To circumvent c_j 's output weight conflicts, out_j controls the error flow from unit j 's output

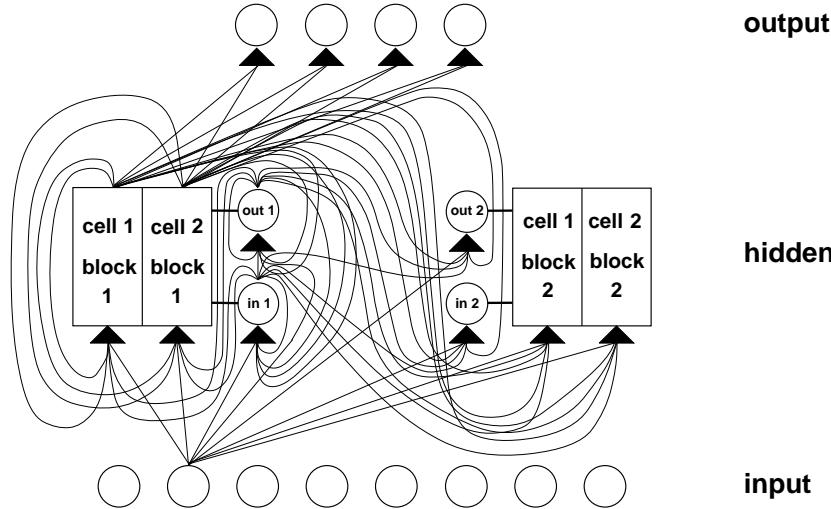


Figure 2: Example of a net with eight input units, four output units, and two memory cell blocks of size 2. in_1 marks the input gate, out_1 marks the output gate, and $cell1/block1$ marks the first memory cell of block 1. $cell1/block1$'s architecture is identical to the one in Figure 1, with gate units in_1 and out_1 (note that by rotating Figure 1 by 90 degrees anticlockwise, it will match with the corresponding parts of Figure 2). The example assumes dense connectivity: each gate unit and each memory cell sees all non-output units. For simplicity, however, outgoing weights of only one type of unit are shown for each layer. With the efficient, truncated update rule, error flows only through connections to output unit, and through fixed self-connections within cell blocks (not shown here; see Figure 1). Error flow is truncated once it “wants” to leave memory cells or gate units. Therefore, no connection shown above serves to propagate error back to the unit from which the connection originates (except for connections to output units), although the connections themselves are modifiable. That is why the truncated LSTM algorithm is so efficient, despite its ability to bridge very long time lags. See the text and the appendix for details. Figure 2 shows the architecture used for experiment 6a; only the bias of the noninput units is omitted.

connections. In other words, the net can use in_j to decide when to keep or override information in memory cell c_j and out_j to decide when to access memory cell c_j and when to prevent other units from being perturbed by c_j (see Figure 1).

Error signals trapped within a memory cell's CEC cannot change, but different error signals flowing into the cell (at different times) via its output gate may get superimposed. The output gate will have to learn which

errors to trap in its CEC by appropriately scaling them. The input gate will have to learn when to release errors, again by appropriately scaling them. Essentially the multiplicative gate units open and close access to constant error flow through CEC.

Distributed output representations typically do require output gates. Both gate types are not always necessary, though; one may be sufficient. For instance, in experiments 2a and 2b in section 5, it will be possible to use input gates only. In fact, output gates are not required in case of local output encoding; preventing memory cells from perturbing already learned outputs can be done by simply setting the corresponding weights to zero. Even in this case, however, output gates can be beneficial: they prevent the net's attempts at storing long-time-lag memories (which are usually hard to learn) from perturbing activations representing easily learnable short-time-lag memories. (This will prove quite useful in experiment 1, for instance.)

4.3 Network Topology. We use networks with one input layer, one hidden layer, and one output layer. The (fully) self-connected hidden layer contains memory cells and corresponding gate units (for convenience, we refer to both memory cells and gate units as being located in the hidden layer). The hidden layer may also contain conventional hidden units providing inputs to gate units and memory cells. All units (except for gate units) in all layers have directed connections (serve as inputs) to all units in the layer above (or to all higher layers; see experiments 2a and 2b).

4.4 Memory Cell Blocks. S memory cells sharing the same input gate and the same output gate form a structure called a memory cell block of size S . Memory cell blocks facilitate information storage. As with conventional neural nets, it is not so easy to code a distributed input within a single cell. Since each memory cell block has as many gate units as a single memory cell (namely, two), the block architecture can be even slightly more efficient. A memory cell block of size 1 is just a simple memory cell. In the experiments in section 5, we will use memory cell blocks of various sizes.

4.5 Learning. We use a variant of RTRL (e.g., Robinson & Fallside, 1987) that takes into account the altered, multiplicative dynamics caused by input and output gates. To ensure nondecaying error backpropagation through internal states of memory cells, as with truncated BPTT (e.g., Williams & Peng, 1990), errors arriving at memory cell net inputs (for cell c_j , this includes net_{c_j} , net_{in_j} , net_{out_j}) do not get propagated back further in time (although they do serve to change the incoming weights). Only within memory cells, are errors propagated back through previous internal states s_{c_j} .² To visualize

² For intracellular backpropagation in a quite different context, see also Doya and Yoshizawa (1989).

this, once an error signal arrives at a memory cell output, it gets scaled by output gate activation and h' . Then it is within the memory cell's CEC, where it can flow back indefinitely without ever being scaled. When it leaves the memory cell through the input gate and g , it is scaled once more by input gate activation and g' . It then serves to change the incoming weights before it is truncated (see the appendix for formulas).

4.6 Computational Complexity. As with Mozer's focused recurrent back-propagation algorithm (Mozer, 1989), only the derivatives $\partial s_{c_j} / \partial w_{il}$ need to be stored and updated. Hence the LSTM algorithm is very efficient, with an excellent update complexity of $O(W)$, where W the number of weights (see details in the appendix). Hence, LSTM and BPTT for fully recurrent nets have the same update complexity per time step (while RTRL's is much worse). Unlike full BPTT, however, LSTM is local in space and time:³ there is no need to store activation values observed during sequence processing in a stack with potentially unlimited size.

4.7 Abuse Problem and Solutions. In the beginning of the learning phase, error reduction may be possible without storing information over time. The network will thus tend to abuse memory cells, for example, as bias cells (it might make their activations constant and use the outgoing connections as adaptive thresholds for other units). The potential difficulty is that it may take a long time to release abused memory cells and make them available for further learning. A similar "abuse problem" appears if two memory cells store the same (redundant) information. There are at least two solutions to the abuse problem: (1) sequential network construction (e.g., Fahlman, 1991): a memory cell and the corresponding gate units are added to the network whenever the error stops decreasing (see experiment 2 in section 5), and (2) output gate bias: each output gate gets a negative initial bias, to push initial memory cell activations toward zero. Memory cells with more negative bias automatically get "allocated" later (see experiments 1, 3, 4, 5, and 6 in section 5).

4.8 Internal State Drift and Remedies. If memory cell c_j 's inputs are mostly positive or mostly negative, then its internal state s_j will tend to drift away over time. This is potentially dangerous, for the $h'(s_j)$ will then adopt very small values, and the gradient will vanish. One way to circumvent this problem is to choose an appropriate function h . But $h(x) = x$, for instance, has the disadvantage of unrestricted memory cell output range. Our simple

³ Following Schmidhuber (1989), we say that a recurrent net algorithm is *local in space* if the update complexity per time step and weight does not depend on network size. We say that a method is *local in time* if its storage requirements do not depend on input sequence length. For instance, RTRL is local in time but not in space. BPTT is local in space but not in time.

but effective way of solving drift problems at the beginning of learning is initially to bias the input gate i_{inj} toward zero. Although there is a trade-off between the magnitudes of $h'(s_j)$ on the one hand and of y^{inj} and f'_{inj} on the other, the potential negative effect of input gate bias is negligible compared to the one of the drifting effect. With logistic sigmoid activation functions, there appears to be no need for fine-tuning the initial bias, as confirmed by experiments 4 and 5 in section 5.4.

5 Experiments

Which tasks are appropriate to demonstrate the quality of a novel long-time-lag algorithm? First, minimal time lags between relevant input signals and corresponding teacher signals must be long for all training sequences. In fact, many previous recurrent net algorithms sometimes manage to generalize from very short training sequences to very long test sequences (see, e.g., Pollack, 1991). But a real long-time-lag problem does not have any short-time-lag exemplars in the training set. For instance, Elman's training procedure, BPTT, offline RTRL, online RTRL, and others fail miserably on real long-time-lag problems. (See, e.g., Hochreiter, 1991; Mozer, 1992.) A second important requirement is that the tasks should be complex enough such that they cannot be solved quickly by simple-minded strategies such as random weight guessing.

Recently we discovered (Schmidhuber & Hochreiter, 1996; Hochreiter & Schmidhuber, 1996, 1997) that many long-time-lag tasks used in previous work can be solved more quickly by simple random weight guessing than by the proposed algorithms. For instance, guessing solved a variant of Bengio and Frasconi's parity problem (1994) much faster⁴ than the seven methods tested by Bengio et al. (1994) and Bengio and Frasconi (1994). The same is true for some of Miller and Giles's problems (1993). Of course, this does not mean that guessing is a good algorithm. It just means that some previously used problems are not extremely appropriate to demonstrate the quality of previously proposed algorithms.

All our experiments (except experiment 1) involve long minimal time lags; there are no short-time-lag training exemplars facilitating learning. Solutions to most of our tasks are sparse in weight space. They require either many parameters and inputs or high weight precision, such that random weight guessing becomes infeasible.

We always use online learning (as opposed to batch learning) and logistic sigmoids as activation functions. For experiments 1 and 2, initial weights are chosen in the range $[-0.2, 0.2]$, for the other experiments in $[-0.1, 0.1]$. Training sequences are generated randomly according to the various task

⁴ Different input representations and different types of noise may lead to worse guessing performance (Yoshua Bengio, personal communication, 1996).

descriptions. In slight deviation from the notation in appendix A.1, each discrete time step of each input sequence involves three processing steps: (1) use current input to set the input units, (2) compute activations of hidden units (including input gates, output gates, memory cells), and (3) compute output unit activations. Except for experiments 1, 2a, and 2b, sequence elements are randomly generated online, and error signals are generated only at sequence ends. Net activations are reset after each processed input sequence.

For comparisons with recurrent nets taught by gradient descent, we give results only for RTRL, except for comparison 2a, which also includes BPTT. Note, however, that untruncated BPTT (see, e.g., Williams & Peng, 1990) computes exactly the same gradient as offline RTRL. With long-time-lag problems, offline RTRL (or BPTT) and the online version of RTRL (no activation resets, online weight changes) lead to almost identical, negative results (as confirmed by additional simulations in Hochreiter, 1991; see also Mozer, 1992). This is because offline RTRL, online RTRL, and full BPTT all suffer badly from exponential error decay.

Our LSTM architectures are selected quite arbitrarily. If nothing is known about the complexity of a given problem, a more systematic approach would be to: start with a very small net consisting of one memory cell. If this does not work, try two cells, and so on. Alternatively, use sequential network construction (e.g., Fahlman, 1991).

Following is an outline of the experiments:

- Experiment 1 focuses on a standard benchmark test for recurrent nets: the embedded Reber grammar. Since it allows for training sequences with short time lags, it is not a long-time-lag problem. We include it because it provides a nice example where LSTM's output gates are truly beneficial, and it is a popular benchmark for recurrent nets that has been used by many authors. We want to include at least one experiment where conventional BPTT and RTRL do not fail completely (LSTM, however, clearly outperforms them). The embedded Reber grammar's minimal time lags represent a border case in the sense that it is still possible to learn to bridge them with conventional algorithms. Only slightly longer minimal time lags would make this almost impossible. The more interesting tasks in our article, however, are those that RTRL, BPTT, and others cannot solve at all.
- Experiment 2 focuses on noise-free and noisy sequences involving numerous input symbols distracting from the few important ones. The most difficult task (task 2c) involves hundreds of distractor symbols at random positions and minimal time lags of 1000 steps. LSTM solves it; BPTT and RTRL already fail in case of 10-step minimal time lags (see also Hochreiter, 1991; Mozer, 1992). For this reason RTRL and BPTT are omitted in the remaining, more complex experiments, all of which involve much longer time lags.

- Experiment 3 addresses long-time-lag problems with noise and signal on the same input line. Experiments 3a and 3b focus on Bengio et al.'s 1994 two-sequence problem. Because this problem can be solved quickly by random weight guessing, we also include a far more difficult two-sequence problem (experiment 3c), which requires learning real-valued, conditional expectations of noisy targets, given the inputs.
- Experiments 4 and 5 involve distributed, continuous-valued input representations and require learning to store precise, real values for very long time periods. Relevant input signals can occur at quite different positions in input sequences. Again minimal time lags involve hundreds of steps. Similar tasks never have been solved by other recurrent net algorithms.
- Experiment 6 involves tasks of a different complex type that also has not been solved by other recurrent net algorithms. Again, relevant input signals can occur at quite different positions in input sequences. The experiment shows that LSTM can extract information conveyed by the temporal order of widely separated inputs.

Section 5.7 provides a detailed summary of experimental conditions in two tables for reference.

5.1 Experiment 1: Embedded Reber Grammar.

5.1.1 Task. Our first task is to learn the embedded Reber grammar (Smith & Zipser, 1989; Cleeremans, Servan-Schreiber, & McClelland, 1989; Fahlman, 1991). Since it allows for training sequences with short time lags (of as few as nine steps), it is not a long-time-lag problem. We include it for two reasons: (1) it is a popular recurrent net benchmark used by many authors, and we wanted to have at least one experiment where RTRL and BPTT do not fail completely, and (2) it shows nicely how output gates can be beneficial.

Starting at the left-most node of the directed graph in Figure 3, symbol strings are generated sequentially (beginning with the empty string) by following edges—and appending the associated symbols to the current string—until the right-most node is reached (the Reber grammar substrings are analogously generated from Figure 4). Edges are chosen randomly if there is a choice (probability: 0.5). The net's task is to read strings, one symbol at a time, and to predict the next symbol (error signals occur at every time step). To predict the symbol before last, the net has to remember the second symbol.

5.1.2 Comparison. We compare LSTM to Elman nets trained by Elman's training procedure (ELM) (results taken from Cleeremans et al., 1989), Fahlman's recurrent cascade-correlation (RCC) (results taken from Fahlman,

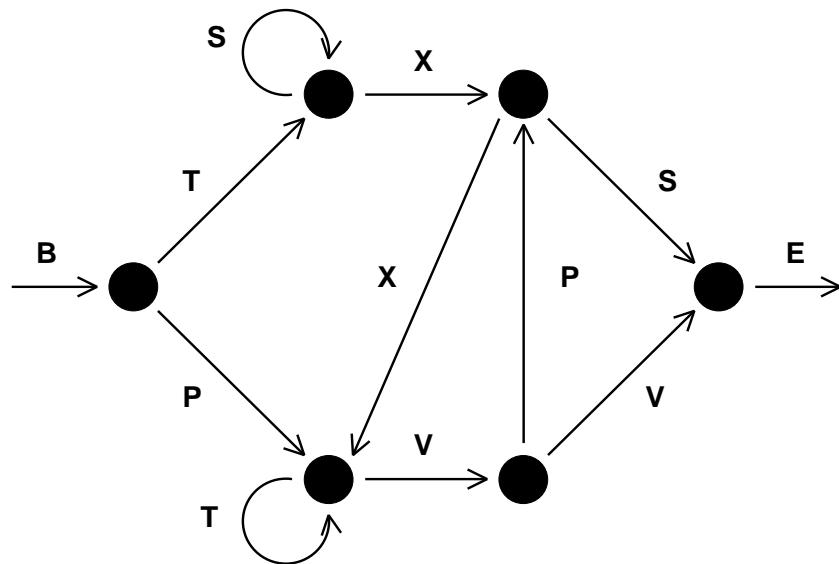


Figure 3: Transition diagram for the Reber grammar.

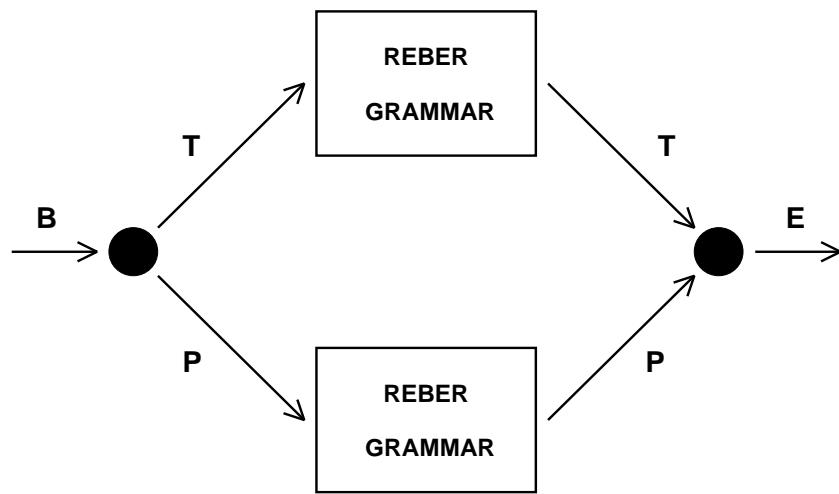


Figure 4: Transition diagram for the embedded Reber grammar. Each box represents a copy of the Reber grammar (see Figure 3).

1991), and RTRL (results taken from Smith & Zipser, 1989), where only the few successful trials are listed). Smith and Zipser actually make the task easier by increasing the probability of short-time-lag exemplars. We did not do this for LSTM.

5.1.3 Training/Testing. We use a local input-output representation (seven input units, seven output units). Following Fahrlman, we use 256 training strings and 256 separate test strings. The training set is generated randomly; training exemplars are picked randomly from the training set. Test sequences are generated randomly, too, but sequences already used in the training set are not used for testing. After string presentation, all activations are reinitialized with zeros. A trial is considered successful if all string symbols of all sequences in both test set and training set are predicted correctly—that is, if the output unit(s) corresponding to the possible next symbol(s) is(are) always the most active ones.

5.1.4 Architectures. Architectures for RTRL, ELM, and RCC are reported in the references listed above. For LSTM, we use three (four) memory cell blocks. Each block has two (one) memory cells. The output layer's only incoming connections originate at memory cells. Each memory cell and each gate unit receives incoming connections from all memory cells and gate units (the hidden layer is fully connected; less connectivity may work as well). The input layer has forward connections to all units in the hidden layer. The gate units are biased. These architecture parameters make it easy to store at least three input signals (architectures 3-2 and 4-1 are employed to obtain comparable numbers of weights for both architectures: 264 for 4-1 and 276 for 3-2). Other parameters may be appropriate as well, however. All sigmoid functions are logistic with output range [0, 1], except for h , whose range is $[-1, 1]$, and g , whose range is $[-2, 2]$. All weights are initialized in $[-0.2, 0.2]$, except for the output gate biases, which are initialized to $-1, -2$, and -3 , respectively (see abuse problem, solution 2 of section 4). We tried learning rates of 0.1, 0.2, and 0.5.

5.1.5 Results. We use three different, randomly generated pairs of training and test sets. With each such pair we run 10 trials with different initial weights. See Table 1 for results (mean of 30 trials). Unlike the other methods, LSTM always learns to solve the task. Even when we ignore the unsuccessful trials of the other approaches, LSTM learns much faster.

5.1.6 Importance of Output Gates. The experiment provides a nice example where the output gate is truly beneficial. Learning to store the first T or P should not perturb activations representing the more easily learnable transitions of the original Reber grammar. This is the job of the output gates. Without output gates, we did not achieve fast learning.

Table 1: Experiment 1: Embedded Reber Grammar.

Method	Hidden Units	Number of Weights	Learning Rate	% of Success	After
RTRL	3	≈ 170	0.05	Some fraction	173,000
RTRL	12	≈ 494	0.1	Some fraction	25,000
ELM	15	≈ 435		0	>200,000
RCC	7–9	$\approx 119\text{--}198$		50	182,000
LSTM	4 blocks, size 1	264	0.1	100	39,740
LSTM	3 blocks, size 2	276	0.1	100	21,730
LSTM	3 blocks, size 2	276	0.2	97	14,060
LSTM	4 blocks, size 1	264	0.5	97	9500
LSTM	3 blocks, size 2	276	0.5	100	8440

Notes: Percentage of successful trials and number of sequence presentations until success for RTRL (results taken from Smith & Zipser, 1989), Elman net trained by Elman's procedure (results taken from Cleeremans et al., 1989), recurrent cascade-correlation (results taken from Fahlman, 1991), and our new approach (LSTM). Weight numbers in the first four rows are estimates, the corresponding papers do not provide all the technical details. Only LSTM almost always learns to solve the task (only 2 failures out of 150 trials). Even when we ignore the unsuccessful trials of the other approaches, LSTM learns much faster (the number of required training examples in the bottom row varies between 3800 and 24,100).

5.2 Experiment 2: Noise-Free and Noisy Sequences.

5.2.1 Task 2a: Noise-Free Sequences with Long Time Lags. There are $p + 1$ possible input symbols denoted $a_1, \dots, a_{p-1}, a_p = x, a_{p+1} = y$. a_i is locally represented by the $p + 1$ -dimensional vector whose i th component is 1 (all other components are 0). A net with $p + 1$ input units and $p + 1$ output units sequentially observes input symbol sequences, one at a time, permanently trying to predict the next symbol; error signals occur at every time step. To emphasize the long-time-lag problem, we use a training set consisting of only two very similar sequences: $(y, a_1, a_2, \dots, a_{p-1}, y)$ and $(x, a_1, a_2, \dots, a_{p-1}, x)$. Each is selected with probability 0.5. To predict the final element, the net has to learn to store a representation of the first element for p time steps.

We compare real-time recurrent learning for fully recurrent nets (RTRL), back-propagation through time (BPTT), the sometimes very successful two-net neural sequence chunker (CH; Schmidhuber, 1992b), and our new method (LSTM). In all cases, weights are initialized in $[-0.2, 0.2]$. Due to limited computation time, training is stopped after 5 million sequence presentations. A successful run is one that fulfills the following criterion: after training, during 10,000 successive, randomly chosen input sequences, the maximal absolute error of all output units is always below 0.25.

Table 2: Task 2a: Percentage of Successful Trials and Number of Training Sequences until Success.

Method	Delay p	Learning Rate	Number of Weights	% Successful Trials	Success After
RTRL	4	1.0	36	78	1,043,000
RTRL	4	4.0	36	56	892,000
RTRL	4	10.0	36	22	254,000
RTRL	10	1.0–10.0	144	0	> 5,000,000
RTRL	100	1.0–10.0	10404	0	> 5,000,000
BPTT	100	1.0–10.0	10404	0	> 5,000,000
CH	100	1.0	10506	33	32,400
LSTM	100	1.0	10504	100	5,040

Notes: Table entries refer to means of 18 trials. With 100 time-step delays, only CH and LSTM achieve successful trials. Even when we ignore the unsuccessful trials of the other approaches, LSTM learns much faster.

Architectures.

RTRL: One self-recurrent hidden unit, $p + 1$ nonrecurrent output units. Each layer has connections from all layers below. All units use the logistic activation function sigmoid in $[0, 1]$.

BPTT: Same architecture as the one trained by RTRL.

CH: Both net architectures like RTRL's, but one has an additional output for predicting the hidden unit of the other one (see Schmidhuber, 1992b, for details).

LSTM: As with RTRL, but the hidden unit is replaced by a memory cell and an input gate (no output gate required). g is the logistic sigmoid, and h is the identity function $h : h(x) = x, \forall x$. Memory cell and input gate are added once the error has stopped decreasing (see abuse problem: solution 1 in section 4).

Results. Using RTRL and a short four-time-step delay ($p = 4$), 7/9 of all trials were successful. No trial was successful with $p = 10$. With long time lags, only the neural sequence chunker and LSTM achieved successful trials; BPTT and RTRL failed. With $p = 100$, the two-net sequence chunker solved the task in only one-third of all trials. LSTM, however, always learned to solve the task. Comparing successful trials only, LSTM learned much faster. See Table 2 for details. It should be mentioned, however, that a hierarchical chunker can also always quickly solve this task (Schmidhuber, 1992c, 1993).

5.2.2 Task 2b: No Local Regularities. With task 2a, the chunker sometimes learns to predict the final element correctly, but only because of pre-

dictable local regularities in the input stream that allow for compressing the sequence. In a more difficult task, involving many more different possible sequences, we remove compressibility by replacing the deterministic subsequence $(a_1, a_2, \dots, a_{p-1})$ by a random subsequence (of length $p - 1$) over the alphabet a_1, a_2, \dots, a_{p-1} . We obtain two classes (two sets of sequences) $\{(y, a_{i_1}, a_{i_2}, \dots, a_{i_{p-1}}, y) \mid 1 \leq i_1, i_2, \dots, i_{p-1} \leq p - 1\}$ and $\{(x, a_{i_1}, a_{i_2}, \dots, a_{i_{p-1}}, x) \mid 1 \leq i_1, i_2, \dots, i_{p-1} \leq p - 1\}$. Again, every next sequence element has to be predicted. The only totally predictable targets, however, are x and y , which occur at sequence ends. Training exemplars are chosen randomly from the two classes. Architectures and parameters are the same as in experiment 2a. A successful run is one that fulfills the following criterion: after training, during 10,000 successive, randomly chosen input sequences, the maximal absolute error of all output units is below 0.25 at sequence end.

Results. As expected, the chunker failed to solve this task (so did BPTT and RTRL, of course). LSTM, however, was always successful. On average (mean of 18 trials), success for $p = 100$ was achieved after 5680 sequence presentations. This demonstrates that LSTM does not require sequence regularities to work well.

5.2.3 Task 2c: Very Long Time Lags—No Local Regularities. This is the most difficult task in this subsection. To our knowledge, no other recurrent net algorithm can solve it. Now there are $p + 4$ possible input symbols denoted $a_1, \dots, a_{p-1}, a_p, a_{p+1} = e, a_{p+2} = b, a_{p+3} = x, a_{p+4} = y$. a_1, \dots, a_p are also called distractor symbols. Again, a_i is locally represented by the $p + 4$ -dimensional vector whose i th component is 1 (all other components are 0). A net with $p + 4$ input units and 2 output units sequentially observes input symbol sequences, one at a time. Training sequences are randomly chosen from the union of two very similar subsets of sequences: $\{(b, y, a_{i_1}, a_{i_2}, \dots, a_{i_{q+k}}, e, y) \mid 1 \leq i_1, i_2, \dots, i_{q+k} \leq q\}$ and $\{(b, x, a_{i_1}, a_{i_2}, \dots, a_{i_{q+k}}, e, x) \mid 1 \leq i_1, i_2, \dots, i_{q+k} \leq q\}$. To produce a training sequence, we randomly generate a sequence prefix of length $q + 2$, randomly generate a sequence suffix of additional elements ($\neq b, e, x, y$) with probability 9/10 or, alternatively, an e with probability 1/10. In the latter case, we conclude the sequence with x or y , depending on the second element. For a given k , this leads to a uniform distribution on the possible sequences with length $q + k + 4$. The minimal sequence length is $q + 4$; the expected length is

$$4 + \sum_{k=0}^{\infty} \frac{1}{10} \left(\frac{9}{10}\right)^k (q + k) = q + 14.$$

The expected number of occurrences of element a_i , $1 \leq i \leq p$, in a sequence is $(q + 10)/p \approx \frac{q}{p}$. The goal is to predict the last symbol, which always occurs after the “trigger symbol” e . Error signals are generated only at sequence

Table 3: Task 2c: LSTM with Very Long Minimal Time Lags $q + 1$ and a Lot of Noise.

q (Time Lag -1)	p (Number of Random Inputs)	$\frac{q}{p}$	Number of Weights	Success After
50	50	1	364	30,000
100	100	1	664	31,000
200	200	1	1264	33,000
500	500	1	3064	38,000
1000	1,000	1	6064	49,000
1000	500	2	3064	49,000
1000	200	5	1264	75,000
1000	100	10	664	135,000
1000	50	20	364	203,000

Notes: p is the number of available distractor symbols ($p + 4$ is the number of input units). q/p is the expected number of occurrences of a given distractor symbol in a sequence. The right-most column lists the number of training sequences required by LSTM (BPTT, RTRL, and the other competitors have no chance of solving this task). If we let the number of distractor symbols (and weights) increase in proportion to the time lag, learning time increases very slowly. The lower block illustrates the expected slowdown due to increased frequency of distractor symbols.

ends. To predict the final element, the net has to learn to store a representation of the second element for at least $q + 1$ time steps (until it sees the trigger symbol e). Success is defined as prediction error (for final sequence element) of both output units always below 0.2, for 10,000 successive, randomly chosen input sequences.

Architecture/Learning. The net has $p + 4$ input units and 2 output units. Weights are initialized in $[-0.2, 0.2]$. To avoid too much learning time variance due to different weight initializations, the hidden layer gets two memory cells (two cell blocks of size 1, although one would be sufficient). There are no other hidden units. The output layer receives connections only from memory cells. Memory cells and gate units receive connections from input units, memory cells, and gate units (the hidden layer is fully connected). No bias weights are used. h and g are logistic sigmoids with output ranges $[-1, 1]$ and $[-2, 2]$, respectively. The learning rate is 0.01. Note that the minimal time lag is $q + 1$; the net never sees short training sequences facilitating the classification of long test sequences.

Results. Twenty trials were made for all tested pairs (p, q) . Table 3 lists the mean of the number of training sequences required by LSTM to achieve success (BPTT and RTRL have no chance of solving nontrivial tasks with minimal time lags of 1000 steps).

Scaling. Table 3 shows that if we let the number of input symbols (and weights) increase in proportion to the time lag, learning time increases very slowly. This is another remarkable property of LSTM not shared by any other method we are aware of. Indeed, RTRL and BPTT are far from scaling reasonably; instead, they appear to scale exponentially and appear quite useless when the time lags exceed as few as 10 steps.

Distractor Influence. In Table 3, the column headed by q/p gives the expected frequency of distractor symbols. Increasing this frequency decreases learning speed, an effect due to weight oscillations caused by frequently observed input symbols.

5.3 Experiment 3: Noise and Signal on Same Channel. This experiment serves to illustrate that LSTM does not encounter fundamental problems if noise and signal are mixed on the same input line. We initially focus on Bengio et al.’s simple 1994 two-sequence problem. In experiment 3c we pose a more challenging two-sequence problem.

5.3.1 Task 3a (Two-Sequence Problem). The task is to observe and then classify input sequences. There are two classes, each occurring with probability 0.5. There is only one input line. Only the first N real-valued sequence elements convey relevant information about the class. Sequence elements at positions $t > N$ are generated by a gaussian with mean zero and variance 0.2. Case $N = 1$: the first sequence element is 1.0 for class 1, and -1.0 for class 2. Case $N = 3$: the first three elements are 1.0 for class 1 and -1.0 for class 2. The target at the sequence end is 1.0 for class 1 and 0.0 for class 2. Correct classification is defined as absolute output error at sequence end below 0.2. Given a constant T , the sequence length is randomly selected between T and $T + T/10$ (a difference to Bengio et al.’s problem is that they also permit shorter sequences of length $T/2$).

Guessing. Bengio et al. (1994) and Bengio and Frasconi (1994) tested seven different methods on the two-sequence problem. We discovered, however, that random weight guessing easily outperforms them all because the problem is so simple.⁵ See Schmidhuber and Hochreiter (1996) and Hochreiter and Schmidhuber (1996, 1997) for additional results in this vein.

LSTM Architecture. We use a three-layer net with one input unit, one output unit, and three cell blocks of size 1. The output layer receives connections only from memory cells. Memory cells and gate units receive inputs from input units, memory cells, and gate units and have bias weights. Gate

⁵ However, different input representations and different types of noise may lead to worse guessing performance (Yoshua Bengio, personal communication, 1996).

Table 4: Task 3a: Bengio et al.’s Two-Sequence Problem.

T	N	Stop: ST1	Stop: ST2	Number of Weights	ST2: Fraction Misclassified
100	3	27,380	39,850	102	0.000195
100	1	58,370	64,330	102	0.000117
1000	3	446,850	452,460	102	0.000078

Notes: T is minimal sequence length. N is the number of information-conveying elements at sequence begin. The column headed by ST1 (ST2) gives the number of sequence presentations required to achieve stopping criterion ST1 (ST2). The right-most column lists the fraction of misclassified posttraining sequences (with absolute error > 0.2) from a test set consisting of 2560 sequences (tested after ST2 was achieved). All values are means of 10 trials. We discovered, however, that this problem is so simple that random weight guessing solves it faster than LSTM and any other method for which there are published results.

units and output unit are logistic sigmoid in $[0, 1]$, h in $[-1, 1]$, and g in $[-2, 2]$.

Training/Testing. All weights (except the bias weights to gate units) are randomly initialized in the range $[-0.1, 0.1]$. The first input gate bias is initialized with -1.0 , the second with -3.0 , and the third with -5.0 . The first output gate bias is initialized with -2.0 , the second with -4.0 , and the third with -6.0 . The precise initialization values hardly matter though, as confirmed by additional experiments. The learning rate is 1.0. All activations are reset to zero at the beginning of a new sequence.

We stop training (and judge the task as being solved) according to the following criteria: ST1: none of 256 sequences from a randomly chosen test set is misclassified; ST2: ST1 is satisfied, and mean absolute test set error is below 0.01. In case of ST2, an additional test set consisting of 2560 randomly chosen sequences is used to determine the fraction of misclassified sequences.

Results. See Table 4. The results are means of 10 trials with different weight initializations in the range $[-0.1, 0.1]$. LSTM is able to solve this problem, though by far not as fast as random weight guessing (see “Guessing” above). Clearly this trivial problem does not provide a very good testbed to compare performance of various nontrivial algorithms. Still, it demonstrates that LSTM does not encounter fundamental problems when faced with signal and noise on the same channel.

5.3.2 Task 3b. The architecture, parameters, and other elements are as in task 3a, but now with gaussian noise (mean 0 and variance 0.2) added to the

Table 5: Task 3b: Modified Two-Sequence Problem.

T	N	Stop: ST1	Stop: ST2	Number of Weights	ST2: Fraction Misclassified
100	3	41,740	43,250	102	0.00828
100	1	74,950	78,430	102	0.01500
1000	1	481,060	485,080	102	0.01207

Note: Same as in Table 4, but now the information-conveying elements are also perturbed by noise.

information-conveying elements ($t \leq N$). We stop training (and judge the task as being solved) according to the following, slightly redefined criteria: ST1: fewer than 6 out of 256 sequences from a randomly chosen test set are misclassified; ST2: ST1 is satisfied, and mean absolute test set error is below 0.04. In case of ST2, an additional test set consisting of 2560 randomly chosen sequences is used to determine the fraction of misclassified sequences.

Results. See Table 5. The results represent means of 10 trials with different weight initializations. LSTM easily solves the problem.

5.3.3 Task 3c. The architecture, parameters, and other elements are as in task 3a, but with a few essential changes that make the task nontrivial: the targets are 0.2 and 0.8 for class 1 and class 2, respectively, and there is gaussian noise on the targets (mean 0 and variance 0.1; S.D. 0.32). To minimize mean squared error, the system has to learn the conditional expectations of the targets given the inputs. Misclassification is defined as absolute difference between output and noise-free target (0.2 for class 1 and 0.8 for class 2) > 0.1 . The network output is considered acceptable if the mean absolute difference between noise-free target and output is below 0.015. Since this requires high weight precision, task 3c (unlike tasks 3a and 3b) cannot be solved quickly by random guessing.

Training/Testing. The learning rate is 0.1. We stop training according to the following criterion: none of 256 sequences from a randomly chosen test set is misclassified, and mean absolute difference between the noise-free target and output is below 0.015. An additional test set consisting of 2560 randomly chosen sequences is used to determine the fraction of misclassified sequences.

Results. See Table 6. The results represent means of 10 trials with different weight initializations. Despite the noisy targets, LSTM still can solve the problem by learning the expected target values.

Table 6: Task 3c: Modified, More Challenging Two-Sequence Problem.

T	N	Stop	Number of Weights	Fraction Misclassified	Average Difference to Mean
100	3	269,650	102	0.00558	0.014
100	1	565,640	102	0.00441	0.012

Notes: Same as in Table 4, but with noisy real-valued targets. The system has to learn the conditional expectations of the targets given the inputs. The right-most column provides the average difference between network output and expected target. Unlike tasks 3a and 3b, this one cannot be solved quickly by random weight guessing.

5.4 Experiment 4: Adding Problem. The difficult task in this section is of a type that has never been solved by other recurrent net algorithms. It shows that LSTM can solve long-time-lag problems involving distributed, continuous-valued representations.

5.4.1 Task. Each element of each input sequence is a pair of components. The first component is a real value randomly chosen from the interval $[-1, 1]$; the second is 1.0, 0.0, or -1.0 and is used as a marker. At the end of each sequence, the task is to output the sum of the first components of those pairs that are marked by second components equal to 1.0. Sequences have random lengths between the minimal sequence length T and $T + T/10$. In a given sequence, exactly two pairs are marked, as follows: we first randomly select and mark one of the first 10 pairs (whose first component we call X_1). Then we randomly select and mark one of the first $T/2 - 1$ still unmarked pairs (whose first component we call X_2). The second components of all remaining pairs are zero except for the first and final pair, whose second components are -1. (In the rare case where the first pair of the sequence gets marked, we set X_1 to zero.) An error signal is generated only at the sequence end: the target is $0.5 + (X_1 + X_2)/4.0$ (the sum $X_1 + X_2$ scaled to the interval $[0, 1]$). A sequence is processed correctly if the absolute error at the sequence end is below 0.04.

5.4.2 Architecture. We use a three-layer net with two input units, one output unit, and two cell blocks of size 2. The output layer receives connections only from memory cells. Memory cells and gate units receive inputs from memory cells and gate units (the hidden layer is fully connected; less connectivity may work as well). The input layer has forward connections to all units in the hidden layer. All noninput units have bias weights. These architecture parameters make it easy to store at least two input signals (a cell block size of 1 works well, too). All activation functions are logistic with output range $[0, 1]$, except for h , whose range is $[-1, 1]$, and g , whose range is $[-2, 2]$.

Table 7: Experiment 4: Results for the Adding Problem.

T	Minimal Lag	Number of Weights	Number of Wrong Predictions	Success After
100	50	93	1 out of 2560	74,000
500	250	93	0 out of 2560	209,000
1000	500	93	1 out of 2560	853,000

Notes: T is the minimal sequence length, $T/2$ the minimal time lag. “Number of Wrong Predictions” is the number of incorrectly processed sequences (error > 0.04) from a test set containing 2560 sequences. The right-most column gives the number of training sequences required to achieve the stopping criterion. All values are means of 10 trials. For $T = 1000$ the number of required training examples varies between 370,000 and 2,020,000, exceeding 700,000 in only three cases.

5.4.3 State Drift Versus Initial Bias. Note that the task requires storing the precise values of real numbers for long durations; the system must learn to protect memory cell contents against even minor internal state drift (see section 4). To study the significance of the drift problem, we make the task even more difficult by biasing all noninput units, thus artificially inducing internal state drift. All weights (including the bias weights) are randomly initialized in the range $[-0.1, 0.1]$. Following section 4’s remedy for state drifts, the first input gate bias is initialized with -3.0 and the second with -6.0 (though the precise values hardly matter, as confirmed by additional experiments).

5.4.4 Training/Testing. The learning rate is 0.5. Training is stopped once the average training error is below 0.01, and the 2000 most recent sequences were processed correctly.

5.4.5 Results. With a test set consisting of 2560 randomly chosen sequences, the average test set error was always below 0.01, and there were never more than three incorrectly processed sequences. Table 7 shows details.

The experiment demonstrates that LSTM is able to work well with distributed representations, LSTM is able to learn to perform calculations involving continuous values, and since the system manages to store continuous values without deterioration for minimal delays of $T/2$ time steps, there is no significant, harmful internal state drift.

5.5 Experiment 5: Multiplication Problem. One may argue that LSTM is a bit biased toward tasks such as the adding problem from the previous subsection. Solutions to the adding problem may exploit the CEC’s built-in integration capabilities. Although this CEC property may be viewed as a

Table 8: Experiment 5: Results for the Multiplication Problem.

T	Minimal Lag	Number of Weights	n_{seq}	Number of Wrong Predictions	MSE	Success After
100	50	93	140	139 out of 2560	0.0223	482,000
100	50	93	13	14 out of 2560	0.0139	1,273,000

Notes: T is the minimal sequence length and $T/2$ the minimal time lag. We test on a test set containing 2560 sequences as soon as less than n_{seq} of the 2000 most recent training sequences lead to error > 0.04 . “Number of Wrong Predictions” is the number of test sequences with error > 0.04 . MSE is the mean squared error on the test set. The right-most column lists numbers of training sequences required to achieve the stopping criterion. All values are means of 10 trials.

feature rather than a disadvantage (integration seems to be a natural subtask of many tasks occurring in the real world), the question arises whether LSTM can also solve tasks with inherently nonintegrative solutions. To test this, we change the problem by requiring the final target to equal the product (instead of the sum) of earlier marked inputs.

5.5.1 Task. This is like the task in section 5.4, except that the first component of each pair is a real value randomly chosen from the interval $[0, 1]$. In the rare case where the first pair of the input sequence gets marked, we set X_1 to 1.0. The target at sequence end is the product $X_1 \times X_2$.

5.5.2 Architecture. This is as in section 5.4. All weights (including the bias weights) are randomly initialized in the range $[-0.1, 0.1]$.

5.5.3 Training/Testing. The learning rate is 0.1. We test performance twice: as soon as less than n_{seq} of the 2000 most recent training sequences lead to absolute errors exceeding 0.04, where $n_{seq} = 140$ and $n_{seq} = 13$. Why these values? $n_{seq} = 140$ is sufficient to learn storage of the relevant inputs. It is not enough though to fine-tune the precise final outputs. $n_{seq} = 13$, however, leads to quite satisfactory results.

5.5.4 Results. For $n_{seq} = 140$ ($n_{seq} = 13$) with a test set consisting of 2560 randomly chosen sequences, the average test set error was always below 0.026 (0.013), and there were never more than 170 (15) incorrectly processed sequences. Table 8 shows details. (A net with additional standard hidden units or with a hidden layer above the memory cells may learn the fine-tuning part more quickly.)

The experiment demonstrates that LSTM can solve tasks involving both continuous-valued representations and nonintegrative information processing.

5.6 Experiment 6: Temporal Order. In this subsection, LSTM solves other difficult (but artificial) tasks that have never been solved by previous recurrent net algorithms. The experiment shows that LSTM is able to extract information conveyed by the temporal order of widely separated inputs.

5.6.1 Task 6a: Two Relevant, Widely Separated Symbols. The goal is to classify sequences. Elements and targets are represented locally (input vectors with only one nonzero bit). The sequence starts with an E , ends with a B (the “trigger symbol”), and otherwise consists of randomly chosen symbols from the set $\{a, b, c, d\}$ except for two elements at positions t_1 and t_2 that are either X or Y . The sequence length is randomly chosen between 100 and 110, t_1 is randomly chosen between 10 and 20, and t_2 is randomly chosen between 50 and 60. There are four sequence classes Q, R, S, U , which depend on the temporal order of X and Y . The rules are: $X, X \rightarrow Q$; $X, Y \rightarrow R$; $Y, X \rightarrow S$; $Y, Y \rightarrow U$.

5.6.2 Task 6b: Three Relevant, Widely Separated Symbols. Again, the goal is to classify sequences. Elements and targets are represented locally. The sequence starts with an E , ends with a B (the trigger symbol), and otherwise consists of randomly chosen symbols from the set $\{a, b, c, d\}$ except for three elements at positions t_1, t_2 , and t_3 that are either X or Y . The sequence length is randomly chosen between 100 and 110, t_1 is randomly chosen between 10 and 20, t_2 is randomly chosen between 33 and 43, and t_3 is randomly chosen between 66 and 76. There are eight sequence classes— Q, R, S, U, V, A, B, C —which depend on the temporal order of the X s and Y s. The rules are: $X, X, X \rightarrow Q$; $X, X, Y \rightarrow R$; $X, Y, X \rightarrow S$; $X, Y, Y \rightarrow U$; $Y, X, X \rightarrow V$; $Y, X, Y \rightarrow A$; $Y, Y, X \rightarrow B$; $Y, Y, Y \rightarrow C$.

There are as many output units as there are classes. Each class is locally represented by a binary target vector with one nonzero component. With both tasks, error signals occur only at the end of a sequence. The sequence is classified correctly if the final absolute error of all output units is below 0.3.

Architecture. We use a three-layer net with eight input units, two (three) cell blocks of size 2, and four (eight) output units for task 6a (6b). Again all noninput units have bias weights, and the output layer receives connections from memory cells, only. Memory cells and gate units receive inputs from input units, memory cells, and gate units (the hidden layer is fully connected; less connectivity may work as well). The architecture parameters for task 6a (6b) make it easy to store at least two (three) input signals. All activation functions are logistic with output range $[0, 1]$, except for h , whose range is $[-1, 1]$, and g , whose range is $[-2, 2]$.

Table 9: Experiment 6: Results for the Temporal Order Problem.

Task	Number of Weights	Number of Wrong Predictions	Success After
Task 6a	156	1 out of 2560	31,390
Task 6b	308	2 out of 2560	571,100

Notes: "Number of Wrong Predictions" is the number of incorrectly classified sequences (error > 0.3 for at least one output unit) from a test set containing 2560 sequences. The right-most column gives the number of training sequences required to achieve the stopping criterion. The results for task 6a are means of 20 trials; those for task 6b of 10 trials.

Training/Testing. The learning rate is 0.5 (0.1) for experiment 6a (6b). Training is stopped once the average training error falls below 0.1 and the 2000 most recent sequences were classified correctly. All weights are initialized in the range $[-0.1, 0.1]$. The first input gate bias is initialized with -2.0 , the second with -4.0 , and (for experiment 6b) the third with -6.0 (again, we confirmed by additional experiments that the precise values hardly matter).

Results. With a test set consisting of 2560 randomly chosen sequences, the average test set error was always below 0.1, and there were never more than three incorrectly classified sequences. Table 9 shows details.

The experiment shows that LSTM is able to extract information conveyed by the temporal order of widely separated inputs. In task 6a, for instance, the delays between the first and second relevant input and between the second relevant input and sequence end are at least 30 time steps.

Typical Solutions. In experiment 6a, how does LSTM distinguish between temporal orders (X, Y) and (Y, X) ? One of many possible solutions is to store the first X or Y in cell block 1 and the second X/Y in cell block 2. Before the first X/Y occurs, block 1 can see that it is still empty by means of its recurrent connections. After the first X/Y , block 1 can close its input gate. Once block 1 is filled and closed, this fact will become visible to block 2 (recall that all gate units and all memory cells receive connections from all nonoutput units).

Typical solutions, however, require only one memory cell block. The block stores the first X or Y ; once the second X/Y occurs, it changes its state depending on the first stored symbol. Solution type 1 exploits the connection between memory cell output and input gate unit. The following events cause different input gate activations: X occurs in conjunction with a filled block; X occurs in conjunction with an empty block. Solution type 2 is based on a strong, positive connection between memory cell output and memory cell input. The previous occurrence of X (Y) is represented by a

Table 10: Summary of Experimental Conditions for LSTM, Part I.

Task	(1) p	(2) lag	(3) b	(4) s	(5) in	(6) out	(8) w	(9) c	(10) ogb	(11) igb	(12) bias	(13) h	(14) g	(15) α
1-1	9	9	4	1	7	7	264	F	-1, -2, -3, -4	r	ga	h1	g2	0.1
1-2	9	9	3	2	7	7	276	F	-1, -2, -3	r	ga	h1	g2	0.1
1-3	9	9	3	2	7	7	276	F	-1, -2, -3	r	ga	h1	g2	0.2
1-4	9	9	4	1	7	7	264	F	-1, -2, -3, -4	r	ga	h1	g2	0.5
1-5	9	9	3	2	7	7	276	F	-1, -2, -3	r	ga	h1	g2	0.5
2a	100	100	1	1	101	101	10,504	B	No og	None	None	id	g1	1.0
2b	100	100	1	1	101	101	10,504	B	No og	None	None	id	g1	1.0
2c-1	50	50	2	1	54	2	364	F	None	None	None	h1	g2	0.01
2c-2	100	100	2	1	104	2	664	F	None	None	None	h1	g2	0.01
2c-3	200	200	2	1	204	2	1264	F	None	None	None	h1	g2	0.01
2c-4	500	500	2	1	504	2	3064	F	None	None	None	h1	g2	0.01
2c-5	1000	1000	2	1	1004	2	6064	F	None	None	None	h1	g2	0.01
2c-6	1000	1000	2	1	504	2	3064	F	None	None	None	h1	g2	0.01
2c-7	1000	1000	2	1	204	2	1264	F	None	None	None	h1	g2	0.01
2c-8	1000	1000	2	1	104	2	664	F	None	None	None	h1	g2	0.01
2c-9	1000	1000	2	1	54	2	364	F	None	None	None	h1	g2	0.01
3a	100	100	3	1	1	1	102	F	-2, -4, -6	-1, -3, -5	b1	h1	g2	1.0
3b	100	100	3	1	1	1	102	F	-2, -4, -6	-1, -3, -5	b1	h1	g2	1.0
3c	100	100	3	1	1	1	102	F	-2, -4, -6	-1, -3, -5	b1	h1	g2	0.1
4-1	100	50	2	2	2	1	93	F	r	-3, -6	All	h1	g2	0.5
4-2	500	250	2	2	2	1	93	F	r	-3, -6	All	h1	g2	0.5
4-3	1000	500	2	2	2	1	93	F	r	-3, -6	All	h1	g2	0.5
5	100	50	2	2	2	1	93	F	r	r	All	h1	g2	0.1
6a	100	40	2	2	8	4	156	F	r	-2, -4	All	h1	g2	0.5
6b	100	24	3	2	8	8	308	F	r	-2, -4, -6	All	h1	g2	0.1

Notes: Col. 1: task number. Col. 2: minimal sequence length p . Col. 3: minimal number of steps between most recent relevant input information and teacher signal. Col. 4: number of cell blocks b . Col. 5: block size s . Col. 6: Number of input units in . Col. 7: Number of output units out . Col. 8: number of weights w . Col. 9: c describes connectivity: F means “output layer receives connections from memory cells; memory cells and gate units receive connections from input units, memory cells and gate units”; B means “each layer receives connections from all layers below.” Col. 10: Initial output gate bias ogb , where r stands for “randomly chosen from the interval $[-0.1, 0.1]$ ” and $no\ og$ means “no output gate used.” Col. 11: initial input gate bias igb (see Col. 10). Col. 12: which units have bias weights? $b1$ stands for “all hidden units”, ga for “only gate units,” and all for “all noninput units.” Col. 13: the function h , where id is identity function, $h1$ is logistic sigmoid in $[-2, 2]$. Col. 14: the logistic function g , where $g1$ is sigmoid in $[0, 1]$, $g2$ in $[-1, 1]$. Col. 15: learning rate α .

positive (negative) internal state. Once the input gate opens for the second time, so does the output gate, and the memory cell output is fed back to its own input. This causes (X, Y) to be represented by a positive internal state, because X contributes to the new internal state twice (via current internal state and cell output feedback). Similarly, (Y, X) gets represented by a negative internal state.

5.7 Summary of Experimental Conditions. Tables 10 and 11 provide an overview of the most important LSTM parameters and architectural details for experiments 1 through 6. The conditions of the simple experiments 2a

Table 11: Summary of Experimental Conditions for LSTM, Part II.

(1) Task	(2) Select	(3) Interval	(4) Test Set Size	(5) Stopping Criterion	(6) Success
1	t1	[−0.2, 0.2]	256	Training and test correctly pred.	See text
2a	t1	[−0.2, 0.2]	no test set	After 5 million exemplars	ABS(0.25)
2b	t2	[−0.2, 0.2]	10,000	After 5 million exemplars	ABS(0.25)
2c	t2	[−0.2, 0.2]	10,000	After 5 million exemplars	ABS(0.2)
3a	t3	[−0.1, 0.1]	2560	ST1 and ST2 (see text)	ABS(0.2)
3b	t3	[−0.1, 0.1]	2560	ST1 and ST2 (see text)	ABS(0.2)
3c	t3	[−0.1, 0.1]	2560	ST1 and ST2 (see text)	See text
4	t3	[−0.1, 0.1]	2560	ST3(0.01)	ABS(0.04)
5	t3	[−0.1, 0.1]	2560	see text	ABS(0.04)
6a	t3	[−0.1, 0.1]	2560	ST3(0.1)	ABS(0.3)
6b	t3	[−0.1, 0.1]	2560	ST3(0.1)	ABS(0.3)

Notes: Col. 1: task number. Col. 2: training exemplar selection, where t1 stands for “randomly chosen from training set,” t2 for “randomly chosen from two classes,” and t3 for “randomly generated on line.” Col. 3: weight initialization interval. Col. 4: test set size. Col. 5: Stopping criterion for training, where $ST3(\beta)$ stands for “average training error below β and the 2000 most recent sequences were processed correctly.” Col. 6: success (correct classification) criterion, where $ABS(\beta)$ stands for “absolute error of all output units at sequence end is below β .”

and 2b differ slightly from those of the other, more systematic experiments, due to historical reasons.

6 Discussion

6.1 Limitations of LSTM.

- The particularly efficient truncated backpropagation version of the LSTM algorithm will not easily solve problems similar to strongly delayed XOR problems, where the goal is to compute the XOR of two widely separated inputs that previously occurred somewhere in a noisy sequence. The reason is that storing only one of the inputs will not help to reduce the expected error; the task is nondecomposable in the sense that it is impossible to reduce the error incrementally by first solving an easier subgoal.

In theory, this limitation can be circumvented by using the full gradient (perhaps with additional conventional hidden units receiving input from the memory cells). But we do not recommend computing the full gradient for the following reasons: (1) It increases computational complexity, (2) constant error flow through CECs can be shown only for truncated LSTM, and (3) we actually did conduct a few experiments with nontruncated LSTM. There was no significant difference to truncated LSTM, exactly because outside the CECs, error flow tends

to vanish quickly. For the same reason, full BPTT does not outperform truncated BPTT.

- Each memory cell block needs two additional units (input and output gate). In comparison to standard recurrent nets, however, this does not increase the number of weights by more than a factor of 9: each conventional hidden unit is replaced by at most three units in the LSTM architecture, increasing the number of weights by a factor of 3^2 in the fully connected case. Note, however, that our experiments use quite comparable weight numbers for the architectures of LSTM and competing approaches.
- Due to its constant error flow through CECs within memory cells, LSTM generally runs into problems similar to those of feedforward nets' seeing the entire input string at once. For instance, there are tasks that can be quickly solved by random weight guessing but not by the truncated LSTM algorithm with small weight initializations, such as the 500-step parity problem (see the introduction to section 5). Here, LSTM's problems are similar to the ones of a feedforward net with 500 inputs, trying to solve 500-bit parity. Indeed LSTM typically behaves much like a feedforward net trained by backpropagation that sees the entire input. But that is also precisely why it so clearly outperforms previous approaches on many nontrivial tasks with significant search spaces.
- LSTM does not have any problems with the notion of recency that go beyond those of other approaches. All gradient-based approaches, however, suffer from a practical inability to count discrete time steps precisely. If it makes a difference whether a certain signal occurred 99 or 100 steps ago, then an additional counting mechanism seems necessary. Easier tasks, however, such as one that requires making a difference only between, say, 3 and 11 steps, do not pose any problems to LSTM. For instance, by generating an appropriate negative connection between memory cell output and input, LSTM can give more weight to recent inputs and learn decays where necessary.

6.2 Advantages of LSTM.

- The constant error backpropagation within memory cells results in LSTM's ability to bridge very long time lags in case of problems similar to those discussed above.
- For long-time-lag problems such as those discussed in this article, LSTM can handle noise, distributed representations, and continuous values. In contrast to finite state automata or hidden Markov models, LSTM does not require an *a priori* choice of a finite number of states. In principle, it can deal with unlimited state numbers.

- For problems discussed in this article, LSTM generalizes well, even if the positions of widely separated, relevant inputs in the input sequence do not matter. Unlike previous approaches, ours quickly learns to distinguish between two or more widely separated occurrences of a particular element in an input sequence, without depending on appropriate short-time-lag training exemplars.
- There appears to be no need for parameter fine tuning. LSTM works well over a broad range of parameters such as learning rate, input gate bias, and output gate bias. For instance, to some readers the learning rates used in our experiments may seem large. However, a large learning rate pushes the output gates toward zero, thus automatically countermanding its own negative effects.
- The LSTM algorithm's update complexity per weight and time step is essentially that of BPTT, namely, $O(1)$. This is excellent in comparison to other approaches such as RTRL. Unlike full BPTT, however, LSTM is local in both space and time.

7 Conclusion

Each memory cell's internal architecture guarantees constant error flow within its CEC, provided that truncated backpropagation cuts off error flow trying to leak out of memory cells. This represents the basis for bridging very long time lags. Two gate units learn to open and close access to error flow within each memory cell's CEC. The multiplicative input gate affords protection of the CEC from perturbation by irrelevant inputs. Similarly, the multiplicative output gate protects other units from perturbation by currently irrelevant memory contents.

To find out about LSTM's practical limitations we intend to apply it to real-world data. Application areas will include time-series prediction, music composition, and speech processing. It will also be interesting to augment sequence chunkers (Schmidhuber, 1992b, 1993) by LSTM to combine the advantages of both.

Appendix

A.1 Algorithm Details. In what follows, the index k ranges over output units, i ranges over hidden units, c_j stands for the j th memory cell block, c_j^v denotes the v th unit of memory cell block c_j , u , l , m stand for arbitrary units, and t ranges over all time steps of a given input sequence.

The gate unit logistic sigmoid (with range $[0, 1]$) used in the experiments is

$$f(x) = \frac{1}{1 + \exp(-x)} . \quad (\text{A.1})$$

The function h (with range $[-1, 1]$) used in the experiments is

$$h(x) = \frac{2}{1 + \exp(-x)} - 1. \quad (\text{A.2})$$

The function g (with range $[-2, 2]$) used in the experiments is

$$g(x) = \frac{4}{1 + \exp(-x)} - 2. \quad (\text{A.3})$$

A.1.1 Forward Pass. The net input and the activation of hidden unit i are

$$\begin{aligned} \text{net}_i(t) &= \sum_u w_{iu} y^u(t-1) \\ y^i(t) &= f_i(\text{net}_i(t)). \end{aligned} \quad (\text{A.4})$$

The net input and the activation of in_j are

$$\begin{aligned} \text{net}_{in_j}(t) &= \sum_u w_{in_j u} y^u(t-1) \\ y^{in_j}(t) &= f_{in_j}(\text{net}_{in_j}(t)). \end{aligned} \quad (\text{A.5})$$

The net input and the activation of out_j are

$$\begin{aligned} \text{net}_{out_j}(t) &= \sum_u w_{out_j u} y^u(t-1) \\ y^{out_j}(t) &= f_{out_j}(\text{net}_{out_j}(t)). \end{aligned} \quad (\text{A.6})$$

The net input $\text{net}_{c_j^v}$, the internal state $s_{c_j^v}$, and the output activation $y^{c_j^v}$ of the v th memory cell of memory cell block c_j are:

$$\begin{aligned} \text{net}_{c_j^v}(t) &= \sum_u w_{c_j^v u} y^u(t-1) \\ s_{c_j^v}(t) &= s_{c_j^v}(t-1) + y^{in_j}(t) g(\text{net}_{c_j^v}(t)) \\ y^{c_j^v}(t) &= y^{out_j}(t) h(s_{c_j^v}(t)). \end{aligned} \quad (\text{A.7})$$

The net input and the activation of output unit k are

$$\begin{aligned} \text{net}_k(t) &= \sum_{u: u \text{ not a gate}} w_{ku} y^u(t-1) \\ y^k(t) &= f_k(\text{net}_k(t)). \end{aligned}$$

The backward pass to be described later is based on the following truncated backpropagation formulas.

A.1.2 Approximate Derivatives for Truncated Backpropagation. The truncated version (see section 4) only approximates the partial derivatives, which is reflected by the \approx_{tr} signs in the notation below. It truncates error flow once it leaves memory cells or gate units. Truncation ensures that there are no loops across which an error that left some memory cell through its input or input gate can reenter the cell through its output or output gate. This in turn ensures constant error flow through the memory cell's CEC.

In the truncated backpropagation version, the following derivatives are replaced by zero:

$$\frac{\partial \text{net}_{inj}(t)}{\partial y^u(t-1)} \approx_{tr} 0 \quad \forall u,$$

$$\frac{\partial \text{net}_{outj}(t)}{\partial y^u(t-1)} \approx_{tr} 0 \quad \forall u,$$

and

$$\frac{\partial \text{net}_g(t)}{\partial y^u(t-1)} \approx_{tr} 0 \quad \forall u.$$

Therefore we get

$$\frac{\partial y^{inj}(t)}{\partial y^u(t-1)} = f'_{inj}(\text{net}_{inj}(t)) \frac{\partial \text{net}_{inj}(t)}{\partial y^u(t-1)} \approx_{tr} 0 \quad \forall u,$$

$$\frac{\partial y^{outj}(t)}{\partial y^u(t-1)} = f'_{outj}(\text{net}_{outj}(t)) \frac{\partial \text{net}_{outj}(t)}{\partial y^u(t-1)} \approx_{tr} 0 \quad \forall u,$$

and

$$\begin{aligned} \frac{\partial y^g(t)}{\partial y^u(t-1)} &= \frac{\partial y^g(t)}{\partial \text{net}_{outj}(t)} \frac{\partial \text{net}_{outj}(t)}{\partial y^u(t-1)} + \frac{\partial y^g(t)}{\partial \text{net}_{inj}(t)} \frac{\partial \text{net}_{inj}(t)}{\partial y^u(t-1)} \\ &+ \frac{\partial y^g(t)}{\partial \text{net}_g(t)} \frac{\partial \text{net}_g(t)}{\partial y^u(t-1)} \approx_{tr} 0 \quad \forall u. \end{aligned}$$

This implies for all w_{lm} not on connections to c_j^v , in_j , out_j (that is, $l \notin \{c_j^v, in_j, out_j\}$):

$$\frac{\partial y^{c_j^v}(t)}{\partial w_{lm}} = \sum_u \frac{\partial y^{c_j^v}(t)}{\partial y^u(t-1)} \frac{\partial y^u(t-1)}{\partial w_{lm}} \approx_{tr} 0.$$

The truncated derivatives of output unit k are:

$$\frac{\partial y^k(t)}{\partial w_{lm}} = f'_k(\text{net}_k(t)) \left(\sum_{u: u \text{ not a gate}} w_{ku} \frac{\partial y^u(t-1)}{\partial w_{lm}} + \delta_{kl} y^m(t-1) \right)$$

$$\begin{aligned}
& \approx_{tr} f'_k(\text{net}_k(t)) \left(\sum_j \sum_{v=1}^{S_j} \delta_{c_j^v l} w_{kc_j^v} \frac{\partial y^{c_j^v}(t-1)}{\partial w_{lm}} \right. \\
& \quad + \sum_j (\delta_{in_j l} + \delta_{out_j l}) \sum_{v=1}^{S_j} w_{kc_j^v} \frac{\partial y^{c_j^v}(t-1)}{\partial w_{lm}} \\
& \quad \left. + \sum_{i: i \text{ hidden unit}} w_{ki} \frac{\partial y^i(t-1)}{\partial w_{lm}} + \delta_{kl} y^m(t-1) \right) \\
& = f'_k(\text{net}_k(t)) \begin{cases} y^m(t-1) & l = k \\ w_{kc_j^v} \frac{\partial y^{c_j^v}(t-1)}{\partial w_{lm}} & l = c_j^v \\ \sum_{v=1}^{S_j} w_{kc_j^v} \frac{\partial y^{c_j^v}(t-1)}{\partial w_{lm}} & l = in_j \text{ OR } l = out_j \\ \sum_{i: i \text{ hidden unit}} w_{ki} \frac{\partial y^i(t-1)}{\partial w_{lm}} & l \text{ otherwise} \end{cases} \quad (A.8)
\end{aligned}$$

where δ is the Kronecker delta ($\delta_{ab} = 1$ if $a = b$ and 0 otherwise), and S_j is the size of memory cell block c_j . The truncated derivatives of a hidden unit i that is not part of a memory cell are:

$$\frac{\partial y^i(t)}{\partial w_{lm}} = f'_i(\text{net}_i(t)) \frac{\partial \text{net}_i(t)}{\partial w_{lm}} \approx_{tr} \delta_{li} f'_i(\text{net}_i(t)) y^m(t-1). \quad (A.9)$$

(Here it would be possible to use the full gradient without affecting constant error flow through internal states of memory cells.)

Cell block c_j 's truncated derivatives are:

$$\begin{aligned}
\frac{\partial y^{in_j}(t)}{\partial w_{lm}} &= f'_{in_j}(\text{net}_{in_j}(t)) \frac{\partial \text{net}_{in_j}(t)}{\partial w_{lm}} \\
&\approx_{tr} \delta_{in_j l} f'_{in_j}(\text{net}_{in_j}(t)) y^m(t-1). \quad (A.10)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial y^{out_j}(t)}{\partial w_{lm}} &= f'_{out_j}(\text{net}_{out_j}(t)) \frac{\partial \text{net}_{out_j}(t)}{\partial w_{lm}} \\
&\approx_{tr} \delta_{out_j l} f'_{out_j}(\text{net}_{out_j}(t)) y^m(t-1). \quad (A.11)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial s_{c_j^v}(t)}{\partial w_{lm}} &= \frac{\partial s_{c_j^v}(t-1)}{\partial w_{lm}} \\
&\quad + \frac{\partial y^{in_j}(t)}{\partial w_{lm}} g(\text{net}_{c_j^v}(t)) + y^{in_j}(t) g'(\text{net}_{c_j^v}(t)) \frac{\partial \text{net}_{c_j^v}(t)}{\partial w_{lm}} \\
&\approx_{tr} (\delta_{in_j l} + \delta_{c_j^v l}) \frac{\partial s_{c_j^v}(t-1)}{\partial w_{lm}} + \delta_{in_j l} \frac{\partial y^{in_j}(t)}{\partial w_{lm}} g(\text{net}_{c_j^v}(t)) \\
&\quad + \delta_{c_j^v l} y^{in_j}(t) g'(\text{net}_{c_j^v}(t)) \frac{\partial \text{net}_{c_j^v}(t)}{\partial w_{lm}}
\end{aligned}$$

$$\begin{aligned}
&= (\delta_{in_j l} + \delta_{c_j^v l}) \frac{\partial s_{c_j^v}(t-1)}{\partial w_{lm}} \\
&\quad + \delta_{in_j l} f'_{in_j}(net_{in_j}(t)) g(net_{c_j^v}(t)) y^m(t-1) \\
&\quad + \delta_{c_j^v l} y^{in_j}(t) g'(net_{c_j^v}(t)) y^m(t-1). \tag{A.12}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial y^{c_j^v}(t)}{\partial w_{lm}} &= \frac{\partial y^{out_j}(t)}{\partial w_{lm}} h(s_{c_j^v}(t)) + h'(s_{c_j^v}(t)) \frac{\partial s_{c_j^v}(t)}{\partial w_{lm}} y^{out_j}(t) \\
&\approx_{tr} \delta_{out_j l} \frac{\partial y^{out_j}(t)}{\partial w_{lm}} h(s_{c_j^v}(t)) \\
&\quad + (\delta_{in_j l} + \delta_{c_j^v l}) h'(s_{c_j^v}(t)) \frac{\partial s_{c_j^v}(t)}{\partial w_{lm}} y^{out_j}(t). \tag{A.13}
\end{aligned}$$

To update the system efficiently at time t , the only (truncated) derivatives that need to be stored at time $t-1$ are

$$\frac{\partial s_{c_j^v}(t-1)}{\partial w_{lm}},$$

where $l = c_j^v$ or $l = in_j$.

A.1.3 Backward Pass. We will describe the backward pass only for the particularly efficient truncated gradient version of the LSTM algorithm. For simplicity we will use equal signs even where approximations are made according to the truncated backpropagation equations above.

The squared error at time t is given by

$$E(t) = \sum_{k: k \text{ output unit}} \left(t^k(t) - y^k(t) \right)^2, \tag{A.14}$$

where $t^k(t)$ is output unit k 's target at time t .

Time t 's contribution to w_{lm} 's gradient-based update with learning rate α is

$$\Delta w_{lm}(t) = -\alpha \frac{\partial E(t)}{\partial w_{lm}}. \tag{A.15}$$

We define some unit l 's error at time step t by

$$e_l(t) := -\frac{\partial E(t)}{\partial net_l(t)}. \tag{A.16}$$

Using (almost) standard backpropagation, we first compute updates for weights to output units ($l = k$), weights to hidden units ($l = i$) and weights

to output gates ($l = out_j$). We obtain (compare formulas A.8, A.9, and A.11):

$$I = k \text{ (output)} : e_k(t) = f'_k(net_k(t)) \left(t^k(t) - y^k(t) \right), \quad (\text{A.17})$$

$$I = i \text{ (hidden)} : e_i(t) = f'_i(net_i(t)) \sum_{k: k \text{ output unit}} w_{ki} e_k(t), \quad (\text{A.18})$$

$I = out_j$ (output gates) :

$$e_{out_j}(t) = f'_{out_j}(net_{out_j}(t)) \left(\sum_{v=1}^{S_j} h(s_{c_j^v}(t)) \sum_{k: k \text{ output unit}} w_{kc_j^v} e_k(t) \right). \quad (\text{A.19})$$

For all possible l time t 's contribution to w_{lm} 's update is

$$\Delta w_{lm}(t) = \alpha e_l(t) y^m(t-1). \quad (\text{A.20})$$

The remaining updates for weights to input gates ($l = in_j$) and to cell units ($l = c_j^v$) are less conventional. We define some internal state $s_{c_j^v}$'s error:

$$\begin{aligned} e_{s_{c_j^v}} &:= -\frac{\partial E(t)}{\partial s_{c_j^v}(t)} \\ &= f'_{out_j}(net_{out_j}(t)) h'(s_{c_j^v}(t)) \sum_{k: k \text{ output unit}} w_{kc_j^v} e_k(t). \end{aligned} \quad (\text{A.21})$$

We obtain for $l = in_j$ or $l = c_j^v$, $v = 1, \dots, S_j$

$$-\frac{\partial E(t)}{\partial w_{lm}} = \sum_{v=1}^{S_j} e_{s_{c_j^v}}(t) \frac{\partial s_{c_j^v}(t)}{\partial w_{lm}}. \quad (\text{A.22})$$

The derivatives of the internal states with respect to weights and the corresponding weight updates are as follows (compare expression A.12):

$$\begin{aligned} I = in_j \text{ (input gates)} : \\ \frac{\partial s_{c_j^v}(t)}{\partial w_{in_j m}} &= \frac{\partial s_{c_j^v}(t-1)}{\partial w_{in_j m}} + g(net_{c_j^v}(t)) f'_{in_j}(net_{in_j}(t)) y^m(t-1); \end{aligned} \quad (\text{A.23})$$

therefore, time t 's contribution to $w_{in_j m}$'s update is (compare expression A.8):

$$\Delta w_{in_j m}(t) = \alpha \sum_{v=1}^{S_j} e_{s_{c_j^v}}(t) \frac{\partial s_{c_j^v}(t)}{\partial w_{in_j m}}. \quad (\text{A.24})$$

Similarly we get (compare expression A.12):

$$I = c_j^y \text{ (memory cells)} : \\ \frac{\partial s_{c_j^y}(t)}{\partial w_{c_j^y m}} = \frac{\partial s_{c_j^y}(t-1)}{\partial w_{c_j^y m}} + g'(net_{c_j^y}(t)) f_{inj}(net_{inj}(t)) y^m(t-1); \quad (\text{A.25})$$

therefore time t 's contribution to $w_{c_j^y m}$'s update is (compare expression A.8):

$$\Delta w_{c_j^y m}(t) = \alpha e_{s_{c_j^y}}(t) \frac{\partial s_{c_j^y}(t)}{\partial w_{c_j^y m}}. \quad (\text{A.26})$$

All we need to implement for the backward pass are equations A.17 through A.21 and A.23 through A.26. Each weight's total update is the sum of the contributions of all time steps.

A.1.4 Computational Complexity. LSTM's update complexity per time step is

$$O(KH + KCS + HI + CSI) = O(W), \quad (\text{A.27})$$

where K is the number of output units, C is the number of memory cell blocks, $S > 0$ is the size of the memory cell blocks, H is the number of hidden units, I is the (maximal) number of units forward connected to memory cells, gate units and hidden units, and

$$W = KH + KCS + CSI + 2CI + HI = O(KH + KCS + CSI + HI)$$

is the number of weights. Expression A.27 is obtained by considering all computations of the backward pass: equation A.17 needs K steps; A.18 needs KH steps; A.19 needs KSC steps; A.20 needs $K(H+C)$ steps for output units, HI steps for hidden units, CI steps for output gates; A.21 needs KCS steps; A.23 needs CSI steps; A.24 needs CSI steps; A.25 needs CSI steps; A.26 needs CSI steps. The total is $K + 2KH + KC + 2KSC + HI + CI + 4CSI$ steps, or $O(KH + KSC + HI + CSI)$ steps. We conclude that LSTM algorithm's update complexity per time step is just like BPTT's for a fully recurrent net.

At a given time step, only the $2CSI$ most recent $\partial s_{c_j^y}/\partial w_{lm}$ values from equations A.23 and A.25 need to be stored. Hence LSTM's storage complexity also is $O(W)$; it does not depend on the input sequence length.

A.2 Error Flow. We compute how much an error signal is scaled while flowing back through a memory cell for q time steps. As a by-product, this analysis reconfirms that the error flow within a memory cell's CEC is indeed constant, provided that truncated backpropagation cuts off error flow trying to leave memory cells (see also section 3.2). The analysis also highlights a

potential for undesirable long-term drifts of s_j , as well as the beneficial, countermanding influence of negatively biased input gates.

Using the truncated backpropagation learning rule, we obtain

$$\begin{aligned}
 \frac{\partial s_j(t-k)}{\partial s_j(t-k-1)} &= 1 + \frac{\partial y^{inj}(t-k)}{\partial s_j(t-k-1)} g(\text{net}_j(t-k)) \\
 &\quad + y^{inj}(t-k) g'(\text{net}_j(t-k)) \frac{\partial \text{net}_j(t-k)}{\partial s_j(t-k-1)} \\
 &= 1 + \sum_u \left[\frac{\partial y^{inj}(t-k)}{\partial y^u(t-k-1)} \frac{\partial y^u(t-k-1)}{\partial s_j(t-k-1)} \right] \\
 &\quad \times g(\text{net}_j(t-k)) \\
 &\quad + y^{inj}(t-k) g'(\text{net}_j(t-k)) \\
 &\quad \times \sum_u \left[\frac{\partial \text{net}_j(t-k)}{\partial y^u(t-k-1)} \frac{\partial y^u(t-k-1)}{\partial s_j(t-k-1)} \right] \\
 &\approx_{tr} 1. \tag{A.28}
 \end{aligned}$$

The \approx_{tr} sign indicates equality due to the fact that truncated backpropagation replaces by zero the following derivatives:

$$\frac{\partial y^{inj}(t-k)}{\partial y^u(t-k-1)} \quad \forall u \quad \text{and} \quad \frac{\partial \text{net}_j(t-k)}{\partial y^u(t-k-1)} \quad \forall u.$$

In what follows, an error $\vartheta_j(t)$ starts flowing back at c_j 's output. We redefine

$$\vartheta_j(t) := \sum_i w_{ic_j} \vartheta_i(t+1). \tag{A.29}$$

Following the definitions and conventions of section 3.1, we compute error flow for the truncated backpropagation learning rule. The error occurring at the output gate is

$$\vartheta_{out_j}(t) \approx_{tr} \frac{\partial y^{out_j}(t)}{\partial \text{net}_{out_j}(t)} \frac{\partial \text{y}^g(t)}{\partial y^{out_j}(t)} \vartheta_j(t). \tag{A.30}$$

The error occurring at the internal state is

$$\vartheta_{s_j}(t) = \frac{\partial s_j(t+1)}{\partial s_j(t)} \vartheta_{s_j}(t+1) + \frac{\partial y^g(t)}{\partial s_j(t)} \vartheta_j(t). \tag{A.31}$$

Since we use truncated backpropagation we have

$$\vartheta_j(t) = \sum_{i: i \text{ no gate and no memory cell}} w_{ic_j} \vartheta_i(t+1);$$

therefore we get

$$\frac{\partial \vartheta_j(t)}{\partial \vartheta_{s_j}(t+1)} = \sum_i w_{ij} \frac{\partial \vartheta_i(t+1)}{\partial \vartheta_{s_j}(t+1)} \approx_{tr} 0. \quad (\text{A.32})$$

Equations A.31 and A.32 imply constant error flow through internal states of memory cells:

$$\frac{\partial \vartheta_{s_j}(t)}{\partial \vartheta_{s_j}(t+1)} = \frac{\partial s_j(t+1)}{\partial s_j(t)} \approx_{tr} 1. \quad (\text{A.33})$$

The error occurring at the memory cell input is

$$\vartheta_{c_j}(t) = \frac{\partial g(\text{net}_{c_j}(t))}{\partial \text{net}_{c_j}(t)} \frac{\partial s_j(t)}{\partial g(\text{net}_{c_j}(t))} \vartheta_{s_j}(t). \quad (\text{A.34})$$

The error occurring at the input gate is

$$\vartheta_{in_j}(t) \approx_{tr} \frac{\partial y^{in_j}(t)}{\partial \text{net}_{in_j}(t)} \frac{\partial s_j(t)}{\partial y^{in_j}(t)} \vartheta_{s_j}(t). \quad (\text{A.35})$$

A.2.1 No External Error Flow. Errors are propagated back from units l to unit v along outgoing connections with weights w_{lv} . This “external error” (note that for conventional units there is nothing but external error) at time t is

$$\vartheta_v^e(t) = \frac{\partial y^v(t)}{\partial \text{net}_v(t)} \sum_l \frac{\partial \text{net}_l(t+1)}{\partial y^v(t)} \vartheta_l(t+1). \quad (\text{A.36})$$

We obtain

$$\begin{aligned} \frac{\partial \vartheta_v^e(t-1)}{\partial \vartheta_j(t)} &= \frac{\partial y^v(t-1)}{\partial \text{net}_v(t-1)} \left(\frac{\partial \vartheta_{out_j}(t)}{\partial \vartheta_j(t)} \frac{\partial \text{net}_{out_j}(t)}{\partial y^v(t-1)} \right. \\ &\quad \left. + \frac{\partial \vartheta_{in_j}(t)}{\partial \vartheta_j(t)} \frac{\partial \text{net}_{in_j}(t)}{\partial y^v(t-1)} + \frac{\partial \vartheta_{c_j}(t)}{\partial \vartheta_j(t)} \frac{\partial \text{net}_{c_j}(t)}{\partial y^v(t-1)} \right) \\ &\approx_{tr} 0. \end{aligned} \quad (\text{A.37})$$

We observe that the error ϑ_j arriving at the memory cell output is not back-propagated to units v by external connections to in_j , out_j , c_j .

A.2.2 Error Flow Within Memory Cells. We now focus on the error back-flow within a memory cell’s CEC. This is actually the only type of error flow that can bridge several time steps. Suppose error $\vartheta_j(t)$ arrives at c_j ’s output

at time t and is propagated back for q steps until it reaches in_j or the memory cell input $g(net_{c_j})$. It is scaled by a factor of

$$\frac{\partial \vartheta_v(t-q)}{\partial \vartheta_j(t)},$$

where $v = in_j, c_j$. We first compute

$$\frac{\partial \vartheta_{s_j}(t-q)}{\partial \vartheta_j(t)} \approx_{tr} \begin{cases} \frac{\partial y^{f_j}(t)}{\partial s_j(t)} & q = 0 \\ \frac{\partial s_j(t-q+1)}{\partial s_j(t-q)} \frac{\partial \vartheta_{s_j}(t-q+1)}{\partial \vartheta_j(t)} & q > 0 \end{cases}. \quad (\text{A.38})$$

Expanding equation A.38, we obtain

$$\begin{aligned} \frac{\partial \vartheta_v(t-q)}{\partial \vartheta_j(t)} &\approx_{tr} \frac{\partial \vartheta_v(t-q)}{\partial \vartheta_{s_j}(t-q)} \frac{\partial \vartheta_{s_j}(t-q)}{\partial \vartheta_j(t)} \\ &\approx_{tr} \frac{\partial \vartheta_v(t-q)}{\partial \vartheta_{s_j}(t-q)} \left(\prod_{m=q}^1 \frac{\partial s_j(t-m+1)}{\partial s_j(t-m)} \right) \frac{\partial y^{f_j}(t)}{\partial s_j(t)} \\ &\approx_{tr} y^{out_j}(t) h'(s_j(t)) \begin{cases} g'(net_{c_j}(t-q)) y^{in_j}(t-q) & v = c_j \\ g(net_{c_j}(t-q)) f'_{in_j}(net_{in_j}(t-q)) & v = in_j \end{cases} \quad (\text{A.39}) \end{aligned}$$

Consider the factors in the previous equation's last expression. Obviously, error flow is scaled only at times t (when it enters the cell) and $t-q$ (when it leaves the cell), but not in between (constant error flow through the CEC). We observe:

1. The output gate's effect is $y^{out_j}(t)$ scales down those errors that can be reduced early during training without using the memory cell. It also scales down those errors resulting from using (activating/deactivating) the memory cell at later training stages. Without the output gate, the memory cell might, for instance, suddenly start causing avoidable errors in situations that already seemed under control (because it was easy to reduce the corresponding errors without memory cells). See "Output Weight Conflict" in section 3 and "Abuse Problem and Solution" (section 4.7).
2. If there are large positive or negative $s_j(t)$ values (because s_j has drifted since time step $t-q$), then $h'(s_j(t))$ may be small (assuming that h is a logistic sigmoid). See section 4. Drifts of the memory cell's internal state s_j can be countermanded by negatively biasing the input gate in_j (see section 4 and the next point). Recall from section 4 that the precise bias value does not matter much.
3. $y^{in_j}(t-q)$ and $f'_{in_j}(net_{in_j}(t-q))$ are small if the input gate is negatively biased (assume f_{in_j} is a logistic sigmoid). However, the potential sig-

nificance of this is negligible compared to the potential significance of drifts of the internal state s_q .

Some of the factors above may scale down LSTM's overall error flow, but not in a manner that depends on the length of the time lag. The flow will still be much more effective than an exponentially (of order q) decaying flow without memory cells.

Acknowledgments

Thanks to Mike Mozer, Wilfried Brauer, Nic Schraudolph, and several anonymous referees for valuable comments and suggestions that helped to improve a previous version of this article (Hochreiter and Schmidhuber, 1995). This work was supported by DFG grant SCHM 942/3-1 from Deutsche Forschungsgemeinschaft.

References

- Almeida, L. B. (1987). A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. In *IEEE 1st International Conference on Neural Networks, San Diego* (Vol. 2, pp. 609–618).
- Baldi, P., & Pineda, F. (1991). Contrastive learning and neural oscillator. *Neural Computation*, 3, 526–545.
- Bengio, Y., & Frasconi, P. (1994). Credit assignment through time: Alternatives to backpropagation. In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in neural information processing systems 6* (pp. 75–82). San Mateo, CA: Morgan Kaufmann.
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166.
- Cleeremans, A., Servan-Schreiber, D., & McClelland, J. L. (1989). Finite-state automata and simple recurrent networks. *Neural Computation*, 1, 372–381.
- de Vries, B., & Principe, J. C. (1991). A theory for neural networks with time delays. In R. P. Lippmann, J. E. Moody, & D. S. Touretzky (Eds.), *Advances in neural information processing systems 3*, (pp. 162–168). San Mateo, CA: Morgan Kaufmann.
- Doya, K. (1992). Bifurcations in the learning of recurrent neural networks. In *Proceedings of 1992 IEEE International Symposium on Circuits and Systems* (pp. 2777–2780).
- Doya, K., & Yoshizawa, S. (1989). Adaptive neural oscillator using continuous-time backpropagation learning. *Neural Networks*, 2, 375–385.
- Elman, J. L. (1988). *Finding structure in time* (Tech. Rep. No. CRL 8801). San Diego: Center for Research in Language, University of California, San Diego.
- Fahlman, S. E. (1991). The recurrent cascade-correlation learning algorithm. In R. P. Lippmann, J. E. Moody, & D. S. Touretzky (Eds.), *Advances in neural information processing systems 3* (pp. 190–196). San Mateo, CA: Morgan Kaufmann.

- Hochreiter, J. (1991). *Untersuchungen zu dynamischen neuronalen Netzen*. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München. See <http://www7.informatik.tu-muenchen.de/~hochreit/>.
- Hochreiter, S., & Schmidhuber, J. (1995). *Long short-term memory* (Tech. Rep. No. FKI-207-95). Fakultät für Informatik, Technische Universität München.
- Hochreiter, S., & Schmidhuber, J. (1996). Bridging long time lags by weight guessing and “long short-term memory.” In F. L. Silva, J. C. Principe, & L. B. Almeida (Eds.), *Spatiotemporal models in biological and artificial systems* (pp. 65–72). Amsterdam: IOS Press.
- Hochreiter, S., & Schmidhuber, J. (1997). LSTM can solve hard long time lag problems. In *Advances in neural information processing systems 9*. Cambridge, MA: MIT Press.
- Lang, K., Waibel, A., & Hinton, G. E. (1990). A time-delay neural network architecture for isolated word recognition. *Neural Networks*, 3, 23–43.
- Lin, T., Horne, B. G., Tino, P., & Giles, C. L. (1996). Learning long-term dependencies in NARX recurrent neural networks. *IEEE Transactions on Neural Networks*, 7, 1329–1338.
- Miller, C. B., & Giles, C. L. (1993). Experimental comparison of the effect of order in recurrent neural networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4), 849–872.
- Mozer, M. C. (1989). A focused back-propagation algorithm for temporal sequence recognition. *Complex Systems*, 3, 349–381.
- Mozer, M. C. (1992). Induction of multiscale temporal structure. In J. E. Moody, S. J. Hanson, & R. P. Lippman (Eds.), *Advances in neural information processing systems 4* (pp. 275–282). San Mateo, CA: Morgan Kaufmann.
- Pearlmutter, B. A. (1989). Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1(2), 263–269.
- Pearlmutter, B. A. (1995). Gradient calculations for dynamic recurrent neural networks: A survey. *IEEE Transactions on Neural Networks*, 6(5), 1212–1228.
- Pineda, F. J. (1987). Generalization of back-propagation to recurrent neural networks. *Physical Review Letters*, 59(19), 2229–2232.
- Pineda, F. J. (1988). Dynamics and architecture for neural computation. *Journal of Complexity*, 4, 216–245.
- Plate, T. A. (1993). Holographic recurrent networks. In S. J. Hanson, J. D. Cowan, & C. L. Giles (Eds.), *Advances in neural information processing systems 5* (pp. 34–41). San Mateo, CA: Morgan Kaufmann.
- Pollack, J. B. (1991). Language induction by phase transition in dynamical recognizers. In R. P. Lippmann, J. E. Moody, & D. S. Touretzky (Eds.), *Advances in neural information processing systems 3* (pp. 619–626). San Mateo, CA: Morgan Kaufmann.
- Puskorius, G. V., and Feldkamp, L. A. (1994). Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks. *IEEE Transactions on Neural Networks*, 5(2), 279–297.
- Ring, M. B. (1993). Learning sequential tasks by incrementally adding higher orders. In S. J. Hanson, J. D. Cowan, & C. L. Giles (Eds.), *Advances in neural information processing systems 5* (pp. 115–122). San Mateo, CA: Morgan Kaufmann.

- Robinson, A. J., & Fallside, F. (1987). *The utility driven dynamic error propagation network* (Tech. Rep. No. CUED/F-INFENG/TR.1). Cambridge: Cambridge University Engineering Department.
- Schmidhuber, J. (1989). A local learning algorithm for dynamic feedforward and recurrent networks. *Connection Science*, 1(4), 403–412.
- Schmidhuber, J. (1992a). A fixed size storage $O(n^3)$ time complexity learning algorithm for fully recurrent continually running networks. *Neural Computation*, 4(2), 243–248.
- Schmidhuber, J. (1992b). Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2), 234–242.
- Schmidhuber, J. (1992c). Learning unambiguous reduced sequence descriptions. In J. E. Moody, S. J. Hanson, & R. P. Lippman (Eds.), *Advances in neural information processing systems 4* (pp. 291–298). San Mateo, CA: Morgan Kaufmann.
- Schmidhuber, J. (1993). *Netzwerkarchitekturen, Zielfunktionen und Kettenregel*. Habilitationsschrift, Institut für Informatik, Technische Universität München.
- Schmidhuber, J., & Hochreiter, S. (1996). *Guessing can outperform many long time lag algorithms* (Tech. Rep. No. IDSIA-19-96). Lugano, Switzerland: Instituto Dalle Molle di Studi sull'Intelligenza Artificiale.
- Silva, G. X., Amaral, J. D., Langlois, T., & Almeida, L. B. (1996). Faster training of recurrent networks. In F. L. Silva, J. C. Principe, & L. B. Almeida (Eds.), *Spatiotemporal models in biological and artificial systems* (pp. 168–175). Amsterdam: IOS Press.
- Smith, A. W., & Zipser, D. (1989). Learning sequential structures with the real-time recurrent learning algorithm. *International Journal of Neural Systems*, 1(2), 125–131.
- Sun, G., Chen, H., & Lee, Y. (1993). Time warping invariant neural networks. In S. J. Hanson, J. D. Cowan, & C. L. Giles (Eds.), *Advances in neural information processing systems 5* (pp. 180–187). San Mateo, CA: Morgan Kaufmann.
- Watrous, R. L., & Kuhn, G. M. (1992). Induction of finite-state languages using second-order recurrent networks. *Neural Computation*, 4, 406–414.
- Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1, 339–356.
- Williams, R. J. (1989). *Complexity of exact gradient computation algorithms for recurrent neural networks* (Tech. Rep. No. NU-CCS-89-27). Boston: Northeastern University, College of Computer Science.
- Williams, R. J. & Peng, J. (1990). An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural Computation*, 4, 491–501.
- Williams, R. J., & Zipser, D. (1992). Gradient-based learning algorithms for recurrent networks and their computational complexity. In Y. Chauvin, & D. E. Rumelhart (Eds.), *Back-propagation: Theory, architectures and applications*. Hillsdale, NJ: Erlbaum.

Received August 28, 1995; accepted February 24, 1997.

PC-GNN: Pearson Correlation-Based Graph Neural Network for Recognition of Human Lower Limb Activity Using sEMG Signal

Ankit Vijayvargiya^{ID}, Senior Member, IEEE, Rajesh Kumar^{ID}, Senior Member, IEEE, and Parul Sharma^{ID}

Abstract—Artificial intelligence has a plethora of applications in the realm of biomedical sciences, such as pattern recognition, diagnosis of disease, human–machine interaction, medical image processing, robotic limbs, or exoskeletons. Robotic limbs, or exoskeletons, are widely employed to assist with lower limb movement. To increase the exoskeleton’s flexibility in the lower extremities, it is critical to recognize the diverse motion intents of the lower limbs of the human body. In this investigation, sEMG signals from lower limb muscles are used for a computer-aided recognition system to correctly identify the lower limb activities because these signals can identify movement ahead of time and enable faster detection of signal fluctuation than other wearable sensors. Several types of noise are introduced into the signal during collection. A multistage classification strategy is proposed to overcome the processing challenges associated with these sEMG signals. Initially, nine time-domain handcrafted features are retrieved using a hybrid of wavelet denoising and ensemble empirical mode decomposition approach with a sliding window of 256 ms and a 25% overlap. Next, a Pearson correlation-based graph is formed from the extracted features and applied to a graph neural network (GNN). GNN not only captures individual information but also makes use of information from other samples to form a graph. The combination of a Pearson correlation-based graph with a GNN is referred to as Pearson correlation-based GNN. The observation states that the approach proposed in the research achieved an accuracy of 99.19%, 99.02%, 96.21% for the walking, sitting, and standing of healthy subjects, while 99.29%, 97.97%, 99.36% for the subjects comprising knee abnormalities, respectively.

Index Terms—Biomedical signal processing, ensemble empirical mode decomposition (EMD), graph neural network (GNN), handcrafted feature extraction, human lower limb activity recognition, wavelet denoising.

I. INTRODUCTION

THE research on monitoring human activities using sensor devices has been attention-grabbing for the past decade.

Manuscript received 9 October 2022; revised 28 August 2023; accepted 22 September 2023. This article was recommended by Associate Editor Y. Li. (*Corresponding author: Ankit Vijayvargiya*.)

Ankit Vijayvargiya is with the Department of Electrical Engineering, Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur 302017, India (e-mail: ankitvijayvargiya29@gmail.com).

Rajesh Kumar and Parul Sharma are with the Department of Electrical Engineering, Malaviya National Institute of Technology, Jaipur 302017, India (e-mail: rkumar.ee@mnit.ac.in; parulrma24@gmail.com).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/THMS.2023.3319356>.

Digital Object Identifier 10.1109/THMS.2023.3319356

Human activities recognition (HAR) uses sensors to monitor data that has been learned and performed in the relevant context to detect a person’s activity. There are several uses for HAR in various fields, including teleimmersion, healthcare monitoring, and smart homes with aided surveillance [1]. Usually, the HAR procedure is attainable by employing various steps, from working on the recorded data, utilizing various sensors, to finally classifying the activity to be performed. An HAR system using various sensory devices can track the lower limb activities of humans. To identify human posture, physical activity state, and behavioral activities, HAR technology often uses multiple multimodal data originating from numerous hardware sensors. Currently, HAR research can be broadly categorized into several situations supported by various technologies, including video, wearable, mobile phone sensors, social networks, and wireless signal [2], [3], [4].

With the advancements in sensor technology, many devices, such as accelerometers, electromyography (EMG), gyroscopes, and barometers, are available to capture activity data [5], [6]. These sensing devices may be infiltrated into wearable devices to acquire data via wearable sensors. For this purpose, the EMG technique has been taken into account, such that it records electrical signals during contraction in skeletal muscles in the lower limbs. Among the techniques of wearable sensors, the EMG technique is better for classifying the activity of the lower limbs because it can identify the movements ahead of time [7]. There are two methods for recording EMG: one uses concentric needle electrodes, known as intramuscular electromyography (iEMG), and the other uses surface electrodes, known as surface electromyography (sEMG) [8]. The signals from the iEMG approach are taken by penetrating a needle into the muscle, which becomes a painful task and can become harmful if the virus gets into the body, whereas the sEMG sensor is used preferably for the gesture of the user and prosthetic control because electrodes are placed on the surface of the body and make it easy to capture the signals [9], [10].

Noise sources always contaminate the EMG signal of a muscle. Its properties are influenced by the subject’s internal anatomy, including blood flow velocity, skin formation, and temperature. The sequence of EMG is determined by the physiological/anatomical features of muscles, making it challenging to assess and identify EMG signals. Several kinds of electrical noise affecting EMG signals are discussed in [7], [11], [12]. The

main obstacles that must be overcome to attain higher performance in the previously mentioned applications are noisy EMG signals. Recent studies have effectively used novel techniques to minimize the noise in sEMG data [13], [14]. Vijayvargiya and colleagues created a hybrid approach using wavelet denoising with ensemble empirical mode decomposition (WD-EEMD) to remove noise and artifacts from the sEMG signal [15], [16].

For automatic control of prostheses or exoskeletons, the sEMG signals from the muscles are captured by employing sensors by performing various activities, and then the actuators are operated using artificial intelligence techniques. This automatic control of a lower limb prosthetic or exoskeleton has risen in popularity over the last decade due to advances in computer vision, artificial intelligence methods, the availability of wearable sensors, and the Internet of Things. There are several applications in which researchers have focused on recognizing lower limb actions. For example, if a lower limb is damaged as a result of a tragedy, medical sickness, or surgery requiring physiatry, these types of applications require robotic limbs that may be controlled via the classification of lower limb activities. Many pieces of research use different machine learning and deep learning algorithms for activity recognition, considering sensory data collected from the human lower limb. In the machine learning algorithms, handcrafted features are required to extract from the raw sEMG signal and can then be applied to the classification task. For various lower limb actions, Naik and colleagues [17] effectively used a multivariate approach known as independent component analysis by entropy bound minimization (ICA-EBM). It uses a linear discriminant analysis classifier to accurately categorize the walking, standing, and sitting behaviors of a few healthy as well as affected, with the correctness of 96.1% and 86.2%, respectively. A new approach [15] to carrying out preprocessing named WD-EEMD is proposed for the purpose of classification of activities associated with the lower limb. The different noises inextricably linked to the sEMG signal are filtered out using WD and EEMD together. The classification is carried out with 90.69% accuracy for healthy subjects and 97.45% for subjects having knee-related abnormalities.

The handcrafted feature-based technique has various shortcomings, including the inability to learn the features from the data by itself and the use of classifiers that only offer a superficial representation of the data. On the contrary, We do not need to manually extract features from the data while using a deep learning-based strategy. During training, the network picks up how to extract features. You merely supply the network with the values. Using a voting-based 1-D convolutional neural network (CNN), Vijayvargiya et al. [18] identified three unique proceedings for respective cases: healthy data; knee abnormal data; and pooled data, or a merged dataset of healthy and knee abnormal participants with a classification accuracy of 99.35, 97.63, and 97.14%, respectively. When calculating the prognosis of the knee joint angle, Gautam et al. [19] created a unique categorization system that takes lower limb motions into account. It is suggested to combine CNN and LSTM to create an architecture for recognizing activities. CNN would gather

features from the signal, while LSTM would anticipate joint angles and interpret the information.

The primary concern is identifying activity related to the human lower limb with optimum efficiency. In this investigation, a combination of handcrafted features and deep learning technique [graph neural network (GNN)] is considered to classify the lower limb activities using sEMG signals. For the past few years, many researchers have utilized this novel concept called GNN to cover different fields, such as recommendation systems [20], traffic forecasting [21], behavior generation in semantic environments [22], language modeling, image classification, and human activity recognition [23]. GNN not only captures the individual's information but also collects the information of other samples by forming a graph. Here, a graph is formed based on the Pearson correlation of the extracted handcraft features based on the WD-EEMD approach and then applied GNN for the classification of the distinct activities. This combination of graph formation and classification is considered as a Pearson correlation-based graph neural network (PC-GNN), and it achieved high accuracy for the classification of distinct activities in both healthy and knee-abnormal subjects. The significant contributions to this investigation are as follows.

- 1) Activity recognition of lower limb in healthy and knee abnormal individuals using sEMG signals from leg muscles.
- 2) Nine time-domain (TD) features based on WD-EEMD are derived from sEMG signals with an overlapping windowing technique.
- 3) A PC-GNN is proposed for classifying lower limb activities, in which a graph is formed based on the co-relationship of the extracted features, with nodes representing the labels.
- 4) A quantitative analysis of the performance of the correlation-based GNN is evaluated and compared with previous studies.

The rest of this article is organized as follows. Section II demonstrates the comprehensive dataset information. The methodology employed can be perceived in Section III. Examined results are discussed in Section IV. Finally, Section V concludes this article.

II. DATASET

The in-use dataset was reaped from the so-called open-source repository named UCI machine learning by Sanchez et al. [24]. The collection of sEMG signal data was performed on 22 subjects with an average age of more than 18 years, and they were made to perform activities associated with their lower limbs, such as walking, sitting, and standing. In the dataset, there are signals from both types of subjects: healthy people and people who have problems with their knees. Four individuals with knee anomalies were impacted with a meniscus injury, six suffered from anterior cruciate ligament, and one-faced sciatic nerve damage. The signals from the lower limb muscles are monitored and collected by the sEMG signal sensors, and a goniometer was also used in the data collection procedure. The wearable sensors are placed on four different lower limb muscles: vastus medialis,

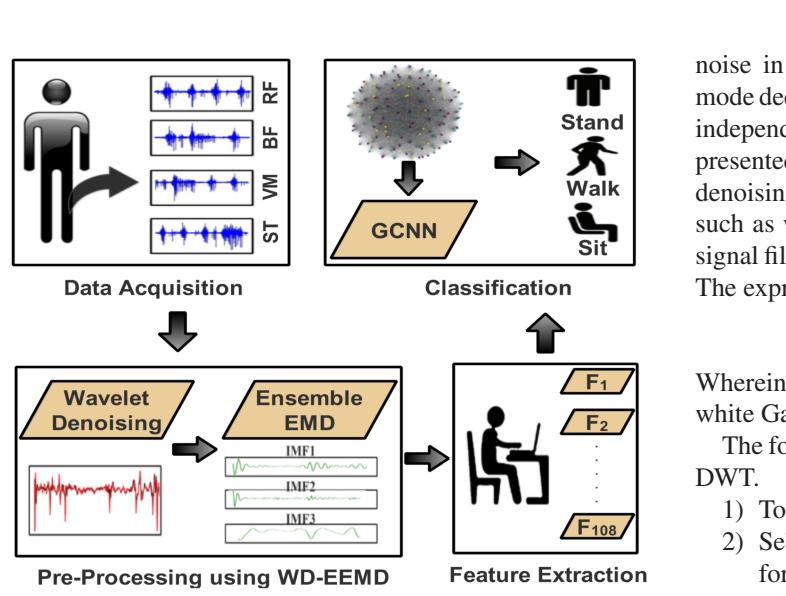


Fig. 1. Block diagram of lower limb activity recognition.

biceps femoris, semitendinosus (ST), and rectus femoris (RF). This study aims to examine the effect of EMG signals from lower limb muscles on lower limb movements and, hence, solely address sEMG signals. The signal data were collected from the 11 healthy subjects' left limbs and the 11 unhealthy subjects' damaged limbs. It did not include transition periods, such as the transition from sitting to standing and walking to sitting. The resolution possessed by the data is 14 bit, with a sampling frequency of 1000 Hz, and then moved through a band-pass filter with a frequency of 20–460 Hz. Therefore, it was then transferred from the MWX8 device to the computer.

III. PROPOSED METHODOLOGY

This section describes the methods recommended for recognizing lower limb activities using the sEMG signal, as shown in Fig. 1. The investigation used publicly available datasets from the UCI machine learning library. Initially, the sEMG signal is preprocessed, which includes noise removal using a WD-EEMD technique. Furthermore, the data are segmented using the overlapping windowing technique, nine TD features are extracted, and a graph is formed based on feature co-relationships. Finally, a GNN predicts lower limb activities in healthy and knee anomaly subjects. The following sections provide a complete overview of the approaches used in developing the proposed methodology.

A. WD-EEMD-Based Preprocessing

Due to the blending of different noise signals and artifacts, such as movement artifacts, intrinsic noise in electrodes, electromagnetic noise, and others, the identification of a genuine sEMG signal originating in the lower limb muscles is lost. These noises or artifacts may impact the handcrafted feature extraction. Therefore, signal denoising is a crucial step that needs to be carried out before the signals are used for any purpose. Recent studies have effectively used novel techniques to minimize the

noise in sEMG data, including wavelet denoising, empirical mode decomposition (EMD), principal component analysis, and independent component analysis (ICA). Phinyomark et al. [25] presented a method to denoise the sEMG signal using a wavelet denoising technique. Several irregular sounds in sEMG signals, such as white Gaussian noises, are difficult to eliminate using signal filtering. Hence, wavelet denoising is used to eliminate it. The expression for a noisy signal is stated as

$$X(n) = Y(n) + K(n). \quad (1)$$

Wherein $Y(n)$ demonstrates the original signal, $K(n)$ signifies white Gaussian noise, and $X(n)$ represents a noisy signal.

The following stages make up the wavelet denoising based on DWT.

- 1) To decompose signals, use the DWT.
- 2) Select a threshold value for the respective wavelet transform.
- 3) The inverse wavelet transform is used with the threshold function to regenerate the denoised signal.

When wavelet denoising is applied to signals, discrete wavelet coefficients are produced when the signal is processed through low-pass and high-pass filters. After signal decomposition and thresholding, details and approximation coefficients are gathered using the wavelet denoising technique. The degree of decomposition determines the total number of coefficients. There are two common methods for thresholding wavelet coefficients: soft and hard [26]. In contrast to hard thresholding, which only provides some information about the original signal, soft thresholding involves a continual departure from the original signal. This work implements universal thresholding [27] across fine-grained coefficients. The form of universal thresholding selection is represented as

$$\lambda = \sigma \sqrt{2 \ln(M)}. \quad (2)$$

Herein, $\sigma = (\text{MAD})/0.6745$ MAD is designated as the median absolute deviation, and M is the length of the signal.

The wavelet denoising method used in this study is combined with sym4, a member of the symlet family, at the beginning of the decomposition process [3]. The mean-squared error, mean absolute error, signal-to-noise ratio (SNR), and peak SNR of the sEMG signal were calculated in the earlier work using various mother wavelets and levels of decomposition on the same dataset. In comparison to various mother wavelets and degrees of decomposition, they claim that sym4 from the family of symlets to the first level of decomposition produces the greatest results.

The decomposition technique can mislead due to the non-linear, nonstationary, and non-Gaussian behavior of the sEMG signal since the assumption considered is linear and stationary. The decomposition technique employed in the research can disintegrate signals possessing nonlinear and nonstationary behavior with complex spatial and temporal features into complete or nearly orthogonal components named intrinsic mode functions. A single-component function or oscillation mode with a single instantaneous frequency is known as an IMF. Using the EMD method, a given signal may be iteratively broken down into several IMFs. However, due to the EMD algorithm's sensitivity to noise, it suffers from frequent appearance mode mixing. The

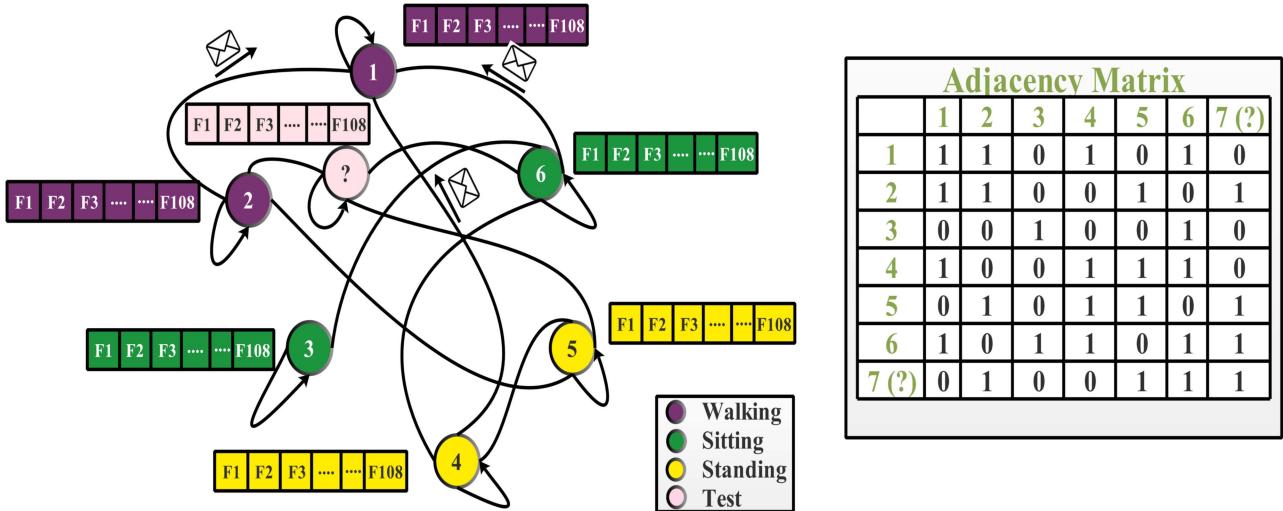


Fig. 2. Graph representation with its adjacency matrix.

EEMD approach for noise-assisted data analysis was presented to address this issue, which defines IMFs as the average of an ensemble of trials.

This investigation uses a preprocessing method that combines WD-EEMD, as it was in our prior study [15]. Wavelet denoising is used to initially filter out unwanted signals, such as noise from other muscles and white Gaussian noise. Ensemble EMD is then used to decompose the data and remove power line interference and baseline wandering noise.

B. Segmentation

As a clever method of processing time-series data used in a deep learning-based model to cut down on computational complexity, segmentation is used as preprocessing. The windowing approach is used to put the segmentation process into practice. Two distinct windowing techniques are available for the segmentation process: adjacent windowing and overlapping windowing. The size of a signal's window will affect the classification accuracy and reaction time. The model becomes more accurate with a greater window size, but the reaction time also grows. In this study, an overlapping windowing technique, a window size of 256 ms, and an overlapping of 25% are considered as per the previous studies [3], [17].

C. Feature Extraction

Feature extraction is generally achieved in three ways: TD, frequency-domain (FD), and time-frequency domain. Applications that require classification of signal are activity recognition of muscles, on-set detection, and contraction of muscle using TD features. The performance of these features in classifying signals is superior in environments possessing low noise and curtailed computational complexity [28]. Therefore, in the research classification of movement provided by the lower limb is conducted, nine features are utilized as mean absolute value, root-mean-square, slope sign changes, difference absolute standard deviation value, kurtosis, variance, skewness, zero crossing,

and average amplitude change [15]. Accordingly, three IMFs of four muscles each with nine handcrafted features create a total of 108 features ($4 \text{ muscles} \times 3 \text{ IMF} \times 9 \text{ handcrafted features} = 108 \text{ features}$). Fig. 3 shows the pair plot for five extracted features: Mean absolute value, zero crossing, slope sign change, skewness, and kurtosis of the first IMF of the RS muscle of the healthy subject.

D. Graph Neural Network

A graph represents a data structure composed of nodes (or vertices) and edges, i.e., $G = (V, E)$ where V and E denote a set of nodes and edges between the nodes, respectively. If n is the total number of nodes and m is the total number of features possessed by each node, then X represents the node feature matrix of size $n \times m$. The nodes in this proposed study are the three distinct activities of walking, standing, and sitting, as indicated by the purple, yellow, and green circles, as shown in Fig. 2 and edges of this graph are formed based on the Pearson correlations between all the extracted features. If the correlation value is more significant than 0.95, the edge between the nodes is considered; otherwise, no edge is considered. Here, a total of seven nodes are shown in Fig. 2 and their adjacency matrix is also presented. Here, node 1 is related to nodes 2, 4, and 6, which indicates that the Pearson correlation between these nodes is more significant than 0.95. However, the Pearson correlation between node 1 and nodes 3, 5, and 7 is less than 0.95, suggesting that there is not a connection between these nodes. The actual graph formed with the extracted handcraft features of the sEMG signal of healthy subject 1 is shown in Fig. 4.

Correlation coefficients are a statistical technique. It is used to determine the linear correlation between two variables precisely, thus indicating the degree of correlation. The Pearson correlation coefficient between two nonzero real-valued random variables is m_1 and m_2 is defined as follows [29]:

$$p(m_1, m_2) = \frac{\text{cov}(m_1, m_2)}{\sigma_{m_1} \sigma_{m_2}} \quad (3)$$



Fig. 3. Pair plots for five extracted features of walking, sitting, and standing state of healthy subject.

where cov is the covariance, σ_{m_1} and σ_{m_2} are the standard deviation of the features m_1 and m_2 , respectively.

GNN layers are comprised of a series of stages that are executed on each and every node in the graph as follows.

- 1) Message passing.
- 2) Aggregation.
- 3) Update.

The idea of message passing is a powerful concept, so it helps to comprehend graph algorithms. Nodes in the graph can send out messages about themselves to their neighbors, and the following nodes collect the neighbor messages they receive. Each node updates itself by collectively using its own messages and neighboring messages to update itself and understand its

environment. This is a simple form of message passing where the messages are the feature vectors. It is performed for entire input features employing a GNN, which is a simple message-passing algorithm.

To carry out the GraphConv layer, initialize the target node (the node that will be updated) with the node feature vector X_i . The feature vector of the target node h_i gets updated by receiving all of the attribute vectors of its directly connected node h_j and by applying an aggregation function with $N(i)$ as the number of neighborhood nodes and then passing this aggregated vector through a dense neural network layer. The output of the dense layer is the new representation of the node, which is the node embedding.

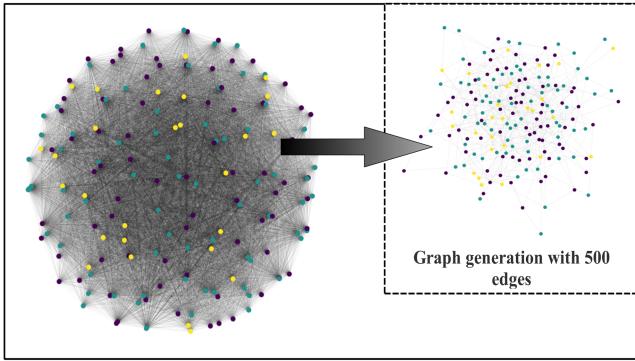


Fig. 4. Graph representation of healthy subject 1.

1) Feature Vector of a Node at Initial Layer:

$$h_i^0 = X_i. \quad (4)$$

2) Neighborhood Aggregation and Neural Network Layer:

$$h_i^t = \sigma \left(W_t \sum \frac{h_j^{t-1}}{N(i)} + B_t h_i^{t-1} \right) \quad (5)$$

where $\sum \frac{h_j^{t-1}}{N(i)}$ shows the aggregation of the feature vectors of the neighboring nodes of target node i and $B_t h_i^{t-1}$ is to include self-connection in that neighborhood aggregation.

3) Node Embedding at Last Layer:

$$z_i = h_i^T \quad (6)$$

where z_i is the node embedding of node i at the final layer. The trainable weight and bias matrix are W_t and B_t , respectively. The node feature vector is updated through an iterative process and can iterate the GraphConv layer over t number of iterations. So, at t th GraphConv layer node embedding h_i^t consists of information from the neighboring nodes, which are distantly t -hops. Here, sigmoid is a nonlinear function, and \sum is used to aggregate neighboring feature vectors.

To work with GNN, there is a step at the beginning of each layer in which node feature vector updation is to be accomplished. All the nodes are simultaneously updated in parallel. It can also be better realized by understanding matrix multiplication by considering its adjacency and degree matrix. The fundamental to the study of graphs is the adjacency matrix, which is a data structure in the form of a matrix A of size $n \times n$ that encodes all of the connections in the graph between nodes. To add self-connections or a node's own value for each node, add an identity matrix I to the adjacency matrix A , which basically means that each node has edge connectivity itself. The adjacency matrix can be used to implement message passing, which is a key concept in understanding graph neural network techniques

$$\hat{A} = A + I \quad (7)$$

$$\tilde{A} = \hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2} X^{t-1} \quad (8)$$

where \hat{D} is the degree matrix having the same size as the adjacency matrix. It is the diagonal matrix with diagonal elements

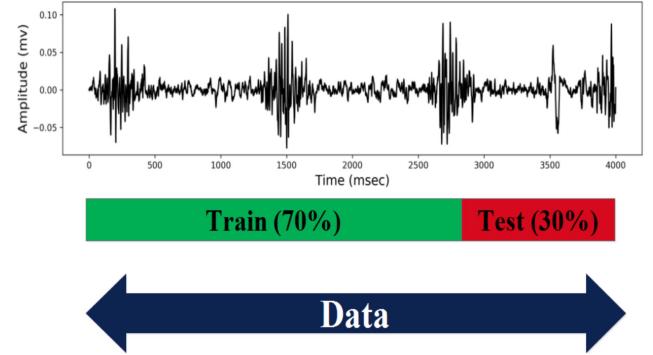


Fig. 5. Train-test split.

filled with the neighborhood size of each node, which is then calculated as the sum of the adjacency matrix across the columns or rows that gives the total number of connected edges to each node. Matrix multiplication in (8) aggregates the feature vectors of neighboring nodes. This will give a new feature matrix of the same shape as the original, but now, each value represents the average of the connected neighborhoods of each node. Passing this averaged feature matrix through a dense neural network layer results in the final representation of the node. To perform the next GraphConv layer, the form of the forward model is as follows:

$$Z = f(X, A) = \text{Softmax}(\tilde{A} \text{ReLU}(\tilde{A} X W^{(0)}) W^{(1)}) \quad (9)$$

where \tilde{A} is scaled adjacency matrix, X is feature vector matrix, $W^{(0)}$ and $W^{(1)}$ are the trainable weights.

The node embedding is then used to perform various tasks, such as node classification, link prediction, graph classification, and clustering. In this study, node classification is performed. The output of the final GNN layer can be used to predict the labels by adjusting the output dimensionality accordingly and using an entropy loss function with backpropagation to train parameters.

IV. EXPERIMENTATION

The current study utilizes sEMG data to classify lower limb movement using a GNN. Eleven healthy volunteers and eleven individuals with anomaly knees were examined, and performance metrics for three distinct activities were calculated: walking, sitting, and standing. The individuals' activity signals were preprocessed with WD-EEMD, and nine TD features were extracted. For signal segmentation, an overlapping window is chosen that holds 256 ms of window size and 25% overlap. Following that, a graph based on co-relations is formed from the extracted features, followed by training a GNN using training data and determining the model's performance parameters over the testing data.

The training dataset is composed of 70% of each subject's sEMG signal, which is a time-series signal, while the testing dataset is composed of the remaining 30% of each subject's signal, as shown in Fig. 5.

TABLE I
MATHEMATICAL EXPRESSION OF PERFORMANCE MEASURES

Performance measures	Walking	Sitting	Standing
Accuracy	$A_W = \frac{C_{WW}}{C_{WW} + C_{WS} + C_{WT}}$	$A_S = \frac{C_{SS}}{C_{SW} + C_{SS} + C_{ST}}$	$A_T = \frac{C_{TT}}{C_{TW} + C_{TS} + C_{TT}}$
Specificity	$SP_W = \frac{C_{SS} + C_{ST} + C_{TS} + C_{TT}}{C_{SW} + C_{SS} + C_{ST} + C_{TW} + C_{TS} + C_{TT}}$	$SP_S = \frac{C_{WW} + C_{WT} + C_{TW} + C_{TT}}{C_{WW} + C_{WS} + C_{WT} + C_{TW} + C_{TS} + C_{TT}}$	$SP_T = \frac{C_{WW} + C_{WS} + C_{SW} + C_{SS}}{C_{WW} + C_{WS} + C_{WT} + C_{SW} + C_{SS} + C_{ST}}$
Sensitivity	$SE_W = \frac{C_{WW}}{C_{WW} + C_{WS} + C_{WT}}$	$SE_S = \frac{C_{SS}}{C_{SW} + C_{SS} + C_{ST}}$	$SE_T = \frac{C_{TT}}{C_{TW} + C_{TS} + C_{TT}}$
Precision	$P_W = \frac{C_{WW}}{C_{WW} + C_{SS} + C_{TW}}$	$P_S = \frac{C_{SS}}{C_{WS} + C_{SS} + C_{TS}}$	$P_T = \frac{C_{TT}}{C_{WT} + C_{ST} + C_{TT}}$
F-score	$F_W = \frac{2 * SE_W * P_W}{SE_W + P_W}$	$F_S = \frac{2 * SE_S * P_S}{SE_S + P_S}$	$F_T = \frac{2 * SE_T * P_T}{SE_T + P_T}$

A. Performance Measures

True positive, false positive, true negative, and false negative are the four classification outcomes. In addition, these outcomes will aid in the development of a confusion matrix. The confusion matrix illustrates the performance of the model over the test dataset. This problem has three classes: walking: W , standing: T , and sitting: S . The below shown confusion matrix takes the following form:

$$C = \begin{bmatrix} C_{WW} & C_{WS} & C_{WT} \\ C_{SW} & C_{SS} & C_{ST} \\ C_{TW} & C_{TS} & C_{TT} \end{bmatrix} \quad (10)$$

where C_{WW}, C_{SS}, C_{TT} : Correct prediction; $C_{WS}, C_{WT}, C_{SW}, C_{ST}, C_{TW}, C_{TS}$: Wrong prediction.

The mathematical expression of the parameters for measuring the performance in this study is shown in Table I [15].

B. Setting of Model Parameters

The proposed model was trained and evaluated on Google Colab's cloud-based platform. The model is trained using a 16-batch minibatch gradient descent algorithm. The Adam optimizer employed is composed of an initial learning rate with a value of 0.01 and a weight decay with a value of $5e^{-4}$. In this study, the dataset was split into training sets and testing sets, with a value of 70% and 30%, respectively, and trained the GNN model with 500 iterations for ten runs.

C. Results and Discussions

Table II shows the subject-specific performance parameters, as defined in Table I of the GNN model for healthy and knee abnormal subjects. Even while accuracy may be regarded as one of the evaluation criteria, this study also includes additional metrics including sensitivity, specificity, precision, and F-score. The average classification accuracy, specificity, sensitivity, and F-score values are 98.67%, 99.36%, 98.14%, and 98.13% for healthy subjects and 98.92%, 99.43%, 98.88%, and 98.90% for knee abnormal subjects, respectively.

Table III exhibits accuracy for each subject individually for all three activities known as walking, sitting, and standing, by

TABLE II
SUBJECT-SPECIFIC PERFORMANCE INDICES OF PC-GNN EXPRESSED AS A PERCENTAGE FOR HEALTHY AND KNEE ABNORMAL INDIVIDUALS

Subjects	Normal				Abnormal			
	Acc	Spec	Sens	F-score	Acc	Spec	Sens	F-score
S1	98.93	99.49	98.98	98.71	98.71	99.28	98.58	98.71
S2	98.56	99.37	98.75	98.21	98.91	99.39	98.87	98.93
S3	96.15	98.28	94.41	94.18	99.62	99.81	99.40	99.52
S4	98.28	99.17	97.65	97.72	99.90	99.96	99.93	99.89
S5	99.14	99.56	99.22	99.06	100.00	100.00	100.00	100.00
S6	98.37	99.14	97.48	97.67	99.36	99.67	99.34	99.36
S7	100.00	100.00	100.00	100.00	98.46	99.22	98.64	98.48
S8	97.76	98.96	97.39	97.24	98.79	99.42	98.72	98.69
S9	99.07	99.48	97.79	98.36	97.25	98.39	97.20	97.46
S10	99.48	99.72	98.11	98.62	97.76	98.91	97.67	97.60
S11	99.62	99.80	99.76	99.71	99.30	99.65	99.28	99.29
Mean	98.67	99.36	98.14	98.13	98.92	99.43	98.88	98.90

TABLE III
SUBJECT-SPECIFIC ACCURACY OF PC-GNN EXPRESSED AS A PERCENTAGE FOR HEALTHY AND KNEE ABNORMAL INDIVIDUALS DURING WALKING, SITTING, AND STANDING

Subjects	Normal			Abnormal		
	Walking	Sitting	Standing	Walking	Sitting	Standing
S1	98.56	99.62	98.75	99.20	96.96	99.57
S2	98.68	98.76	98.81	99.47	97.67	99.47
S3	98.25	96.41	88.57	100.00	98.19	100.00
S4	99.11	99.26	94.57	100.00	100.00	99.79
S5	98.43	100.00	99.23	100.00	100.00	100.00
S6	99.00	99.58	93.87	99.73	98.61	99.68
S7	100.00	100.00	100.00	95.92	100.00	100.00
S8	100.00	96.24	95.93	97.89	99.23	99.05
S9	100.00	99.71	93.67	100.00	91.59	100.00
S10	99.78	99.67	94.89	100.00	96.79	96.22
S11	99.29	100.00	100.00	100.00	98.62	99.23
Mean	99.19	99.02	96.21	99.29	97.97	99.36

employing sEMG data taken from healthy as well as abnormal subjects. The average classification accuracy of walking, sitting, and standing is 99.19%, 99.02%, and 96.21% for healthy subjects, and 99.29%, 97.97%, and 99.36% for knee abnormal subjects, respectively.

TABLE IV
ACCURACY OF THE STUDIED MACHINE LEARNING MODELS

Subjects	Normal				Abnormal			
	GNN	MLP	KNN	SVC	GNN	MLP	KNN	SVC
S1	98.93	95.02	91.42	39.30	98.71	89.57	95.27	63.68
S2	98.56	90.56	91.94	39.50	98.91	89.82	96.66	32.41
S3	96.15	92.29	86.35	52.76	99.62	95.02	99.61	77.89
S4	98.28	93.71	91.08	39.04	99.90	91.91	98.03	73.49
S5	99.14	94.37	97.07	40.31	100.00	98.25	98.81	66.13
S6	98.37	91.24	89.65	46.14	99.36	83.02	90.56	87.72
S7	100.00	96.18	97.88	57.60	98.46	81.06	88.37	36.45
S8	97.76	88.92	94.32	53.86	98.79	95.41	87.98	33.00
S9	99.07	92.10	92.24	64.09	97.25	82.41	89.26	44.56
S10	99.48	95.25	93.66	61.13	97.76	82.30	95.64	37.68
S11	99.62	96.21	93.83	49.94	99.30	90.52	91.10	42.83
Mean	98.67	93.26	92.68	49.43	98.92	89.03	93.75	54.17

Best values found are in bold.

TABLE V
COMPUTATIONAL TIME (IN SEC)

Subjects	Normal		Abnormal	
	Training time	Testing time	Training time	Testing time
S1	913.50	0.23	2525.75	0.80
S2	883.90	0.22	2057.50	0.47
S3	1932.80	0.34	18369.35	1.72
S4	1013.15	0.21	38146.95	2.61
S5	1101.85	0.25	21544.35	1.80
S6	1610.80	0.28	14035.96	1.29
S7	1720.25	0.30	1538.90	0.27
S8	2201.10	0.39	1625.45	0.31
S9	1612.90	0.28	6361.36	0.65
S10	5447.05	1.16	5461.55	0.86
S11	628.70	0.16	5677.80	0.68
Mean	1733.27	0.35	10667.72	1.04

In this study, the accuracy of the most common machine learning algorithms is calculated, such as multilayer perceptron (MLP), k-nearest neighbor (KNN), and support vector machine, and compared with our proposed work, shown in Table IV. From the presented results, it may be observed that the classification accuracy of the proposed work is better than other machine learning classifiers in each subject and has attained classification accuracy in its average form. Its value is 98.67% and 98.92% for healthy subjects and knee abnormal subjects, respectively.

D. Computational Time

Table V shows the time required for training and testing the GNN model for healthy and knee-abnormal subjects. The length of each subject's sEMG signal is different, so the time needed to test and train the GNN models is also different. The average training time is 1733.27 s for healthy people and 10667.72 s for people with knee problems. The mean values of testing time for healthy people and people with knee problems are 0.35 and 1.04 s, respectively.

TABLE VI
COMPARISON OF THE PROPOSED METHODOLOGY'S PERFORMANCE WITH LITERATURE STUDIES WITH SIMILAR DATASETS

Approach	Subject	Walking	Sitting	Standing
MP-ANN [30]	Knee abnormal	88	94	92
Transfer learning-based LRCN [19]	Healthy	98.2	97.7	98.4
	Knee abnormal	92.8	92.3	92.2
ICA-EBM [17]	Healthy	96.0	96.2	96.2
	Knee abnormal	86.6	86.4	85.5
WD-EEMD [15]	Healthy	85.11	88.70	93.50
	Knee abnormal	98.86	96.38	96.77
Proposed method	Healthy	99.19	99.02	96.21
	Knee abnormal	99.29	97.97	99.36

E. Comparison With Existing Methods

Many approaches for activity recognition through sEMG signals for the lower limb have been proposed in the literature. Herrera-Gonzalez et al. [30] used the MP-ANN approach for the creation of the classifier with an 88% accuracy for walking activity, a 94% accuracy for sitting activity, and a 92% accuracy for standing activity. Naik et al. [17] provided 96.14% and 86.17% accuracy in healthy individuals and knee abnormality individuals, respectively, for the classification of lower limb movements. In other research, Gautam and others [19] developed the LRCN model based on transfer learning for the recognition of walking as well as sitting and standing activities. A hybrid technique named WD-EEMD to accomplish preprocessing tasks is performed by Vijayvargiya et al. [15] for the classification of the same problem, attaining an accuracy level of 98.2%, 97.7%, and 98.4% for healthy people and 92.8%, 92.3%, and 92.2% for individuals with knee abnormalities, respectively. Vijayvargiya et al. used a hybrid technique named WD-EEMD to accomplish a preprocessing task for the classification of the same problem, with an accuracy of 85.11%, 88.70%, and 93.50% for healthy people, and 98.86%, 96.38%, and 96.77% for individuals with knee anomalies, respectively. This article considers the identical sEMG data for lower limb muscles as the other contributors. Table VI enables comparing the suggested model's performance to previous studies with the same dataset. It concluded that the proposed technique performed well for recognition of the movements of the lower limb in both abnormal and healthy subjects. The previous studies considered a similar dataset as our study.

V. CONCLUSION

The study demonstrates the automatic recognition of lower limb movements utilizing sEMG signals for a patient with a knee deformity as well as a healthy subject using GNNs. Previously conducted research on the accuracy of sEMG utilizing three distinct lower limb movements (walking, standing, and sitting) revealed that it was poor. As a result, the suggested work addresses the improvement in the accuracy of the classification task through the application of a graph-based method. WD-EEMD preprocessing was used to first denoise the sEMG signal, and then an overlapping windowing method was used to extract

nine TD handmade features from each muscle signal. Then, a Pearson correlation-based graph was formed, and a GNN was used to recognize different gait activities by evaluating their performance parameters. In comparison to other classifiers, the GNN performed the best in all healthy and knee abnormal participants. The suggested technique attained the classification accuracy level in terms of its average form as 98.67% for healthy subjects and 98.92% for subjects suffering from knee abnormality.

Extensions to the suggested work are still possible in the future. To begin with, the dataset utilized contains data from a somewhat small sample size and no transition effect is considered. As a further study, the proposed methodology is evaluated using a significant number of subjects that would be able to mitigate the bias introduced by the short dataset and also analyze the transition effects of activities. Second, since the suggested methodology got validated by the use of an offline dataset, future research is capable of focusing on its clinical validation by the use of a real-time dataset. Here, only TD features are considered. The researchers can use unsupervised features or FD features in their further studies.

COMPLIANCE WITH ETHICAL STANDARDS

Ethical approval: This article does not contain any studies with human participants performed by any of the authors.

REFERENCES

- [1] S. Ranasinghe, F. Al Machot, and H. C. Mayr, "A review on applications of activity recognition systems with regard to performance and evaluation," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 8, pp. 1–22, Aug. 2016.
- [2] R. Gopura, K. Kiguchi, G. Mann, and D. Torricelli, "Robotic prosthetic limbs," *J. Robot.*, vol. 2018, no. 8, pp. 1–2, 2018.
- [3] A. Vijayvargiya, C. Prakash, R. Kumar, S. Bansal, and J. M. R. Tavares, "Human knee abnormality detection from imbalanced sEMG data," *Biomed. Signal Process. Control*, vol. 66, Apr. 2021, Art. no. 102406.
- [4] D. Hendry, K. Chai, A. Campbell, L. Hopper, P. O'Sullivan, and L. Straker, "Development of a human activity recognition system for ballet tasks," *Sports Med.-Open*, vol. 6, no. 1, pp. 1–10, 2020.
- [5] M. A. R. Ahad, A. D. Antar, and M. Ahmed, "Sensor-based human activity recognition: Challenges ahead," in *IoT Sensor-Based Activity Recognition*. Berlin, Germany: Springer, 2021, pp. 175–189.
- [6] A. Vijayvargiya, P. Singh, R. Kumar, and N. Dey, "Hardware implementation for lower limb surface EMG measurement and analysis using explainable AI for activity recognition," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–9, 2022.
- [7] N. Nazmi, M. A. Abdul Rahman, S.-I. Yamamoto, S. A. Ahmad, H. Zamzuri, and S. A. Mazlan, "A review of classification techniques of EMG signals during isotonic and isometric contractions," *Sensors*, vol. 16, no. 8, 2016, Art. no. 1304.
- [8] D. Farina and F. Negro, "Accessing the neural drive to muscle and translation to neurorehabilitation technologies," *IEEE Rev. Biomed. Eng.*, vol. 5, pp. 3–14, 2012.
- [9] G. Biagiotti, P. Crippa, L. Falaschetti, S. Orcioni, and C. Turchetti, "Human activity monitoring system based on wearable sEMG and accelerometer wireless sensor nodes," *Biomed. Eng. online*, vol. 17, no. 1, pp. 1–18, 2018.
- [10] A. Vijayvargiya, B. Singh, R. Kumar, and J. M. R. Tavares, "Human lower limb activity recognition techniques, databases, challenges and its applications using sEMG signal: An overview," *Biomed. Eng. Lett.*, vol. 12, no. 4, pp. 343–358, 2022.
- [11] M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, "Techniques of EMG signal analysis: Detection, processing, classification and applications," *Biol. Procedures Online*, vol. 8, no. 1, pp. 11–35, 2006.
- [12] R. H. Chowdhury, M. B. Reaz, M. A. B. M. Ali, A. A. Bakar, K. Chellappan, and T. G. Chang, "Surface electromyography signal processing and classification techniques," *Sensors*, vol. 13, no. 9, pp. 12431–12466, 2013.
- [13] C.-F. Jiang and S.-L. Kuo, "A comparative study of wavelet denoising of surface electromyographic signals," in *Proc. IEEE 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2007, pp. 1868–1871.
- [14] A. O. Andrade, S. Nasuto, P. Kyberd, C. M. Sweeney-Reed, and F. Van Kanijn, "EMG signal filtering based on empirical mode decomposition," *Biomed. Signal Process. Control*, vol. 1, no. 1, pp. 44–55, 2006.
- [15] A. Vijayvargiya, V. Gupta, R. Kumar, N. Dey, and J. M. R. Tavares, "A hybrid WD-EEMD sEMG feature extraction technique for lower limb activity recognition," *IEEE Sensors J.*, vol. 21, no. 18, pp. 20431–20439, Sep. 2021.
- [16] P. L. Singh, S. M. Verma, A. Vijayvargiya, and R. Kumar, "WD-EEMD based voting classifier for hand gestures classification using sEMG signals," in *Proc. IEEE 6th Int. Conf. Comput., Commun. Autom.*, 2021, pp. 225–230.
- [17] G. R. Naik et al., "An ICA-EBM-based sEMG classifier for recognizing lower limb movements in individuals with and without knee pathology," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 3, pp. 675–686, Mar. 2018.
- [18] A. Vijayvargiya et al., "Voting-based 1 D CNN model for human lower limb activity recognition using sEMG signal," *Phys. Eng. Sci. Med.*, vol. 44, no. 4, pp. 1297–1309, 2021.
- [19] A. Gautam, M. Panwar, D. Biswas, and A. Acharyya, "MyoNet: A transfer-learning-based LRCN for lower limb movement recognition and knee joint angle prediction for remote monitoring of rehabilitation progress from sEMG," *IEEE J. Transl. Eng. health Med.*, vol. 8, pp. 1–10, 2020.
- [20] W. Fan et al., "Graph neural networks for social recommendation," in *Proc. World Wide Web Conf.*, 2019, pp. 417–426.
- [21] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *Expert Syst. Appl.*, vol. 207, 2022, Art. no. 117921.
- [22] P. Hart and A. Knoll, "Graph neural networks and reinforcement learning for behavior generation in semantic environments," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 1589–1594.
- [23] R. Mondal, D. Mukherjee, P. K. Singh, V. Bhateja, and R. Sarkar, "A new framework for smartphone sensor-based human activity recognition using graph neural network," *IEEE Sensors J.*, vol. 21, no. 10, pp. 11461–11468, May 2021.
- [24] O. Sanchez, J. Sotelo, M. Gonzales, and G. Hernandez, "EMG dataset in lower limb data set," *UCI Mach. Learn. Repository*, vol. 2, 2014.
- [25] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Wavelet-based denoising algorithm for robust EMG pattern recognition," *Fluctuation Noise Lett.*, vol. 10, no. 2, pp. 157–167, 2011.
- [26] A. Graps, "An introduction to wavelets," *IEEE Comput. Sci. Eng.*, vol. 2, no. 2, pp. 50–61, 1995.
- [27] C. He, J. Xing, J. Li, Q. Yang, and R. Wang, "A new wavelet threshold determination method considering interscale correlation in signal denoising," *Math. Problems Eng.*, pp. 1–10, 2015.
- [28] W. M. B. W. Daud, A. B. Yahya, C. S. Horng, M. F. Sulaima, and R. Sudirman, "Features extraction of electromyography signals in time domain on biceps brachii muscle," *Int. J. Model. Optim.*, vol. 3, no. 6, pp. 515–519, 2013.
- [29] K. Pearson, "Mathematical contributions to the theory of evolution.—on a form of spurious correlation which may arise when indices are used in the measurement of organs," *Proc. Roy. Soc. London*, vol. 60, no. 359–367, pp. 489–498, 1897.
- [30] M. Herrera-González, G. A. Martínez-Hernández, J. L. Rodríguez-Sotelo, and Ó. F. Avilés-Sánchez, "Knee functional state classification using surface electromyographic and goniometric signals by means of artificial neural networks," *Ingeniería y Universidad*, vol. 19, no. 1, pp. 51–66, 2015.



Ankit Vijayvargiya (Senior Member, IEEE) received the B.Tech. (Hons.) degree in electrical engineering from Rajasthan Technical University, Kota, India, in 2010, the M.Tech. degree in instrumentation from Devi Ahilya University, Indore, India, in 2014, and the Ph.D. degree in electrical engineering from the Malaviya National Institute of Technology, Jaipur, India.

He is an Associate Professor with the Department of Electrical Engineering, Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur, India. His research interests include biomedical signal processing, pattern recognition, and machine learning.



Rajesh Kumar (Senior Member, IEEE) received the B.Tech. (Hons.) degree in electrical engineering from the National Institute of Technology, Kurukshetra, India, in 1994, the M.E. (Hons.) degree in power systems from the Malaviya National Institute of Technology (MNIT) Jaipur, Jaipur, India, in 1997, and the Ph.D. degree in intelligent systems from the University of Rajasthan, Jaipur, in 2005.

Since 1995, he has been a Faculty Member of the Department of Electrical Engineering, MNIT Jaipur, where he is currently a Professor. From 2009 to 2011, he was a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His research interests include the theory and practice of intelligent systems, computational intelligence and analytics, electric transportation, smart grid, and robotics.



Parul Sharma received the B.E. degree in electrical and electronics engineering from Mugneeram Bangur Memorial Engineering College, Jodhpur, India, in 2018, and the M.Tech. degree in power electronics and drives engineering from the Malaviya National Institute of Technology, Jaipur, India.

She is currently a Product Design Engineer with Mercedes-Benz. Her research interests include power electronics and machine learning applications.



sEMG-based deep learning framework for the automatic detection of knee abnormality

Ankit Vijayvargiya^{1,2} · Bharat Singh¹ · Nidhi Kumari² · Rajesh Kumar¹

Received: 2 November 2021 / Revised: 5 June 2022 / Accepted: 2 July 2022 / Published online: 24 July 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

Knee abnormality is a vital issue that can be diagnosed utilizing a sEMG signal to detect muscle abnormalities. Manually analyzing EMG data is time-consuming and requires skilled doctors. Hence, this paper aims to provide an automated system for the diagnosis of knee abnormality. Here, sEMG signal acquired from four different lower limbs muscles of 22 volunteers with three activities: walking, sitting, and standing, of which 11 seem healthy, and the rest were diagnosed clinically with knee abnormality. Noises are present during the sEMG signal recording, so a multi-step classification approach is proposed here. At first, wavelet denoising was implemented to denoise the sEMG signals. Further, the overlapping windowing method with a window size of 256 ms along with an overlapping of 25% was utilized to minimize the computational complexity. Afterward, a hybrid convolutional neural network with long short-term memory (Conv-LSTM) model is used for screening abnormal subjects. In this hybrid approach, a convolutional neural network (CNN) is used for temporal learning, while long short-term memory (LSTM) is for sequence learning. The results exhibit that the proposed wavelet-based denoising followed by Conv-LSTM model is the most precise and convenient model used for the detection of knee abnormality using sEMG signals so far.

Keywords Knee abnormality detection · Wavelet denoising · Convolutional neural network · Long short-term memory · Deep learning · Surface EMG signal

1 Introduction

Knee pain is an ordinary illness that influences individuals of all age groups. The main cause of knee pain is aging, injuries like torn cartilage, ruptured ligaments, and knee osteoarthritis. Technology, such as tracking recovery progress, has a substantial potential for boosting the quality of lifestyle of such physically challenged people. As claimed by a study in

[1], out of four individuals, one is a patient of joint symptoms or arthritis because of injury like knee osteoarthritis. The knee joint is the only joint that imparts motion of the leg within the body of a human being. In addition to acting as a shock absorber, it also stabilizes and provides balance to an individual's body. It constitutes distinct parts such as ligaments, cartilage, muscles, fluids, bones, and tendons. Formation of the knee joint that takes place via articulation among the femur, patella, and tibia. Thus, any abnormality or external harm to any parts of the knee joint can produce knee pain or abnormality in the knee [2].

Further, various medical imaging modalities such as magnetic resonance imaging (MRI), X-Ray, medical radiation, angiography, and computed tomography (CT) scanners can be used to detect abnormalities in the knee. MRI is an imaging approach that is utilized in radioscopy to configure images of the anatomy together with the physiological procedures of the body. They utilize intense magnetic fields, radio waves, magnetic field gradients along with a computer to create pictures of the body's organs. It is implemented to analyze pain in the knee; however, it is quite expensive. X-ray radiation

✉ Ankit Vijayvargiya
ankitvijayvargiya29@gmail.com

Bharat Singh
bharatsingh1993@yahoo.com

Nidhi Kumari
nidhikumari0998@gmail.com

Rajesh Kumar
rkumar.ee@mnit.ac.in

¹ Department of Electrical Engineering, Malaviya National Institute of Technology, Jaipur, India

² Department of Electrical Engineering, Swami Keshvanand Institute of Technology, Jaipur, India

generates images of body parts. Various problems that affect soft tissue, including internal organs, breaks, and fractures in the bone can be diagnosed with the help of an X-ray [3]. It is also utilized in the examination of knee pain. Computer tomography involves a computerized imaging process inside which confined X-ray radiation is directed toward the patient, that tends to generate signals which are treated with the help of a computer to create partial pictures of the body. Abnormalities of the knee can be traced by using wearable sensors such as gyrometer, EMG, accelerometer, etc. along with the visual sensors [4]. Electromyography is a demonstrative process that estimates the condition of nerve cells and muscles. EMG can also be utilized to recognize the activity of an individual by forecasting the actions in advance, and it quickly recognizes variations in the signal. There are two approaches to accumulate electromyography signals that is intramuscular EMG (iEMG) and surface EMG (sEMG) [5]. sEMG has several benefits compared to iEMG, as the probability of inflammation becomes nil; also electrodes could be implemented without any medical observation. The sEMG signal collected with the help of sEMG sensors for everyday human activities can be used in distinct applications like health monitoring, detection of neuromuscular disorders, automated command of upper and lower limb exoskeletons, etc [6,7].

Recently, machine learning approaches have been widely used in the medical field to detect the various types of disease with the help of biomedical data [7]. Research over spinal cord injury by utilizing the sEMG signal that was collected while the activities of the upper limb were being executed is done by Silva et al. [9]. Chen et al. examined the human lower limb extension angle utilizing a deep neural network in [10]. For motorized lower prosthetic limbs, Varol et al. [11] established a multi-class real-time intended classification technique relying upon sEMG signals. Choi et al. [12] determined the pattern of sEMG signals based on neural networks. A comparative analysis of machine learning models for the detection of knee abnormality is discussed in [13]. Diagnosis of knee abnormality with imbalanced sEMG data for gait activity was examined by Vijayvargiya et al. [14]. They have shown the effect of an imbalanced signal during the diagnosis of a knee abnormality and have used various oversampling techniques to improve the performance of machine learning models. Ertuğrul et al. have developed an adaptive local binary pattern (ALBP) technique to extract features and classify the healthy and knee abnormal subjects with an accuracy of 85% [15].

Machine learning models are required to handcrafted features from the signals where features have to be selected by some machine learning or statistical methods. It is a very tedious task to choose the appropriate feature set manually. As per the study, deep learning models such as CNN, LSTM, GRU, etc. are used to overcome this problem. In the deep

learning approaches, the features are first extracted by the algorithm, and then the classification process is performed. These deep learning models have been implemented to different applications and showed very high performance as per the previous studies [16]. CNN model is utilized for the processing of images, it also exhibits promising outcomes in acquiring features through stationary datasets. The architecture of CNN constitutes an input layer, an output layer, it also consists of several pooling layers, dense layers, convolutional layers, rectified linear unit layers in addition to dropout layers. It recognizes the main features without the need for supervision. The inability to evaluate the earlier or temporal data within time-series signals is one of the main problems with CNN. Though, to process and acquire temporal information, LSTM can be used.

According to an analysis of relevant literature, there are relatively few studies on knee abnormality identification using sEMG signals. The majority of research employs a handcrafted method, and there is currently no study that uses a deep learning strategy to identify knee problems using sEMG data. As a result, in this research, a combination of CNN and LSTM is presented to identify the knee abnormalities using the sEMG signal. The CNN-LSTM model comprises layers of CNN along with the LSTM network. A convolutional neural network is used for temporal learning whereas LSTM is used for sequence learning in the proposed CNN-LSTM model. Therefore, the CNN-LSTM [17] hybrid is an absolute model which permits learning on features for EMG signals together with long-term dependencies.

The major contribution of this research is:

1. This study comprises use of the sEMG signals that are acquired through leg muscles of an individual to recognize the knee abnormality by utilizing a hybrid Conv-LSTM-based deep learning framework.
2. In order to eliminate the artifacts of raw sEMG signals obtained via muscles of lower limb, wavelet denoising is employed as a section of preprocessing.
3. Performance parameters of individual models are also evaluated and compared with the proposed hybrid Conv-LSTM model. The proposed model is a high-performance model for automatic recognition of knee abnormality.

This paper includes the following section: In Sect. 2, an overview of the dataset is given. Section 3 comprises the proposed methodology. Section 4 consists of results along with discussions. Section 5 involves the conclusion and future scope.

2 Dataset

In this paper, authors have used the publically available sEMG signal dataset at UCI machine learning repository via Lichman et al. [18]. The dataset involves twenty-two candidates whose age was above 18 years. Amid those candidates, eleven looks fit and knee abnormality was exhibited by rest. The dataset constitutes the sEMG signal of the lower limb of the participants. No preceding medical record was discovered with reference to injury or pain in the knee of candidates who were fit. Out of knee abnormal candidates, six were affected with anterior cruciate ligament (ACL), one was encountered with sciatic nerve injury and four suffered from a meniscus injury. While the participants were engaged in different tasks such as flexure of leg up, leg extension from sitting position, and walking, data were assembled by utilizing the DataLOG (MWX8) via biometrics Ltd along with a goniometer. By using a goniometer that is connected to the joint of the knee on the external side, the dataset of four muscles including vastus medialis(VM), semitendinosus(ST), biceps femoris (BF) along with rectus femoris (RF) was collected. In this research work, only the sEMG signal collected from the three different tasks: walking, sitting, and standing were used. The sEMG signal was obtained from the infected limb of the knee abnormal participants and the left limb of the healthy participant. Dataset was collected at a sampling rate of about 1000Hz along with 14-bit resolution. Filtration of signals was done with the help of a bandpass filter accompanied by passband frequency from 20 Hz to 460Hz.

3 Proposed methodology

This section illustrates the proposed methodology as shown in Fig. 1, for the detection of abnormalities in the knee through the sEMG signal. The dataset utilized in the following research is used via the UCI machine learning repository which includes sEMG signals from lower limbs of about 22 volunteers among which 11 are fit and the rest 11 are diagnosed with abnormalities in the knee. Initially, the sEMG signal is pre-processed, which involves the elimination of noises by using a discrete wavelet denoising approach. Further, the segmentation of signal is done by the overlapping windowing technique. Afterward, the deep learning models including CNN, LSTM, and proposed convolutional LSTM were used to analyze the data for knee abnormality detection.

The following subsection includes the major part of the proposed methodology:

3.1 Wavelet denoising

Generally, four kinds of noises are commonly present while collecting the sEMG signal [19] such as (1) inherent noise

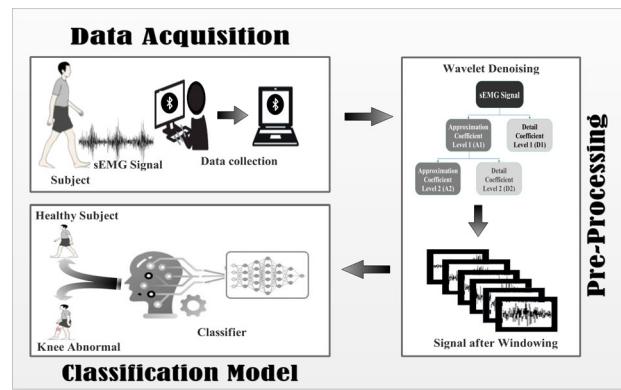


Fig. 1 Flow diagram of proposed methodology

generated via electronic appliances, (2) ambient noise resulting through electromagnetic devices, (3) inherent noise uncertainty because of the firing rate of motor units, and (4) motion artifacts generated through the sEMG electrodes movement. On that account, denoising of a signal is necessary, it also carried out before the signal is utilized for purpose of classification. To reduce the noise that is out of range for the sEMG signal spectrum band, a conventional approach of filtering including low pass, high pass together with bandpass may be utilized. sEMG signal proceeds through the bandpass filter (20 to 460 Hz). In recent research, a lot of different approach such as empirical mode decomposition (EMD), wavelet denoising, independent component analysis (ICA) have been utilized to minimize the noises in the sEMG signal [20]. Implementation of wavelet denoising on the sEMG signal that is obtained through lower and upper limbs has proven to be effective. The scheme to denoise the sEMG signal with the help of a wavelet denoising algorithm is demonstrated by Phinyomark et al. [21]. Among sEMG signals, several irregular noises such as white Gaussian noise are inconvenient to reduce via signal filtering. Hence, wavelet denoising is used to eliminate it. The expression for noisy signal $X(n)$ is stated as Eq. (1):

$$X(n) = Y(n) + K(n) \quad (1)$$

wherein $Y(n)$ demonstrates original signal, $K(n)$ is white Gaussian noise, and $X(n)$ presents noisy signal.

The wavelet denoising based upon DWT comprises the following steps:

- Signal decomposition is done by the DWT;
- Select the threshold for each wavelet transform;
- To regenerate the denoised signal, the inverse wavelet transforms along with the threshold function is used.

As a result of the implementation of wavelet denoising over signals, discrete coefficients of wavelets are gener-

ated when the signal is passed via low-pass as well as high pass filters. With the help of the wavelet denoising approach, detail and approximation coefficients are acquired after signal decomposition and thresholding is accomplished. The total number of coefficients is decided by the level of decomposition. Two conventional approaches are present for thresholding wavelet coefficients: soft and hard thresholding [22]. In soft thresholding, there is a continuous deviation respective to the original signal whereas hard thresholding consists of incomplete information about the original signal. This study constitutes universal thresholding [23] that is implemented over detailed coefficients. The form of universal thresholding selection is represented as Eq. (2):

$$\lambda = \sigma \sqrt{2 \ln(M)} \quad (2)$$

Herein, $\sigma = (MAD)/0.6745$ MAD is designated as median absolute deviation, M is the length of the signal.

This research consists of a wavelet denoising approach which is utilized along with sym4 originating from the family of symlet to the first level of decomposition [14]. In this previous study, calculated the value of mean squared error, mean absolute error, signal-to-noise ratio, and peak signal-to-noise ratio of the sEMG signal with different mother wavelets and level of decomposition on same dataset. They have proposed that sym4 originating from the family of symlet to the first level of decomposition is giving the best results compared to other mother wavelets and level of decomposition.

3.2 Segmentation

The segmentation is used as a part of pre-processing as it is a smart way to process time-series data used in a deep learning-based model to reduce the computational complexity. To implement the segmentation process, the windowing method is utilized. Two discrete methods of windowing are present to accomplish the process of segmentation: adjacent windowing and overlapping windowing [24]. The classification accuracy and classification response time will depend upon the window size of a signal. The accuracy of model increases with larger window size, but the response time is also increased. As per the previous studies, a delay of 150–250 ms interval is considered. So, we considered the overlapping windowing technique along with a window size of 256ms and an overlapping of 25% [25].

3.3 Deep learning frameworks

The model of CNN proceeds across the dataset through discrete hidden layers; however, the outcome is not supplemented back toward the network over here. Hence, models of CNN slightly lack in pursuing sequential information, though it is fine in acquiring temporal features. To eliminate

this issue, a hybrid of the CNN and LSTM model (CNN-LSTM) is introduced in this paper. LSTM can be utilized for sequential data learning due to its capability of learning through training and it can also recollect whatever it has grasped to predict the adjacent element. It retains long-term dependencies and along with these features is also processed sequentially by LSTM.

3.3.1 Convolutional neural network

CNN is majorly used to acquire necessary features through sEMG signals to train the model. It generally comprises a composition of convolution operation and neural network. The convolution operation is practiced via streaming the particular kernel on the inserted data to obtain the feature map. A CNN model is majorly constructed using multi-layers, where high-level layers possess a significant quantity of kernels whereas lower-level layers constitute less quantity of kernels. In comparison with various computational classifiers, CNN possess lessened pre-processed data and feature extracted from the dataset. In contrast, CNN is basically utilized to employ 2D data including images and videos. Because of this, CNN is usually referred as 2D CNN. In recent times, 1D CNN are discovered as a moderation of 2D CNNs for time series data [26].

In this research work, feature extraction using CNN block considered three convolutional layers with 1 max-pooling layer. After that one fully connected layer is connected with the output layer. Here, signals are collected from the four channels and the window size used is 256, so the input sequence of 1D CNN is 1024, and batch size is considered 32. The ReLU activation function is used in a fully connected layer while the softmax activation function has been used in the output layer. The softmax function is employed as a classifier that predicts the input signal's class.

The input given to k th neuron of i th layer is determined by Eq. (3):

$$x_k^l = \sum_{i=1}^{N_{l-1}} \text{Conv1D}(W_{ik}^{l-1}, o_i^{l-1}) + b_k^l \quad (3)$$

Here x_k^l denotes the input provided to k th neuron of i th layer; b_k^l represents bias of k th neuron of i th layer; N_{l-1} designates the quantity of neurons within $(l-1)$ th layer and W_{ik}^l indicates the kernel through i th neuron of layer $l-1$ to the k th neuron of layer l . The output after implementing activation function to x_k^l is stated as Eq. (4):

$$y_k^l = f(x_k^l) \quad (4)$$

Here $f(\cdot)$ symbolizes activation function and we have used ReLU function which is demonstrated as Eq. (5):

$$y_k^l = \max(0, x_k^l) \quad (5)$$

Basically, its purpose is to restore the entire negative values with zero. After implementing the operation of max pooling to y output is given as Eq. (6):

$$o_k^l = \text{for each pooling window } \max(y_k^l[0], y_k^l[1], \dots, y_k^l[\text{pool size}]) \quad (6)$$

Loss function which is utilized for back-propagation is given as Eq. (7):

$$L = - \sum_{i=1}^C t_i \log(f(s)_i) \quad (7)$$

where the total number of classes in the dataset is denoted by C , t_i indicates the target output for input instance, and $f(s)_i$ is the probability determined by softmax function, which measures the likelihood of an input instance belonging to the i th class.

Parameter updation in back-propagation is given as Eqs. (8) and (9):

$$w_{ik}^{l-1}(t+1) = w_{ik}^{l-1}(t) - \alpha * \frac{\delta L}{\delta w_{ik}^{l-1}} \quad (8)$$

$$b_k^l(t+1) = b_k^l(t) - \alpha * \frac{\delta L}{\delta b_k^l} \quad (9)$$

where α is a learning rate hyperparameter.

3.3.2 Long short-term memory

In 1997, Schmid Huber together with Hochreiter proposed the architecture of LSTM for sequence learning [27]. To represent the long-term as well as short-term memory, sequence learning is mandatory. These are designed to attenuate the issue of vanishing and exploding gradient. The ordinary LSTM network constitutes significant hidden units termed as memory cells that reminisce the preceding input for a long duration.

Figure 2 illustrates the architecture of the LSTM cell which comprises forget gate, input gate, and output gate. A forget gate(f_t) eliminates the information from the cell state. This is essential to optimize the LSTM network's performance. Two inputs that is h_{t-1} together with x_t is fed into this gate. h_{t-1} denotes the hidden state of the preceding cell whereas x_t is the input at a specific time step. The specified inputs are then multiplied by their respective weights and

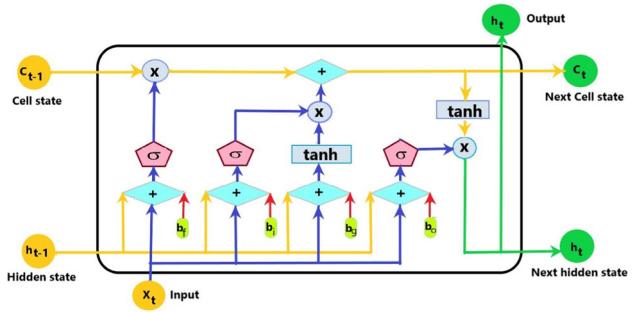


Fig. 2 Architecture of LSTM

a bias is added. Further, the sigmoid activation function is implemented over this value. The sigmoid function decides which value to be kept and which to be discarded. The equation of forget gate is demonstrated as Eq. (10):

$$f_t = \sigma[(W_{fh} * h_{t-1}) + (W_{fx} * x_t) + b_f] \quad (10)$$

where: W_{fh} and W_{fx} are the weights assigned to activation state and input state, respectively, b_f denotes the bias.

Similarly, the equation for input gate can be calculated as Eq. (11):

$$i_t = \sigma[(W_{ih} * h_{t-1}) + (W_{ix} * x_t) + b_i] \quad (11)$$

An input gate (i_t) adds information to the cell state. It consists of an input gate along with an input node. The input node is represented as Eq. (12):

$$g_t = \tanh[(W_{gh} * h_{t-1}) + (W_{gx} * x_t) + b_g] \quad (12)$$

An output gate (o_t) ensures whether information regarding the current cell state is visible or not. The equation for the output gate is given as Eq. (13):

$$o_t = [(W_{oh} * h_{t-1}) + (W_{ox} * x_t) + b_o] \quad (13)$$

A correlation is determined by the LSTM model for hidden activation h_t through the LSTM cell, which intakes x_t as the input at the current time step, in addition to this, it also obtains the information h_{t-1} from the preceding step.

Each cell of LSTM consists of a cell state c_t which acts as a memory that allows hidden units to retain information from the past. It is represented as Eq. (14):

$$c_t = c_{t-1} * f_t + i_t * g_t \quad (14)$$

The model of LSTM generates hidden activation h_t which is utilized further in order to make predictions. It is depicted as Eq. (15):

$$h_t = \tanh(c_t) * o_t \quad (15)$$

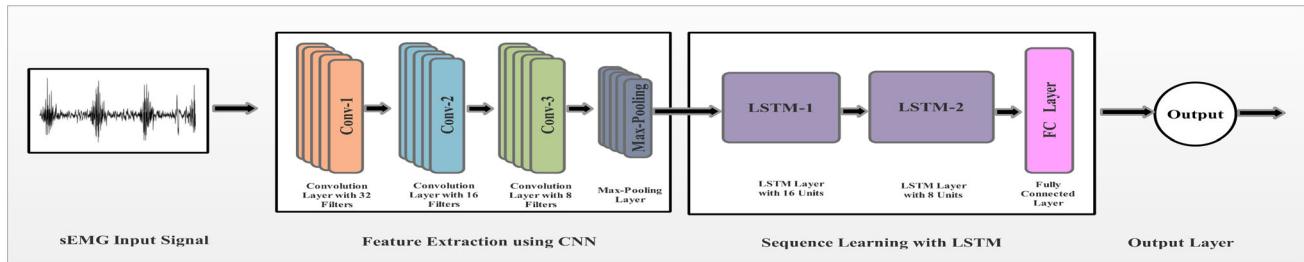


Fig. 3 Proposed hybrid Conv-LSTM network

Table 1 Parameters of studied deep learning models

Model parameters		CNN	LSTM	Conv-LSTM
Convolutional Layer1	Number of Filter	32	–	32
	Kernel Size	5	–	5
Convolutional Layer2	Number of Filter	16	–	16
	Kernel Size	3	–	3
Convolutional Layer3	Number of Filter	8	–	8
	Kernel Size	3	–	3
Pooling Layer	Type of Pooling	Max	–	Max
	Kernel Size	2	–	2
LSTM	Unit Size	–	16	16
	Return Sequence	–	True	True
LSTM	Unit Size	–	8	8
	Return Sequence	–	True	True
Fully Connected	Unit Size	16	16	16
Fully Connected	Activation	Relu	Relu	Relu
Fully Connected	Unit Size	2	2	2
Fully Connected	Activation	Softmax	Softmax	Softmax

3.3.3 Proposed hybrid Conv-LSTM model

The framework of the 8-layer hybrid CNN-LSTM model is demonstrated in Fig. 3. The initial three layers are the convolutional layer with the Relu activation function for extracting the features from the sEMG signals. Further, a max-pooling layer is used to reduce the number of parameters, so the complexity of the model is reduced. For sequence learning, LSTM layer 5 along with layer 6 is used. Fully-connected 7 layer is associated with 16 fully-connected neurons. The final layer that is layer 8 comprises 2 outputs neurons that distinguish among healthy and diagnosed (with knee abnormality) volunteers. Table 1 constitutes each layer used in the CNN-LSTM model and the parameter corresponding to each layer.

4 Results and discussion

4.1 Model parameters and performance indices

The proposed model needs to be trained on a training dataset that requires the weight parameters to be learned from the data. This research comprises the use of Adam optimizer [28] which involves the conventional back-propagation approach with cross-entropy loss function and stochastic gradient descent approach. The default values of entire six hyper-parameters including: learning rate (0.001), epsilon (0.00000001), beta1 (0.9), beta2 (0.999), and used locking (false) of adam's algorithms is utilized. Python using Keras libraries has been used to implement the model which is a freely available deep learning library from google. We trained our model with 50 iterations.

As the sEMG signal is a time-series signal, so here training and testing data is divided sequentially with starting 70% signal of each subject as a training dataset and the remaining 30% of the signal from each subject as a testing dataset. This problem is having two-class problems in which two different classes are: healthy and knee abnormal. The confusion matrix will be formed as Eq. (16):

$$C = \begin{bmatrix} C_{AA} & C_{AN} \\ C_{NA} & C_{NN} \end{bmatrix} \quad (16)$$

where C_{AA} is the number of cases in knee abnormal class as predicted knee abnormal, C_{NN} is the number of cases in healthy (normal) class as predicted healthy, C_{AN} is the number of cases in knee abnormal class as predicted healthy, C_{NA} is the number of cases in healthy class as predicted knee abnormal. Here, four performance matrices: accuracy, sensitivity, specificity, and F-score are considered for the evaluation of models shown in Table 2.

4.2 Performance evaluation

Table 3 comprises the performance of studied deep learning methods. These results indicate that the proposed hybrid Conv-LSTM method classifies the healthy and knee abnormal subjects more reasonably than other studied deep learn-

Table 2 Mathematical expression of performance indices

Performance measures	Mathematical expression
Accuracy	$\frac{C_{AA}+C_{NN}}{C_{AA}+C_{NN}+C_{AN}+C_{NA}}$
Sensitivity (Recall)	$\frac{C_{AA}}{C_{AA}+C_{AN}}$
Specificity	$\frac{C_{NN}}{C_{NN}+C_{NA}}$
Precision	$\frac{C_{AA}}{C_{AA}+C_{NA}}$
F-Score	$\frac{2C_{AA}}{2C_{AA}+C_{AN}+C_{NA}}$

ing models. The accuracy of the proposed Conv-LSTM was 98.61% while for the CNN and LSTM deep learning models, it was 96.03% and 98.31%, respectively. Similarly, the F-score value of Conv-LSTM was 98.92% while for the CNN and LSTM deep learning models, it was 96.87% and 98.68%, respectively. Therefore, the accuracy of the proposed hybrid Conv-LSTM model is higher than the CNN and LSTM model.

Figure 4 depicts the confusion matrix attained from the studied deep learning models: CNN, LSTM, and hybrid

Table 3 Performance parameters obtained from studied deep learning models in %

Models	Accuracy	Sensitivity	Specificity	F score
CNN	96.03	95.23	97.48	96.87
LSTM	98.31	98.08	98.74	98.68
Conv-LSTM	98.61	98.92	98.04	98.92

Bold values indicate the accuracy and F-score value of proposed Conv-LSTM model is better compared with CNN and LSTM models

Conv-LSTM. The confusion matrix is utilized to illustrate the performance of models which comprises information regarding the predicted and true labels estimated by a model. Figure 4c shows the confusion matrix of the Conv-LSTM model in which a total of 1286 (98.92%) samples are correctly predicted as a knee abnormal and 701 (98.04%) samples are correctly predicted as healthy while 14 (1.96%) and 14 (1.07%) samples are incorrectly predicted as knee abnormal and healthy, respectively. Similarly, Fig. 4a and b are the confusion matrix for the CNN and LSTM models.

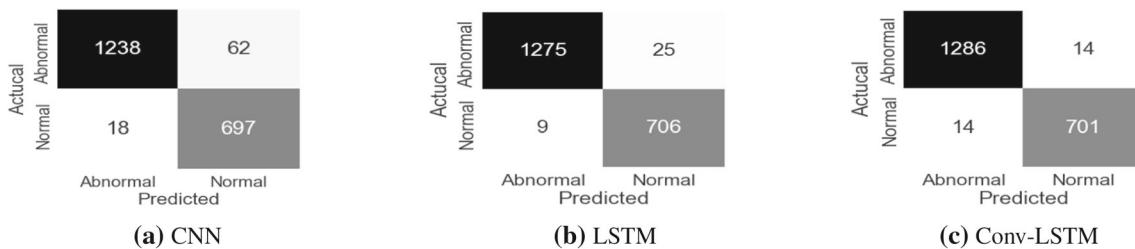
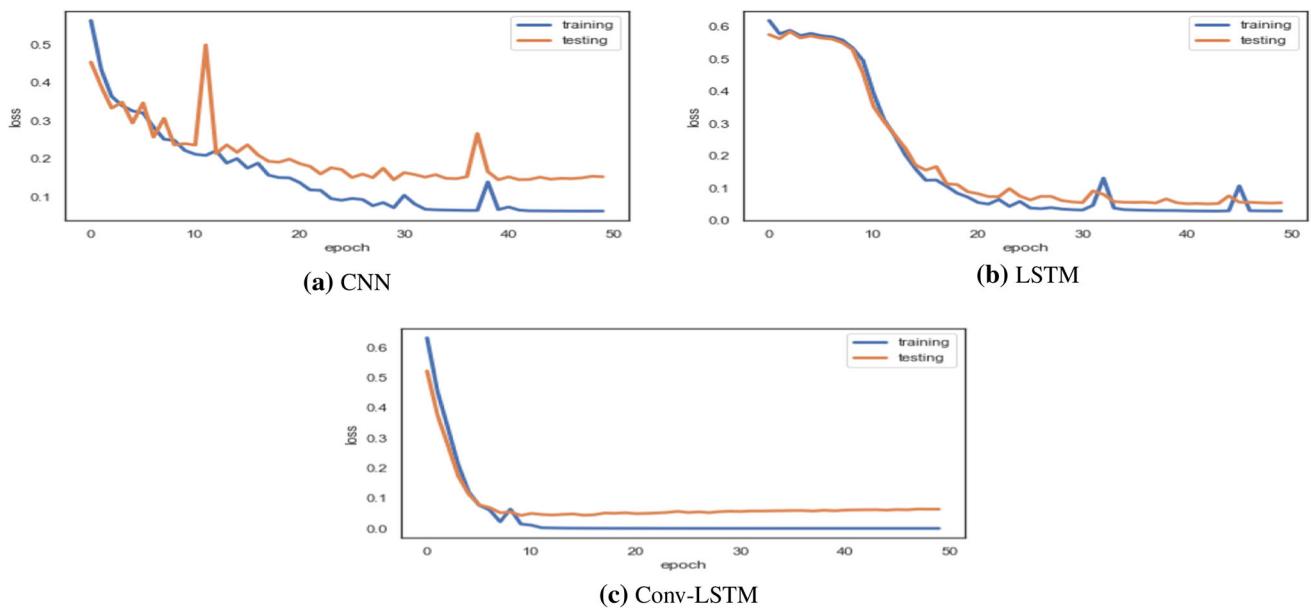
**Fig. 4** Confusion matrix**Fig. 5** Loss versus epoch plot

Table 4 Time taken per epoch

	CNN	LSTM	Conv-LSTM
Time	9 sec	240 Sec	16 Sec

4.3 Computational evaluation

The studied deep learning models are tested on a 30% testing dataset after training. Figure 5a, b and c shows the plot between loss vs epoch for CNN, LSTM and hybrid Conv-LSTM models with test and train dataset. The following plots emphasize that with the increase in the number of the epoch, the value of the loss function decreases and after some epochs, it reaches a steady state, which indicates that the issue of over-fitting has reduced. Figure 5 shows that the proposed hybrid Conv-LSTM model reaches a steady state at near to the 10th epoch and accuracy also reaches its maximum value that is far better than the other studied models.

Table 4 shows the time taken for completion of an epoch of studied deep learning models. It shows that if the CNN layer is used to extract the feature, then the time taken to complete an epoch is required less. Here, 16 sec is required in the proposed hybrid Conv-LSTM while for the CNN and LSTM deep learning models, it was of 9 sec and 240 sec. Therefore, if the models are compared based on their computational time then the significant difference was found between the Conv-LSTM and LSTM, while a very less significant difference between Conv-LSTM and CNN.

The proposed work is compared with previous similar work on automated detection of lower limb knee abnormality detection. Ertuğrul et al. have developed an adaptive local binary pattern (ALBP) technique to extract features and classify the healthy and knee abnormal subjects with an accuracy of 84.85% [15]. Vijayvargiya et al. have conducted a comparative analysis of five machine learning models for classification between abnormal and healthy subjects of the knee [13]. This study extracted eleven time-domain features after denoising the signal by wavelet denoising technique. The extra tree classifier has been shown the best performance (91.3% accuracy and 88.8% f-score) in comparison with other classifiers, which are the support vector machine, decision tree, K-nearest neighbor, and random forest classifier. For an appropriate comparison, the dataset considered in our study is the same as the other contributors whose results have been discussed earlier.

5 Conclusion and future scope

This study proposed wavelet-based denoising followed by Conv-LSTM model for the automated knee abnormality detection using sEMG signal of lower limb muscles with

three distinct activities. Initially, wavelet denoising is implemented to denoise the raw sEMG signals and then segmentation of sEMG signals is carried out by utilizing overlapping windowing technique. For screening of abnormal subjects, a hybrid Conv-LSTM model is proposed in which CNN is used for temporal learning and LSTM is used for sequence learning. Thus, the hybrid of Conv-LSTM shows the attributes of the best performer with an accuracy of 98.61% and computational time for an epoch of 16 sec. It appeared to be the most precise and convenient model that can be utilized for knee abnormality detection using sEMG signals.

To examine the proposed approach, the sEMG dataset utilized in this work consists of data collected from 22 subjects only. Thus, in the future, the approach could be justified by using a huge dataset obtained in real time. With the availability of more data, instead of just classifying it as abnormal or normal, the deep learning algorithm can be prepared to predict the level of knee abnormality in abnormal subjects. It could be incorporated in the network of the Internet of Medical Things (IoMT) for regular observation by a healthcare professional.

Acknowledgements This publication is supported by Visvesvaraya PhD Scheme, MeitY, Govt. of India, MEITY-PHD-2942.

References

- Arthritis Foundation. Arthritis By The Numbers. <https://www.arthritis.org/getmedia/e1256607-fa87-4593-aa8a-8db4f291072a/2019-abtn-final-march-2019.pdf> (2019)
- Richebé, P., Capdevila, X., Rivat, C.: Persistent postsurgical pain: pathophysiology and preventative pharmacologic considerations. *Anesthesiology* **129**(3), 590–607 (2018)
- Bedson, J., Jordan, K., Croft, P.: How do gpus use x rays to manage chronic knee pain in the elderly? a case study. *Ann. Rheum. Dis.* **62**(5), 450–454 (2003)
- Hussain, T., Maqbool, H.F., Iqbal, N., Salman, M.K., Dehghani-Sanij, A.A.: Computational model for the recognition of lower limb movement using wearable gyroscope sensor. *Int J Sensor Netw* **30**(1), 35–45 (2019)
- Merletti, R., De Luca, C.J.: New techniques in surface electromyography. *Comput. Aided Electromyogr. Expert Syst.* **9**(3), 115–124 (1989)
- Vijayvargiya, A., Singh, P.L., Verma, S.M., Kumar, R., Bansal, S.: Performance comparison analysis of different classifier for early detection of knee osteoarthritis. In: Sensors for Health Monitoring. Elsevier (2019)
- Vijayvargiya, A., Kumar, R., Dey, N., Manuel, J., Tavares, R.S.: Comparative analysis of machine learning techniques for the classification of knee abnormality. In: IEEE 5th International Conference on Computing Communication and Automation (ICCCA). IEEE (2020)
- Dhanka, B., Vijayvargiya, A., Kumar, R., Ghanshyam, S.: A comparative assessment of machine learning techniques for epilepsy detection using eeg signal. In: IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON). IEEE (2020)

9. Da Silva, U.S.L.G., Villagra, H.A., Oliva, L.L., Marconi, N.F.: Emg activity of upper limb on spinal cord injury individuals during whole-body vibration. *Physiol. Int. (Acta Physiologica Hungarica)* **103**(3), 361–367 (2016)
10. Chen, J., Zhang, X., Cheng, Y., Xi, N.: Surface emg based continuous estimation of human lower limb joint angles by using deep belief networks. *Biomed. Signal Process. Control* **40**, 335–342 (2018)
11. Varol, H.A., Sup, F., Goldfarb, M.: Multiclass real-time intent recognition of a powered lower limb prosthesis. *IEEE Trans. Biomed. Eng.* **57**(3), 542–551 (2009)
12. Choi, H.K., Jeong, J.H., Hwang, S.H., Choi, H.C. and Hak C.W.: Feature evaluation and pattern recognition of lower limb muscle emg during postural balance control. In: Key Engineering Materials, vol. 326, pp. 867–870. Trans Tech Publ (2006)
13. Vijayvargiya, A., Kumar, R., Dey, N., Tavares, J.M.R.S.: Comparative analysis of machine learning techniques for the classification of knee abnormality. In: IEEE 5th International Conference on Computing Communication and Automation (ICCCA). IEEE (2020)
14. Vijayvargiya, A., Prakash, C., Kumar, R., Bansal, S., Tavares, J.M.R.S.: Human knee abnormality detection from imbalanced sEMG data. *Biomed. Signal Process. Control* **66**, 102406 (2021)
15. Ertuğrul, Ö.F., Kaya, Y., Tekin, R.: A novel approach for semg signal classification with adaptive local binary patterns. *Med. Biol. Eng. Comput.* **54**(7), 1137–1146 (2016)
16. Acharya, U.R., Oh, S.L., Hagiwara, Y., Tan, J.H., Adeli, H.: Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Comput. Biol. Med.* **100**, 270–278 (2018)
17. Sanchez, O.F.A., Sotelo, J.L.R., Gonzales, M.H., Hernandez, G.A.M.: Emg dataset in lower limb data set. *UCI Mach. Learn. Repos.* 2 (2014)
18. Lichman, M. : UCI Machine Learning Repository, School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, Tech. Rep., 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>
19. Chowdhury, R.H., Reaz, M.B., Ali, M.A., Bakar, A.A., Chellappan, K., Chang, T.G.: Surface electromyography signal processing and classification techniques. *Sensors* **13**(9), 12431–12466 (2013)
20. Jiang, C.-F., Kuo, S.-L.: A comparative study of wavelet denoising of surface electromyographic signals. In: 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE (2007)
21. Phinyomark, A., Phukpattaranont, P., Limsakul, C.: Wavelet-based denoising algorithm for robust EMG pattern recognition. *Fluct. Noise Lett.* **10**(02), 157–167 (2011)
22. Graps, A.: An introduction to wavelets. *IEEE Comput. Sci. Eng.* **2**(2), 50–61 (1995)
23. He, C., Xing, J., Li, J., Yang, Q., Wang, R.: A new wavelet threshold determination method considering interscale correlation in signal denoising. *Math. Probl. Eng.* (2015)
24. Banos, O., Galvez, J.-M., Damas, M., Pomares, H., Rojas, I.: Window size impact in human activity recognition. *Sensors* **14**(4), 6474–6499 (2014)
25. Naik, G.R., Selvan, S.E., Arjunan, S.P., Acharyya, A., Kumar, D.K., Ramanujam, A., Nguyen, H.T.: An ICA-EBM-based sEMG classifier for recognizing lower limb movements in individuals with and without knee pathology. *IEEE Trans. Neural Syst. Rehabili Eng.* **26**(3), 675–686 (2018)
26. Kiranyaz, S., Ince, T., Gabbouj, M.: Real-time patient-specific ECG classification by 1-d convolutional neural networks. *IEEE Trans. Biomed. Eng.* **63**(3), 664–675 (2015)
27. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
28. Kingma, D.P., Adam, J.B.: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Voting-based 1D CNN model for human lower limb activity recognition using sEMG signal

Ankit Vijayvargiya^{1,2} · Khimraj³ · Rajesh Kumar¹ · Nilanjan Dey⁴

Received: 7 May 2021 / Accepted: 5 October 2021 / Published online: 8 November 2021
© Australasian College of Physical Scientists and Engineers in Medicine 2021

Abstract

Surface electromyography (sEMG) signal classification has many applications such as human-machine interaction, diagnosis of kinesiological studies, and neuromuscular diseases. However, these signals are complicated because of different artifacts added to the sEMG signal during recording. In this study, a multi-stage classification technique is proposed for the identification of distinct movements of the lower limbs using sEMG signals acquired from leg muscles of healthy knee and abnormal knee subjects. This investigation involves 11 subjects with a knee abnormality and 11 without knee abnormality for three distinct activities viz. walking, leg extension from sitting position (sitting), and flexion of the leg (standing). Discrete wavelet denoising to fourth level decomposition has been implemented for the artifact reduction and the signal has been segmented using overlapping windowing technique. A study of four different architectures of 1D convolutional neural network models is undertaken for the prediction of lower limb activities and the final prediction is achieved via a voting mechanism of all four model results. The performance parameters of CNN models have been calculated for three different cases: (1) healthy subjects (2) subjects with knee abnormality (3) Pooled data (combination of abnormal knee and healthy knee subjects) using nested threefold cross-validation. It has been found that the voting mechanism yields an average classification accuracy as 99.35%, 97.63%, and 97.14% for healthy subjects, knee abnormal subjects, and pooled data, respectively. The result validates that the proposed voting-based 1D CNN model is efficient and useful in lower limb activity recognition using the sEMG signal.

Keywords Surface electromyography · Lower limb activity recognition · Healthcare monitoring · Overlapping windowing · Discrete wavelet denoising · Convolutional neural network

Introduction

-
- ✉ Ankit Vijayvargiya
ankitvijayvargiya29@gmail.com
- Khimraj
khimrajsuthar@gmail.com
- Rajesh Kumar
rkumar.ee@mnit.ac.in
- Nilanjan Dey
neelanjan.dey@gmail.com

- ¹ Department of Electrical Engineering, Malaviya National Institute of Technology, Jaipur, India
- ² Department of Electrical Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur, India
- ³ Department of Computer Science and Engineering, Malaviya National Institute of Technology, Jaipur, India
- ⁴ Department of Computer Science, JIS University, Kolkata, India

Lower limb activity recognition (LLAR) plays a significant role in the determination of activities of a person/group for many different applications such as active and assistive living systems, monitoring and surveillance system, tele-immersion, and healthcare monitoring [1]. The data of human activities can be collected in two different ways, visual sensors or wearable sensors [2]. Wearable sensors are placed on the body of the subject for the collection of data whereas, visual sensors do not require to be placed on the subject's body. Visual sensors such as cameras are suitable for security applications but have certain limitations in terms of privacy, pervasiveness, and complexity for LLAR [3]. With the advancements in wearable sensor technology, various technologies are available for recording data, for example, accelerometers, EMG, gyroscopes, and barometers. EMG sensors are better as compared to the other wearable sensors since they can recognize the movement

in advance [4] within very less time [5]. The EMG signal is a biomedical signal that measures the electrical activity produced by the skeletal muscles. For recording the EMG signal, two techniques have been used: intramuscular EMG (invasive) and surface EMG (non-invasive) [6]. Intramuscular EMG (iEMG) signals are recorded by inserting a fine wire into the muscle while surface EMG (sEMG) signals are recorded above the surface of the muscle. The advantages of non-invasive techniques are: (1) Electrodes can be applied without any medical supervision and discomfort (2) No risk of infection [7].

sEMG signals can be used for a variety of healthcare applications viz. upper and lower limb prosthesis or exoskeleton control [8], neuromuscular disorders examination [9], monitoring of fitness and exercise [10]. The exoskeleton is a wearable device used to enhance the physical performance of an injured or disabled person in their daily activities. Prosthetic is an artificial substitute for a missing limb or part of a limb, which helps to gain independence after amputation. sEMG sensors can acquire the data from muscles during various daily activities of humans, viz. walking, standing, climbing, sitting and can automatically control the prosthesis and exoskeleton. Kiguchi et al. proposed a neuro-fuzzy controller for sEMG based upper limb robotic exoskeleton [11]. This exoskeleton can be used for elderly people, injured, disabled, and physically challenged people in their daily activities. Krasin et al. developed a low-cost elbow joint powered exoskeleton based on sEMG for bicep brachii strength enhancement [12]. When the biceps muscle is under-loaded, it produces a separate EMG signal, and the exoskeleton automatically shifts to the helpful lifting activity until the muscle relaxes. A low-cost sEMG was controlled prosthetic arm developed by Sharmila et al. for the upper limb amputee [13]. The sEMG signals of muscles are recorded using sensors during various activities for autonomous control of prostheses, and subsequently, actuators are controlled using artificial intelligence approaches. Cai et al. used sEMG signals to recognize the upper limb motion pattern with a support vector machine algorithm for controlling a rehabilitation robot [14]. Pancholi et al. presented a real-time arm gesture recognition system for amputees [15]. In the last decade, a lot of work focused on the upper limb sEMG signal classification and pattern recognition compared to the lower limb sEMG signal [16–18]. Classification of the lower limb sEMG signal has been more challenging than the upper limb signals due to the complexity of the lower limb sEMG signals and the muscles are buried deep beneath the skin with considerable overlap between each other. In the last few years, many machine learning and deep learning models have been explored to recognize the activities of the lower limbs to control the exoskeleton or prosthesis.

In the last decade, several attempts have been made to identify lower limb activities using sEMG signals of different lower limb muscles. An overview of some very popular studies related to sEMG based lower limb activity recognition is discussed here. Souit et. al. designed a sEMG based lower limb exoskeleton for the gait control [19]. By using the sEMG signal of muscles during various activities, the exoskeleton can be controlled automatically using artificial intelligence techniques. An artificial leg designed by Xie et. al. is similar to the joint of the lower limb of the human body [20]. Researchers have proposed different methods for detecting knee abnormality using hand-crafted features from sEMG signals with different preprocessing techniques. Erkaymaz et al. analyzed the knee abnormality from sEMG signal by using an artificial neural network [21]. Miller et al. adopted LDA and SVM classifiers to recognize seven different kinds of walking modes for the amputee and non-amputee groups with an accuracy of 97.9% and 94.7%, respectively [22]. Naik et al. investigated the detection of three different lower limb movements by using ICA-EBM with a classification accuracy of 96.1% and 86.2% with and without knee abnormality, respectively [23].

In recent years, deep learning methods have been applied in several different areas. A deep learning approach does not require extracting the hand-crafted features. These algorithms create features and later classify them accordingly. A convolutional neural network (CNN) is a type of deep learning algorithm that takes an input image, creates features, assigns importance to various features, and differentiates one from the other. The variants of CNN such as 2D CNN (VGGNET, ALEXNET, RESNET, etc.) and 3D CNN have shown better performance in many applications where the input is an image [24]. 1D CNN has also shown better results in the area of text classification, music classification and other time-series data such as EEG signal, EMG signal or ECG signal [25, 26]. Gautam and others proposed a deep learning model based on transfer learning to classify the three different lower limb activities [27].

Motivated by the necessity for neural control of the artificial limb or lower limb exoskeleton, we explored different sorts of mobility activities that might be appropriately detected using sEMG signals acquired from healthy knee and abnormal knee subjects. The findings of this investigation will aid the creation of a neural-controlled mechanical exoskeleton with a wide range of activities for injured or disabled people in the future. In this work, 1D CNN with the voting mechanism is proposed for classifying the three different lower limb movements: walking, sitting and standing. A total of four 1D CNN models are implemented and a system is designed as an ensemble of 1D CNN models, employing a majority vote strategy. The proposed system considers the sEMG signal of lower limb muscles which is denoised with the help of a wavelet

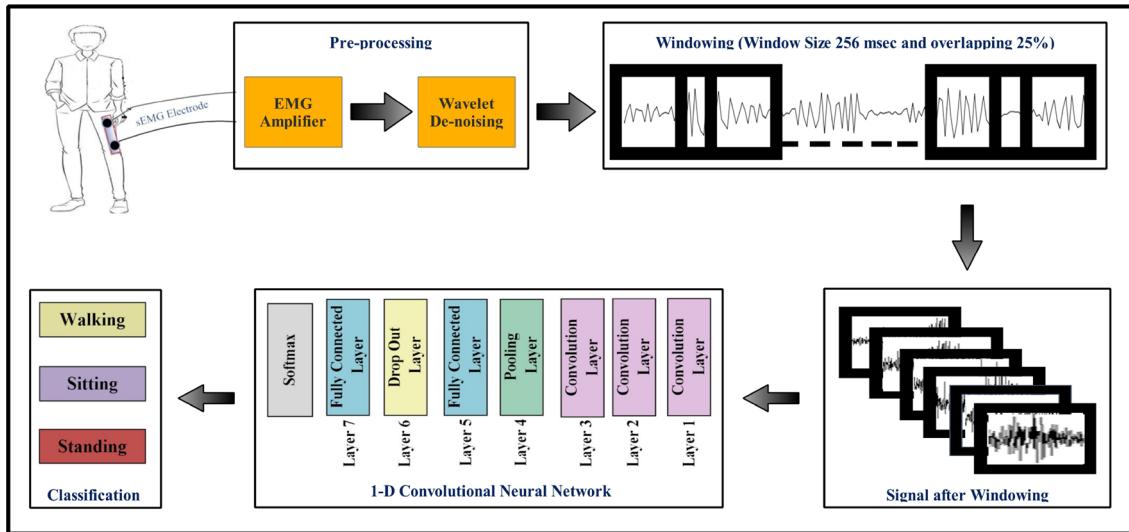


Fig. 1 Block diagram of the proposed methodology for lower limb activity classification

denoising technique and then segments it with a sliding window. After this, each segment is passed to all four 1D CNN models that process it to give the local decision to the majority-voting module. Then, a final decision will be taken by a voting-based 1D CNN classifier. If the result of a voting classifier is a tie then we consider the result of that individual model that has the highest accuracy.

The Research gap addressed in the study are:

1. In several studies, handcrafted features such as time-domain features, frequency domain features, etc. have been extracted. The main drawback of the handcrafted feature is that the classification performance is dependent on the selection of features.
2. According to the literature survey, the single architecture of the deep learning model is used to identify the movements of the lower limbs using the sEMG signal. Multiple architectures of deep learning models with a Voting-based mechanism are not explored in the literature which helps to improve the performance of the model by combining multiple model results.
3. The development of a neural-controlled artificial lower limb or exoskeleton requires higher accuracy for recognition of the activities of the lower limb.

The major contribution of this research are:

1. Lower limb activity recognition (LLAR) using sEMG signals acquired from individuals with three different cases: normal, knee abnormal, and pooled dataset.
2. A voting-based 1D CNN model is proposed, in which prediction is done via a voting mechanism of all the four 1D CNN model results.

3. Wavelet denoising is used as a part of preprocessing so that artifacts can be removed from the raw sEMG signal of lower limb muscles.
4. The proposed voting-based 1D CNN model is a high performance compared to the existing literature.

Methodology

The proposed system for lower limb activity recognition using sEMG signal based on deep learning is shown in Fig. 1. Different preprocessing techniques viz. denoising, windowing are applied before passing the signal to the 1D CNN to improve the accuracy of the model. In this study, the initial signal is denoised using the wavelet denoising technique. The nature of the sEMG signal is stochastic, therefore, the data augmentation method is used to minimize this problem. This sEMG signal corresponding to a normal or abnormal case is divided into the overlapping window and each window is treated as an independent instance to train the 1D CNN model. The final prediction is done via a voting mechanism using the majority of all the four model results.

Dataset

In this work, the data is taken from the UCI machine learning open source repository by Sanchez et al for the LLAR [28]. We have considered publicly available datasets for the classification of lower limb activity at the UCI machine learning repository. For the collection of the dataset, an experiment was performed considering the subjects of age above 18 years. We considered 22 subjects to perform the lower limb movements such as walking, sitting, and

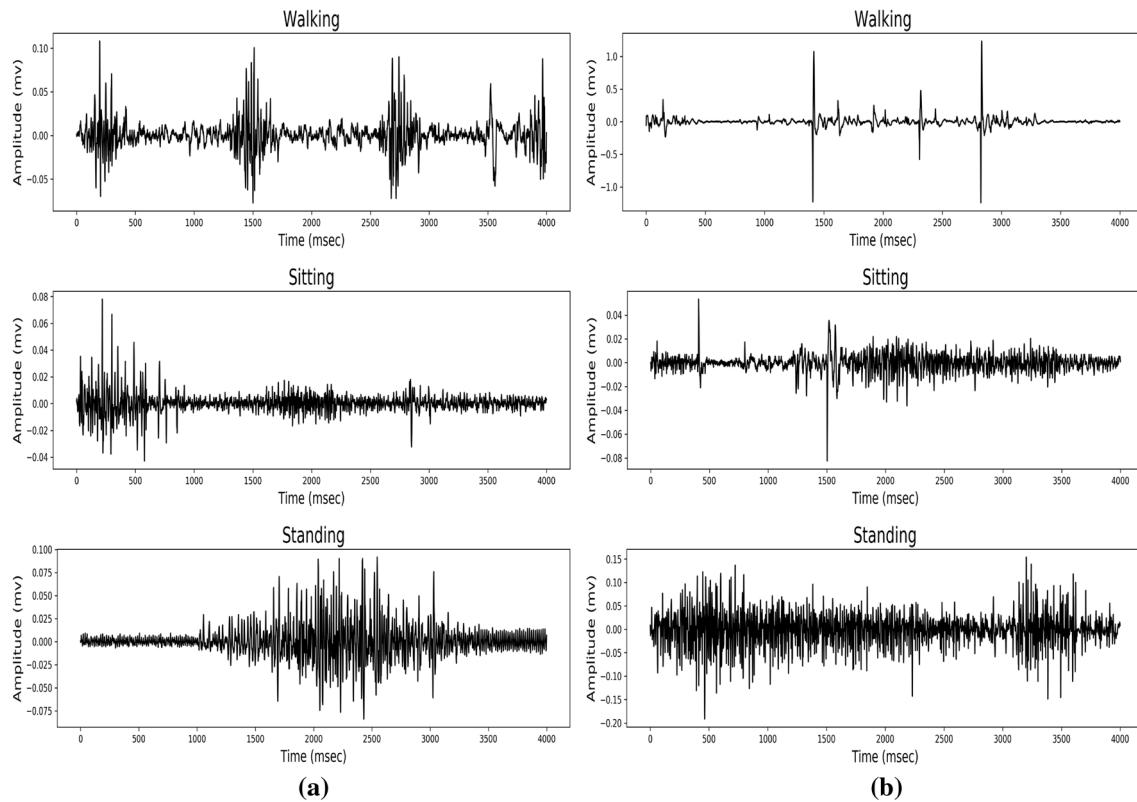


Fig. 2 sEMG signal taken for three different postures (walking, standing, sitting): **a** normal subject **b** abnormal subject [29]

standing to observe sEMG signal data of lower limb, out of which 50% of data were collected from healthy subjects having no previous record of knee detriment and left from knee abnormal subjects ones having knee irregularities like sciatic nerve, anterior cruciate ligament, and meniscus injury. The metal connector used for collecting data and monitoring lower limb movements portably is DataLog MWX8 latest data acquisition technology and a goniometer is used in the data collection process. The four lower limb muscles taken into account for the placement of wearable sensors and collection of sEMG signals are vastus medialis (VM), biceps femoris (BF), semitendinosus(ST), and rectus femoris (RF) with a goniometer connected to the knee joint externally. The sEMG signals are collected from the left limb of 11 healthy subjects and the damaged limb of 11 unhealthy subjects. The signals captured in the transition period are ignored for this investigation such as sitting to standing, standing to walk. The sEMG signal data has a 14-bit resolution and is sampled at 1kHz sampling frequency and then passed through a band-pass filter having a frequency range 20-460Hz it was therefore transferred to the computer from the MWX8 device. Figure 2a as well as Fig. 2b provide an example of sEMG recordings from normal and knee abnormal subjects in three distinct postures.

Wavelet denoising

Several types of noise are produced during the recording of a sEMG signal, and due to these artifacts or noises, the identity of the actual signal is lost. The features of the sEMG signal depend on various factors like skin formation, the structure of the body, tissue structure, skin temperature, measuring site, etc. These factors produced different types of noises and affect the diagnosis which is based on sEMG signals. Some of these noises are: ambient noise due to electromagnetic devices, inherent noise produced by electronic components, motion artifacts noise due to the motion of electrodes during the exercise or movement or lossy interface between the electrodes, and skin of the body, inherent noise instability due to the firing rate of the motor units [30]. The amplitude range of the sEMG signal is 0 V to 10 mV and the frequency range is around 10 Hz to 500 Hz [7]. If the frequency range of noise is not in the frequency range of the sEMG signal then it can be removed by using filtering techniques. Random noises such as white gaussian noise of sEMG signal are difficult to minimize by using the filtering process. It can be effectively removed with the help of wavelet denoising.

Over the past few years, wavelet denoising for the upper and lower limb sEMG signals has been a huge success.

Phinyomark et al. gave an idea about the denoising of the sEMG signal by using a wavelet denoising algorithm [31]. This algorithm has five parameters: (1) type of wavelet function or mother wavelet, (2) the scale, (3) threshold selection rule, (4) thresholding rescaling method and, (5) threshold function. There are 324 wavelet functions from 15 different wavelet families [30]. At various levels, wavelet families include haar, daubechies, coiflet, symlet, and others. The distinct wavelet coefficients are obtained by passing the signal through high-pass (Detail Coefficient) and low-pass (Approximate Coefficient) filters when DWT is applied to the signals. The number of coefficients depends on the level of decomposition. The approximate/detail coefficients have been obtained after the decomposition of the signal by using wavelet denoising. Small details coefficients can be removed from the equation to eliminate noise without sacrificing vital information. Thresholding, on the other hand, sets all of the coefficients to zero if they are less than a certain value. There are two popular methods of thresholding, hard and soft thresholding, some shortcomings have been found in both of them [32]. Jing-yi et al. have proposed garotte thresholding for denoising the signal [33].

In this study, wavelet denoising is used with db7 from the daubechies family till fourth level decomposition [34]. Garotte thresholding has been applied to detail coefficients D2, D3 and, D4 as illustrated in Fig. 3.

Windowing

The nature of the sEMG signal is stochastic, therefore, segmentation of the signal is necessary rather than the full signal. Different length of sEMG data affects the classification accuracy of the model. Segmentation can be done by using the windowing techniques which are classified into

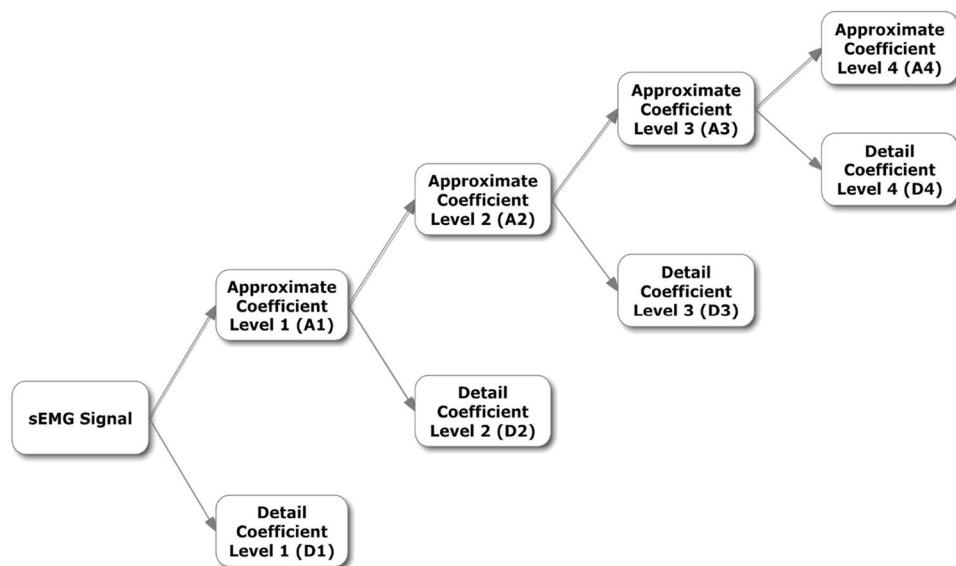
two types as shown in Fig. 4: adjacent windowing and overlapping windowing [7, 35]. In the overlap window, the next segment overlaps with the previous segment, while in the adjacent window no segments overlap with each other. In this study, we have considered 256 msec window length with 25% overlapping [23].

1D convolutional neural network

Convolutional neural networks are very similar to simple neural networks and they are biologically motivated from the feed-forward artificial neural networks. The preprocessing of data and feature extraction from data is much lower in CNN as compared to other machine learning algorithms. Conversely, CNN is mainly used to operate 2D data such as videos and images. This is the reason why CNN is often referred to as 2D CNNs. Some of 1D CNN as a modification of 2D CNNS are recently invented [26, 36]. The detailed working of the CNN model which has been used in this study is shown in Fig. 1.

In this investigation, Four channels were used for the data collection at a sampling frequency of 1 kHz. The overlapping windowing with 256 msec window length (256 samples) is considered and data of four channels are sorted serially, so the array size of the signal is 1×1024 for the input of 1D CNN. First, the input signal is normalized between 0 to 1. After normalization, the signal is passed through the three convolutional blocks which consist of a convolutional layer and nonlinear activation function (ReLU). The output of the ReLU layer is processed through the Max Pooling layer and then passed through a fully connected layer (FC1), dropout layer, and fully connected layer (FC2). The Dropout layer has been used to avoid the overfitting problem. The ReLU function is used in FC1 while the softmax function has been used in FC2. The

Fig. 3 Wavelet decomposition



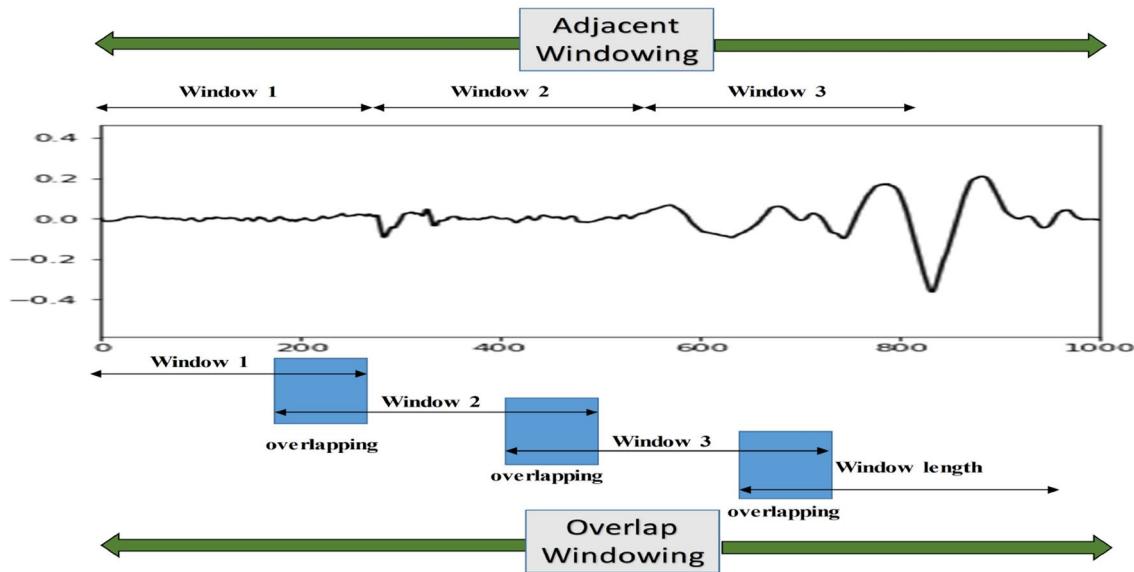


Fig. 4 Windowing techniques

softmax function has been used as a classifier and predicts the class of the input signal.

The input to the k th neuron of i th layer is given as:

$$x_k^l = \sum_{i=1}^{N_{l-1}} \text{Conv1D}(W_{ik}^{l-1}, o_i^{l-1}) + b_k^l \quad (1)$$

Where x_k^l is the input to k th neuron of i th layer; b_k^l is the bias of k th neuron of i th layer; N_{l-1} is number of neurons in the $(l-1)$ th layer and W_{ik}^l is the filter(kernel) from i th neuron of layer $l-1$ to the k th neuron of layer l . After applying activation function to x_k^l output is:

$$y_k^l = f(x_k^l) \quad (2)$$

where $f(x_k^l)$ is the activation function and we have used ReLU function which is given as:

$$y_k^l = \max(0, x_k^l) \quad (3)$$

It basically replaces all negative values by a zero. After applying max pooling operation to y_k^l , output is:

$$o_k^l = \text{for each pooling window } \max(y_k^l[0], y_k^l[1], \dots, y_k^l[\text{poolsize}]) \quad (4)$$

Loss function used to do backpropagation is given as:

$$L = - \sum_{i=1}^C t_i \log(f(s)_i) \quad (5)$$

where C is the total number of classes in the dataset; t_i is the target output for input instance and $f(s)_i$ is the probability

calculated by softmax function measuring chances of input instance to belong to i th class. Parameter updation in back-propagation is given as:

$$w_{ik}^{l-1}(t+1) = w_{ik}^{l-1}(t) - \alpha * \frac{\delta L}{\delta w_{ik}^{l-1}} \quad (6)$$

$$b_k^l(t+1) = b_k^l(t) - \alpha * \frac{\delta L}{\delta b_k^l} \quad (7)$$

where α is a learning rate hyperparameter.

A CNN model is commonly structured by adopting a coarse to fine approach, in which high-level layers have a large number of kernels and low-level layers contain a small number of kernels. Ullah et al. [37] proposed a pyramid architecture in which low-level layers contain a large number of kernels and high-level layers have a small number of kernels. For voting, a total of four 1D CNN models have been implemented as shown in Table 1. Model M1 and M2 are similar to the commonly structured CNN model in which the number of filters is in decreasing order (24, 16, 8) while in the other two models, M3 and M4 are similar to the pyramid structured CNN model in which the number of filters is in increasing order (8, 16, 24). In the models M1 and M3, the dropout layer is not considered while the other two models M2 and M4 considered 0.5 dropouts which are used to avoid the overfitting problem. The other parameters in Table 1 are derived through the trial-and-error method.

In this work, we have proposed a voting-based 1D CNN classifier. In the voting-based classifier (shown in Fig. 5), the accuracy is calculated with the individual models (M1, M2, M3, and M4) and then the final decision

Table 1 Specification of 1D CNN models

Model parameters		M1	M2	M3	M4
Convolution Layer 1	Number of filters	24	24	8	8
	Kernel size	5	5	5	5
Convolution Layer 2	Number of filters	16	16	16	16
	Kernel size	3	3	3	3
Convolution Layer 3	Number of filters	8	8	24	24
	Kernel size	3	3	3	3
Pooling Layer	Type of pooling	Max	Max	Max	Max
	Kernel size	2	2	2	2
Fully connected Layer (FC1)	–	100	100	100	100
Dropout layer	–	–	0.5	–	0.5
Fully connected Layer (FC2)	–	3	3	3	3

is taken by using the majority results of the four models. If the result of a voting classifier is a tie, then we consider the result of that individual model that has the highest accuracy. The model has to be trained with a training dataset initially. It requires the weight parameters to be learned from the data. In this study, we have used Adam optimizer [38] which consists of the traditional backpropagation technique with cross-entropy loss function [39]. It adapts faster when compared to other optimizers like SGD. We used default values of all six hyperparameters of adam algorithm which are: learning rate (.001), epsilon (0.00000001), beta1 (0.9), beta2 (0.999), and used locking (false). TensorFlow (“TensorFlow, 2017,”) has been used to implement the model. It is a freely available Deep Learning library from Google. We trained our models with 50 iterations which are enough to show a good empirical convergence.

Performance measures

In classification techniques, the classifier gives four outcomes which are: True Positive (TP), i.e., correct positive prediction, True Negative (TN), i.e., correct negative prediction, False Positive (FP), i.e., incorrect positive prediction, and False Negative (FN), i.e., incorrect negative prediction. A confusion matrix is formed from these four outcomes of classification. Our model’s performance on our test dataset may be seen using the confusion matrix. We have three classes in this problem: walking (W), standing (T), and sitting (S). The confusion matrix will be formed as:

$$C = \begin{bmatrix} C_{WW} & C_{WS} & C_{WT} \\ C_{SW} & C_{SS} & C_{ST} \\ C_{TW} & C_{TS} & C_{TT} \end{bmatrix} \quad (8)$$

where: C_{WW} : walking samples predicted as walking, C_{WS} : walking samples predicted as sitting, C_{WT} : walking samples predicted as standing, C_{SW} : sitting samples predicted as walking, C_{SS} : sitting samples predicted as sitting, C_{ST} : sitting samples predicted as Standing, C_{TW} : Standing samples predicted as Walking, C_{TS} : Standing samples predicted as sitting, C_{TT} : standing samples predicted as standing

The mathematical expression of studied performance parameters is shown in Table 2 [40].

Results

In this section, we present our results for different cases related to lower limb activity recognition. The proposed model was trained and tested with a cloud-based platform, google colab to recognize the lower limb activities of a human. In the colab notebook, the model’s code runs on a cloud server provided by Google, which means use google hardware including graphics processing units (GPUs) and

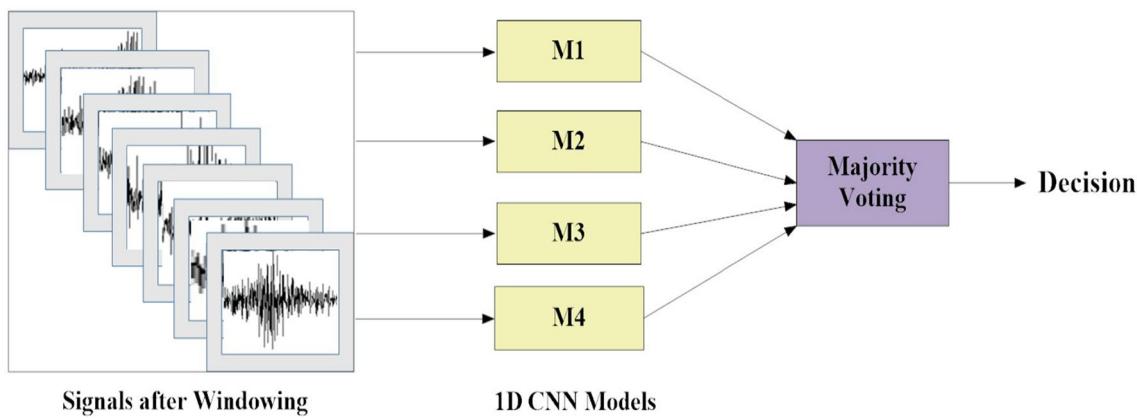
**Fig. 5** Block diagram of voting classifier

Table 2 Mathematical expression of performance measures

Performance measures	Walking	Sitting	Standing
Accuracy	$A_W = \frac{C_{WW}}{C_{WW} + C_{WS} + C_{WT}}$	$A_S = \frac{C_{SS}}{C_{SW} + C_{SS} + C_{ST}}$	$A_T = \frac{C_{TT}}{C_{TW} + C_{TS} + C_{TT}}$
Specificity	$SP_W = \frac{C_{SS} + C_{ST} + C_{TS} + C_{TT}}{C_{SW} + C_{SS} + C_{ST} + C_{TW} + C_{TS} + C_{TT}}$	$SP_S = \frac{C_{WW} + C_{WT} + C_{TW} + C_{TT}}{C_{WW} + C_{WS} + C_{WT} + C_{TW} + C_{TS} + C_{TT}}$	$SP_T = \frac{C_{WW} + C_{WS} + C_{SW} + C_{SS}}{C_{WW} + C_{WS} + C_{WT} + C_{SW} + C_{SS} + C_{ST}}$
Sensitivity	$SE_W = \frac{C_{WW}}{C_{WW} + C_{WS} + C_{WT}}$	$SE_S = \frac{C_{SS}}{C_{SW} + C_{SS} + C_{ST}}$	$SE_T = \frac{C_{TT}}{C_{TW} + C_{TS} + C_{TT}}$
Precision	$P_W = \frac{C_{WW}}{C_{WW} + C_{SW} + C_{TW}}$	$P_S = \frac{C_{SS}}{C_{WS} + C_{SS} + C_{TS}}$	$P_T = \frac{C_{TT}}{C_{WT} + C_{ST} + C_{TT}}$
F-score	$F_W = \frac{2 * SE_W * P_W}{SE_W + P_W}$	$F_S = \frac{2 * SE_S * P_S}{SE_S + P_S}$	$F_T = \frac{2 * SE_T * P_T}{SE_T + P_T}$

tensor processing units (TPUs), regardless of the capacity of our machine. The keras libraries of python were used to implement the different models of 1D CNN and results were calculated.

In this study, 11 healthy people and 11 people with knee problems were investigated, and performance metrics for three lower limb movements are computed such as walking, sitting, and standing in three different cases: (1) healthy individuals, (2) individuals with knee abnormality, and (3) pooled data with healthy and knee abnormal individual. The activity signal of the subjects is denoised with the help of discrete wavelet denoising and then divided into training, validation, and testing sets with threefold nested cross-validation technique as shown in Fig. 6. When dealing with time-series datasets, the nested cross-validation technique is utilised to reduce temporal dependencies and arbitrary test set selection. The sEMG signal is time-series data in this case. As a result, we used the 3-fold nested cross-validation method. An overlapping window with a window size of 256 msec and a 25% overlap was chosen for signal segmentation.

Then, using the training dataset, multiple architectures of 1D CNN models were trained, and the parameters that minimised error on the validation set were chosen. The model is then trained using the entire training set, and the performance parameters of several models using the testing dataset are determined. The final prediction is done via a voting mechanism using all four model results.

Table 3 shows the comparison of the performance indices of the various architecture of the 1D CNN model with the five performance indices outlined in “Performance Measures” section. In this study, we also evaluated various performance indices for 10 runs to check the variability of the results. Table 4 presents the class-wise accuracy of the studied architecture of 1D CNN models and voting-based 1D CNN classifier for classification between three different exercises of walking, sitting, and standing with normal, abnormal, and pooled data. Accuracy can be considered as one of the evaluation parameters but there is a need for other evaluation parameters such as sensitivity, specificity, precision, and F-score; considered in this study. Table 5 shows

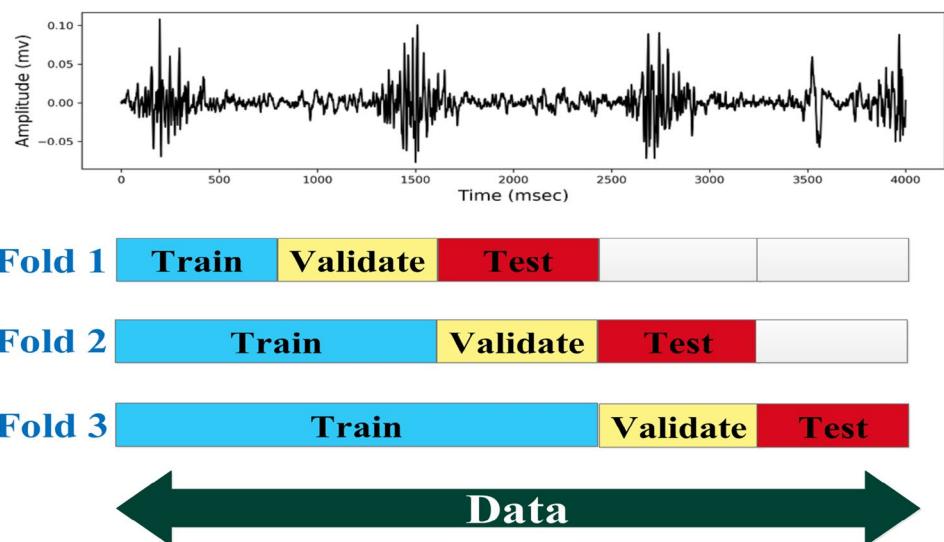
Fig. 6 Threefold nested cross-validation

Table 3 Performance obtained by studied 1D CNN classifiers on the different type of subjects (best values found in bold)

Subject	Performance Parameter	M1	M2	M3	M4	Voting
Normal	Accuracy	99.28	98.71	99.19	99.00	99.35
	Specificity	99.61	99.31	99.56	99.47	99.64
	Sensitivity	98.74	97.56	98.50	98.22	98.80
	Precision	99.24	98.64	99.16	98.90	99.38
	F-score	98.99	98.10	98.83	98.56	99.09
Abnormal	Accuracy	96.43	95.04	96.02	95.34	97.63
	Specificity	98.20	97.51	97.97	97.65	98.81
	Sensitivity	96.15	94.74	95.66	95.00	97.45
	Precision	96.32	94.87	96.01	95.23	97.54
	F-score	96.23	94.81	95.83	95.11	97.49
Pooled data	Accuracy	95.57	94.39	93.90	95.20	97.14
	Specificity	97.82	97.22	96.91	97.62	98.59
	Sensitivity	95.61	94.39	93.80	95.18	97.11
	Precision	95.48	94.28	94.01	95.11	97.05
	F-score	95.55	94.33	93.90	95.14	97.08

Table 4 Class-wise accuracy of different 1D CNN models (best values found in bold)

Model	Normal			Abnormal			Pooled data		
	Walking	Sitting	Standing	Walking	Sitting	Standing	Walking	Sitting	Standing
M1	96.81	99.46	99.93	98.53	93.93	95.99	97.94	93.59	95.3
M2	93.48	99.21	100	96.92	92.08	95.23	96.4	92.18	94.6
M3	96.04	99.49	99.95	98.97	92.95	95.06	94.59	91.19	95.6
M4	95.48	99.21	99.98	98.52	92.66	93.82	96.76	92.96	95.8
Voting	96.86	99.57	99.98	99.06	96.04	97.24	98.06	95.65	97.63

Table 5 The mean accuracies of different 1D CNN models for 10 runs (best values found in bold)

Dataset	Model	1	2	3	4	5	6	7	8	9	10	Mean	Std deviation
Normal	M1	99.00	99.71	99.29	99.29	99.57	99.43	99.07	98.86	99.50	99.07	99.28	0.28
	M2	99.21	99.36	98.86	98.36	98.43	98.86	98.07	99.00	98.07	98.86	98.71	0.45
	M3	99.43	99.07	99.21	98.86	99.64	99.29	98.93	99.21	98.93	99.29	99.19	0.25
	M4	98.86	98.50	98.79	99.29	98.93	99.14	99.36	99.29	98.86	99.00	99.00	0.27
	Voting	99.21	99.43	99.43	99.29	99.57	99.50	99.07	99.43	99.21	99.36	99.35	0.15
Abnormal	M1	96.13	97.95	98.14	97.33	93.77	97.10	96.05	96.44	94.97	96.40	96.43	1.33
	M2	94.85	95.97	96.09	94.85	91.75	93.34	96.32	96.01	94.77	96.40	95.04	1.50
	M3	96.44	96.17	95.08	95.35	96.83	96.05	97.72	95.43	95.66	95.43	96.02	0.81
	M4	96.52	93.34	96.09	95.86	94.31	94.70	95.59	95.59	96.01	95.43	95.34	0.96
	Voting	97.68	98.03	97.91	97.60	97.29	97.75	97.91	97.48	97.25	97.37	97.63	0.27
Pooled	M1	94.10	94.90	96.59	96.39	94.58	95.76	97.01	96.13	96.13	94.10	95.57	1.06
	M2	94.28	91.97	94.68	95.73	94.93	95.16	94.70	93.90	93.83	94.68	94.39	1.02
	M3	96.69	95.96	95.01	94.95	95.38	96.36	95.80	96.64	95.03	97.16	95.90	0.80
	M4	93.80	95.93	96.41	96.06	95.93	94.55	94.55	94.40	96.61	93.72	95.20	1.10
	Voting	97.14	96.84	97.16	97.19	96.91	97.09	97.34	97.21	96.89	97.59	97.14	0.23

the average classification accuracy along with their standard deviation of the voting classifier for 10 runs of a normal, abnormal, and pooled dataset.

Discussion

In this present study, automated lower limb activities were recognized with a voting-based 1D CNN model using the sEMG signal. Table 4 shows that the voting classifier gives the highest performance indices value with all three movements of abnormal and pooled data while the accuracy of walking and standing movements in healthy subjects is higher than in the other studied models and the accuracy of sitting activity is 99.98% which is slightly less than the model M2 which is 100%. The difference is very small, so we can say that the proposed voting classifier gives a better performance than the other models for all three activities with healthy, knee abnormal, and pooled data.

In this study, we also evaluated the performance indices of the various architecture of the 1D CNN model and compared them in Table 3. It shows that the voting-based 1D CNN model obtained the highest performance indices compared to other studied models and existing research work. Table 5 shows the average classification accuracy along with their standard deviation of the voting classifier for 10 runs of normal, abnormal, and pooled data is 99.35 ± 0.15 , 97.63 ± 0.27 and 97.14 ± 0.23 respectively.

Figure 7 depicts the confusion matrices for the normal, abnormal, and pooled data with three different exercises. The confusion matrix helps conceptualize the performance of the classification algorithm in a tabular form. Figure 7a shows the confusion matrix obtained for the model M1 of the 1D CNN classifier. It can be seen that 96.8%, 99.5%, 99.9% samples are correctly predicted as walking, sitting, and standing, respectively while 67 samples of walking, 30 samples of sitting, and 4 samples of standing are incorrectly classified into the other activities. Figure 7b–o also show the confusion matrix of other models that contains the information about the true and predicted labels evaluated by a model.

Figure 8a–c show the plot between loss vs epoch for the four different 1D CNN models with test and train data. These plots show when increasing the number of epoch than the value of the loss function decreases and after some epoch, it reaches a steady-state, which means the over-fitting problem reduces. From these figures, we can say that 50 iterations are sufficient to show a good empirical convergence. So, 50 iterations are considered to run the various architecture of 1D CNN which are studied in this work.

Lower limb activity recognition has been approached in a variety of ways. Herrera-Gonzalez et al. used MP-ANN to construct a classifier for three different lower limb

movements, with an accuracy of 88, 94, and 92 % for walking, sitting, and standing tasks, respectively [41]. The performance parameters estimated in this technique have been shown in Table 6. Using the Empirical Mode Decomposition (EMD) based technique, Zhang et al. identified distinct lower limb movements of healthy volunteers as shown in Table 6 [42].

Naik et al. built a classifier that can accurately categorise walking, sitting, and standing activities with an accuracy of 96.14 % for a healthy person and 86.17 % for a person with a knee deformity, respectively. [23]. Gautam and others proposed a deep learning model based on transfer learning to classify the walking, sitting, and standing activities with an accuracy of 98.2%, 97.7%, and 98.4% for healthy and 92.8%, 92.3%, and 92.2% for walking, sitting and standing tasks for individual suffering from the knee abnormality [27]. The sEMG data of lower limb muscles that we used in our investigation is the same as that of the other authors whose findings were discussed. Table 6 presents the comparative performance of the proposed model with the other literature studies that show the proposed voting-based 1D CNN model is high performance for lower limb activity recognition.

Conclusion

In this work, a voting-based 1D CNN model has been proposed for automatic detection of lower limb movements using sEMG signal for a subject suffering from knee abnormality as well as a healthy subject. The previous study on sEMG based on three different lower limb movements walking, standing and sitting have poor accuracy. For this reason, the proposed work focuses on improving classification accuracy by incorporating the deep learning model. In the deep learning algorithm, we don't need to extract the handcrafted features. These algorithms first create the features by using deep neural networks and then classify them. First, we have denoised the raw sEMG signal by using discrete wavelet denoising. To minimize the problem of the small dataset, we have introduced overlapping windowing techniques for data segmentation and then implemented four different 1D CNN models and then the final decision takes by using the results of the four individual models. The proposed method achieves an average classification accuracy of 99.35%, 97.63%, and 97.14% for healthy subjects, abnormal subjects, and pooled data respectively with a voting-based 1D CNN classifier.

The projected work will continue to be extended in the future. For starters, the dataset employed in the study

Fig. 7 Confusion matrices with healthy, knee abnormal and pooled dataset

Model		Normal	Abnormal	Pooled																																				
M1	Actual	<table border="1"> <tr> <td>Walking</td><td>96.8% 2033/2100</td><td>3.1% 66</td><td>0.0% 1</td></tr> <tr> <td>Sitting</td><td>0.3% 15</td><td>99.5% 5505/5580</td><td>0.3% 15</td></tr> <tr> <td>Standing</td><td>0.0% 2</td><td>0.0% 2</td><td>99.9% 6326/6330</td></tr> </table>	Walking	96.8% 2033/2100	3.1% 66	0.0% 1	Sitting	0.3% 15	99.5% 5505/5580	0.3% 15	Standing	0.0% 2	0.0% 2	99.9% 6326/6330	<table border="1"> <tr> <td>Walking</td><td>98.5% 9982/10110</td><td>1.0% 104</td><td>0.4% 44</td></tr> <tr> <td>Sitting</td><td>3.5% 248</td><td>93.9% 6594/7020</td><td>2.5% 178</td></tr> <tr> <td>Standing</td><td>1.1% 99</td><td>2.9% 250</td><td>96.0% 8355/8700</td></tr> </table>	Walking	98.5% 9982/10110	1.0% 104	0.4% 44	Sitting	3.5% 248	93.9% 6594/7020	2.5% 178	Standing	1.1% 99	2.9% 250	96.0% 8355/8700	<table border="1"> <tr> <td>Walking</td><td>97.9% 11958/12210</td><td>1.5% 178</td><td>0.6% 74</td></tr> <tr> <td>Sitting</td><td>4.5% 563</td><td>93.6% 11792/12600</td><td>1.9% 245</td></tr> <tr> <td>Standing</td><td>2.9% 330</td><td>1.6% 275</td><td>95.3% 14325/15030</td></tr> </table>	Walking	97.9% 11958/12210	1.5% 178	0.6% 74	Sitting	4.5% 563	93.6% 11792/12600	1.9% 245	Standing	2.9% 330	1.6% 275	95.3% 14325/15030
Walking	96.8% 2033/2100	3.1% 66	0.0% 1																																					
Sitting	0.3% 15	99.5% 5505/5580	0.3% 15																																					
Standing	0.0% 2	0.0% 2	99.9% 6326/6330																																					
Walking	98.5% 9982/10110	1.0% 104	0.4% 44																																					
Sitting	3.5% 248	93.9% 6594/7020	2.5% 178																																					
Standing	1.1% 99	2.9% 250	96.0% 8355/8700																																					
Walking	97.9% 11958/12210	1.5% 178	0.6% 74																																					
Sitting	4.5% 563	93.6% 11792/12600	1.9% 245																																					
Standing	2.9% 330	1.6% 275	95.3% 14325/15030																																					
(a)	(b)	(c)																																						
M2	Actual	<table border="1"> <tr> <td>Walking</td><td>93.5% 1963/2100</td><td>6.0% 126</td><td>0.5% 11</td></tr> <tr> <td>Sitting</td><td>0.5% 29</td><td>99.2% 5336/5380</td><td>0.3% 15</td></tr> <tr> <td>Standing</td><td></td><td></td><td>100.0% 6330/6330</td></tr> </table>	Walking	93.5% 1963/2100	6.0% 126	0.5% 11	Sitting	0.5% 29	99.2% 5336/5380	0.3% 15	Standing			100.0% 6330/6330	<table border="1"> <tr> <td>Walking</td><td>96.9% 9789/10110</td><td>1.8% 185</td><td>1.2% 126</td></tr> <tr> <td>Sitting</td><td>5.9% 273</td><td>92.1% 6464/7020</td><td>4.0% 383</td></tr> <tr> <td>Standing</td><td>1.6% 141</td><td>3.1% 274</td><td>95.2% 8285/8700</td></tr> </table>	Walking	96.9% 9789/10110	1.8% 185	1.2% 126	Sitting	5.9% 273	92.1% 6464/7020	4.0% 383	Standing	1.6% 141	3.1% 274	95.2% 8285/8700	<table border="1"> <tr> <td>Walking</td><td>96.4% 11770/12210</td><td>2.8% 344</td><td>0.8% 96</td></tr> <tr> <td>Sitting</td><td>4.7% 588</td><td>92.2% 11615/12600</td><td>3.2% 397</td></tr> <tr> <td>Standing</td><td>3.2% 485</td><td>2.2% 327</td><td>94.6% 14218/15030</td></tr> </table>	Walking	96.4% 11770/12210	2.8% 344	0.8% 96	Sitting	4.7% 588	92.2% 11615/12600	3.2% 397	Standing	3.2% 485	2.2% 327	94.6% 14218/15030
Walking	93.5% 1963/2100	6.0% 126	0.5% 11																																					
Sitting	0.5% 29	99.2% 5336/5380	0.3% 15																																					
Standing			100.0% 6330/6330																																					
Walking	96.9% 9789/10110	1.8% 185	1.2% 126																																					
Sitting	5.9% 273	92.1% 6464/7020	4.0% 383																																					
Standing	1.6% 141	3.1% 274	95.2% 8285/8700																																					
Walking	96.4% 11770/12210	2.8% 344	0.8% 96																																					
Sitting	4.7% 588	92.2% 11615/12600	3.2% 397																																					
Standing	3.2% 485	2.2% 327	94.6% 14218/15030																																					
(d)	(e)	(f)																																						
M3	Actual	<table border="1"> <tr> <td>Walking</td><td>96.0% 2037/2100</td><td>4.0% 83</td><td></td></tr> <tr> <td>Sitting</td><td>0.3% 16</td><td>99.5% 5552/5580</td><td>0.2% 12</td></tr> <tr> <td>Standing</td><td>0.0% 1</td><td>0.0% 2</td><td>100.0% 6327/6330</td></tr> </table>	Walking	96.0% 2037/2100	4.0% 83		Sitting	0.3% 16	99.5% 5552/5580	0.2% 12	Standing	0.0% 1	0.0% 2	100.0% 6327/6330	<table border="1"> <tr> <td>Walking</td><td>99.0% 10004/10110</td><td>0.7% 71</td><td>0.3% 33</td></tr> <tr> <td>Sitting</td><td>4.7% 327</td><td>92.9% 6525/7020</td><td>2.4% 168</td></tr> <tr> <td>Standing</td><td>1.9% 163</td><td>3.1% 267</td><td>95.1% 8270/8700</td></tr> </table>	Walking	99.0% 10004/10110	0.7% 71	0.3% 33	Sitting	4.7% 327	92.9% 6525/7020	2.4% 168	Standing	1.9% 163	3.1% 267	95.1% 8270/8700	<table border="1"> <tr> <td>Walking</td><td>94.6% 11540/12210</td><td>1.5% 199</td><td>3.9% 472</td></tr> <tr> <td>Sitting</td><td>3.9% 496</td><td>91.2% 11490/12600</td><td>4.9% 614</td></tr> <tr> <td>Standing</td><td>2.8% 423</td><td>1.6% 237</td><td>95.6% 14370/15030</td></tr> </table>	Walking	94.6% 11540/12210	1.5% 199	3.9% 472	Sitting	3.9% 496	91.2% 11490/12600	4.9% 614	Standing	2.8% 423	1.6% 237	95.6% 14370/15030
Walking	96.0% 2037/2100	4.0% 83																																						
Sitting	0.3% 16	99.5% 5552/5580	0.2% 12																																					
Standing	0.0% 1	0.0% 2	100.0% 6327/6330																																					
Walking	99.0% 10004/10110	0.7% 71	0.3% 33																																					
Sitting	4.7% 327	92.9% 6525/7020	2.4% 168																																					
Standing	1.9% 163	3.1% 267	95.1% 8270/8700																																					
Walking	94.6% 11540/12210	1.5% 199	3.9% 472																																					
Sitting	3.9% 496	91.2% 11490/12600	4.9% 614																																					
Standing	2.8% 423	1.6% 237	95.6% 14370/15030																																					
(g)	(h)	(i)																																						
M4	Actual	<table border="1"> <tr> <td>Walking</td><td>95.5% 2005/2100</td><td>4.5% 95</td><td></td></tr> <tr> <td>Sitting</td><td>0.5% 27</td><td>99.2% 5336/5380</td><td>0.3% 17</td></tr> <tr> <td>Standing</td><td>0.0% 1</td><td>0.0% 2</td><td>100.0% 6326/6330</td></tr> </table>	Walking	95.5% 2005/2100	4.5% 95		Sitting	0.5% 27	99.2% 5336/5380	0.3% 17	Standing	0.0% 1	0.0% 2	100.0% 6326/6330	<table border="1"> <tr> <td>Walking</td><td>98.5% 9980/10110</td><td>0.8% 83</td><td>0.7% 67</td></tr> <tr> <td>Sitting</td><td>4.3% 304</td><td>92.7% 6505/7020</td><td>3.0% 211</td></tr> <tr> <td>Standing</td><td>2.0% 173</td><td>4.2% 365</td><td>93.8% 8162/8700</td></tr> </table>	Walking	98.5% 9980/10110	0.8% 83	0.7% 67	Sitting	4.3% 304	92.7% 6505/7020	3.0% 211	Standing	2.0% 173	4.2% 365	93.8% 8162/8700	<table border="1"> <tr> <td>Walking</td><td>96.8% 11814/12210</td><td>2.7% 325</td><td>0.6% 71</td></tr> <tr> <td>Sitting</td><td>3.9% 494</td><td>93.0% 11714/12600</td><td>3.1% 392</td></tr> <tr> <td>Standing</td><td>2.7% 399</td><td>1.6% 232</td><td>95.8% 14399/15030</td></tr> </table>	Walking	96.8% 11814/12210	2.7% 325	0.6% 71	Sitting	3.9% 494	93.0% 11714/12600	3.1% 392	Standing	2.7% 399	1.6% 232	95.8% 14399/15030
Walking	95.5% 2005/2100	4.5% 95																																						
Sitting	0.5% 27	99.2% 5336/5380	0.3% 17																																					
Standing	0.0% 1	0.0% 2	100.0% 6326/6330																																					
Walking	98.5% 9980/10110	0.8% 83	0.7% 67																																					
Sitting	4.3% 304	92.7% 6505/7020	3.0% 211																																					
Standing	2.0% 173	4.2% 365	93.8% 8162/8700																																					
Walking	96.8% 11814/12210	2.7% 325	0.6% 71																																					
Sitting	3.9% 494	93.0% 11714/12600	3.1% 392																																					
Standing	2.7% 399	1.6% 232	95.8% 14399/15030																																					
(j)	(k)	(l)																																						
Voting	Actual	<table border="1"> <tr> <td>Walking</td><td>96.9% 2034/2100</td><td>3.1% 66</td><td></td></tr> <tr> <td>Sitting</td><td>0.2% 9</td><td>99.4% 5556/5580</td><td>0.3% 15</td></tr> <tr> <td>Standing</td><td></td><td>0.0% 1</td><td>100.0% 6329/6330</td></tr> </table>	Walking	96.9% 2034/2100	3.1% 66		Sitting	0.2% 9	99.4% 5556/5580	0.3% 15	Standing		0.0% 1	100.0% 6329/6330	<table border="1"> <tr> <td>Walking</td><td>99.1% 10015/10110</td><td>0.7% 75</td><td>0.2% 20</td></tr> <tr> <td>Sitting</td><td>2.8% 196</td><td>96.9% 6742/7020</td><td>1.2% 92</td></tr> <tr> <td>Standing</td><td>0.6% 48</td><td>2.2% 192</td><td>97.2% 8466/8700</td></tr> </table>	Walking	99.1% 10015/10110	0.7% 75	0.2% 20	Sitting	2.8% 196	96.9% 6742/7020	1.2% 92	Standing	0.6% 48	2.2% 192	97.2% 8466/8700	<table border="1"> <tr> <td>Walking</td><td>98.1% 11973/12210</td><td>1.6% 199</td><td>0.3% 38</td></tr> <tr> <td>Sitting</td><td>3.2% 399</td><td>95.7% 12052/12600</td><td>1.2% 149</td></tr> <tr> <td>Standing</td><td>1.7% 261</td><td>0.6% 95</td><td>97.6% 14674/15030</td></tr> </table>	Walking	98.1% 11973/12210	1.6% 199	0.3% 38	Sitting	3.2% 399	95.7% 12052/12600	1.2% 149	Standing	1.7% 261	0.6% 95	97.6% 14674/15030
Walking	96.9% 2034/2100	3.1% 66																																						
Sitting	0.2% 9	99.4% 5556/5580	0.3% 15																																					
Standing		0.0% 1	100.0% 6329/6330																																					
Walking	99.1% 10015/10110	0.7% 75	0.2% 20																																					
Sitting	2.8% 196	96.9% 6742/7020	1.2% 92																																					
Standing	0.6% 48	2.2% 192	97.2% 8466/8700																																					
Walking	98.1% 11973/12210	1.6% 199	0.3% 38																																					
Sitting	3.2% 399	95.7% 12052/12600	1.2% 149																																					
Standing	1.7% 261	0.6% 95	97.6% 14674/15030																																					
(m)	(n)	(o)																																						

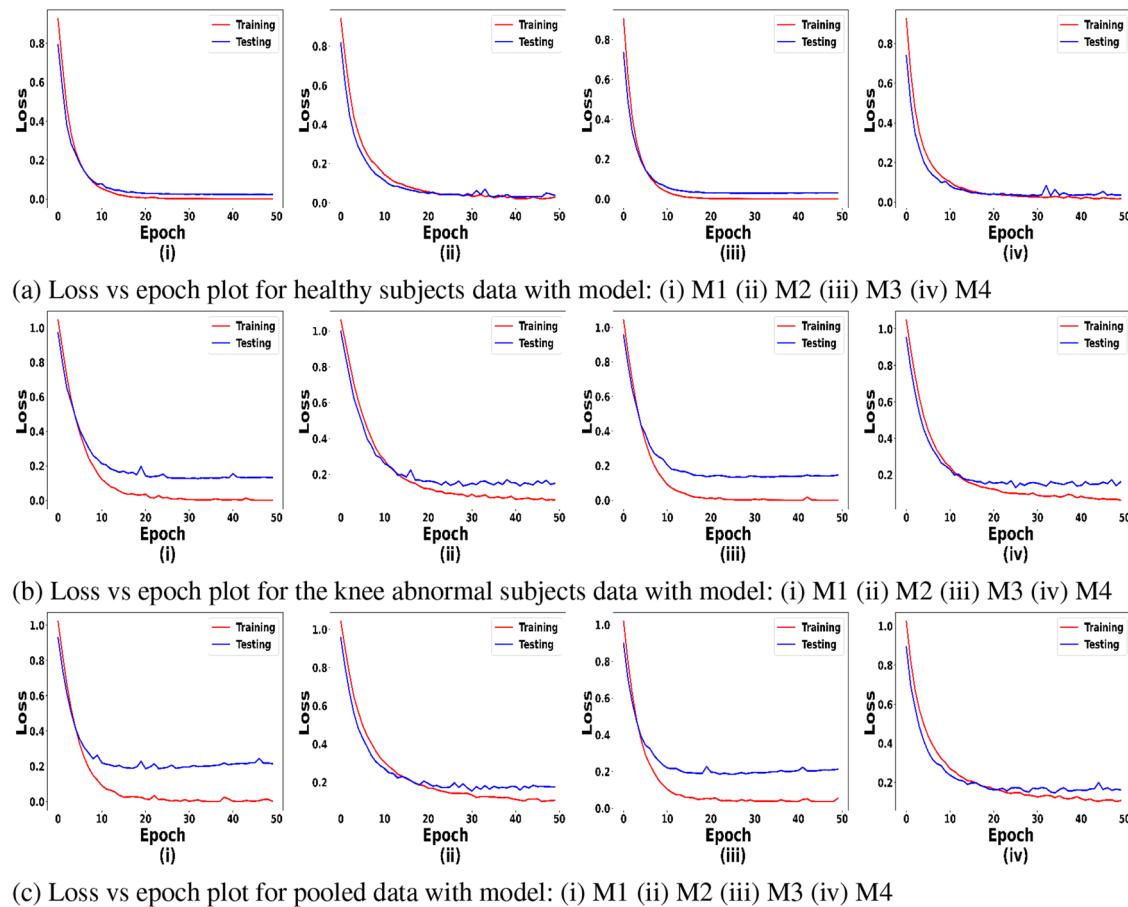


Fig. 8 Loss vs epoch plot for different 1D CNN models

Table 6 Comparative performance of the proposed model with the other literature studies (best values found in bold)

Approach	Subject	Walking	Sitting	Standing
EMD [42]	Healthy	64	67	69
MEMD [42]	Healthy	73	79	82
NA-EMD [42]	Healthy	79	83	83
MP-ANN [41]	Knee Abnormal	88	94	92
Transfer Learning based LRCN [27]	Healthy	98.2	97.7	98.4
	Knee Abnormal	92.8	92.3	92.2
ICA-EBM [23]	Healthy	96.0	96.2	96.2
	Knee Abnormal	86.6	86.4	85.5
Voting-based 1D CNN (proposed)	Healthy	96.86	99.57	99.98
	Knee abnormal	99.06	96.04	97.24
	pooled data	98.06	95.65	97.63

included data from a smaller number of patients. This method can be validated using a large number of subjects, reducing the issue of biasing caused by a small dataset. The suggested approach uses an offline dataset to validate the results. For clinical validation, more studies can be validated

with a real-time dataset. Besides this, the proposed system can also be used to control the prosthesis or artificial limb for lower limb amputees.

Acknowledgements This publication is supported by Visvesvaraya PhD Scheme, Meity, Govt. of India, MEITY-PHD-2942.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any study with human participants performed by any of the authors.

References

- Ranasinghe S, Al Machot F, Mayr HC (2016) A review on applications of activity recognition systems with regard to performance and evaluation. Int J Distrib Sensor Netw 12(8):1550147716665520

2. Liu X, Liu L, Simske SJ, Liu J (2016) Human daily activity recognition for healthcare using wearable and visual sensing data. 24–31
3. Lara OD, Labrador MA (2012) A survey on human activity recognition using wearable sensors. *IEEE Commun Surv Tutor* 15(3):1192–1209
4. Yang B-S, Liao S-T (2012) Fall detecting using inertial and electromyographic sensors. In: Proceedings of the 36th annual meeting of the American Society of Biomechanics, Gainsville, FL, USA, pp 15–18
5. Cheng J, Chen X, Shen M (2012) A framework for daily activity monitoring and fall detection based on surface electromyography and accelerometer signals. *IEEE J Biomed Health Inform* 17(1):38–45
6. Farina D, Negro F (2012) Accessing the neural drive to muscle and translation to neurorehabilitation technologies. *IEEE Rev Biomed Eng* 5:3–14
7. Nazmi N, Rahman A, Azizi M, Yamamoto S-I, Ahmad SA, Zamzuri H, Mazlan SA (2016) A review of classification techniques of EMG signals during isotonic and isometric contractions. *Sensors* 16(8):1304
8. Au SK, Bonato P, Herr H (2005) An EMG-position controlled system for an active ankle-foot prosthesis: an initial experimental study. In: 9th international conference on rehabilitation robotics, 2005 (ICORR 2005), pp 375–379. IEEE
9. Vijayvargiya A, Singh PL, Verma SM, Kumar R, Bansal S (2019) Performance comparison analysis of different classifier for early detection of knee osteoarthritis. *Sensors Health Monitor*. Elsevier, pp 243–257
10. Burkow-Heikkinen L (2011) Non-invasive physiological monitoring of exercise and fitness. *Neurol Res* 33(1):3–17
11. Kiguchi K, Tanaka T, Fukuda T (2004) Neuro-fuzzy control of a robotic exoskeleton with EMG signals. *IEEE Trans Fuzzy Syst* 12(4):481–490
12. Krasin V, Gandhi V, Yang Z, Karamanoglu M (2015) EMG based elbow joint powered exoskeleton for biceps Brachii strength augmentation. pp 1–6
13. Sharmila K, Sarath TV, Ramachandran KI (2016) EMG controlled low cost prosthetic arm. pp 169–172
14. Cai S, Chen Y, Huang S, Yan W, Zheng H, Li X, Xie L (2019) SVM-based classification of sEMG signals for upper-limb self-rehabilitation training. *Front Neurorobot* 13:31
15. Pancholi S, Joshi AM (2019) Electromyography-based hand gesture recognition system for upper limb amputees. *IEEE Sensors Lett.* 3(3):1–4
16. Eisenberg GD, Fylie KGHM, Mohamed A-K (2017) Real-time segmentation and feature extraction of electromyography: towards control of a prosthetic hand. *IFAC-PapersOnLine* 50(2):151–156
17. Mukhopadhyay AK, Samui S (2020) An experimental study on upper limb position invariant EMG signal classification based on deep neural network. *Biomed Signal Process Control* 55:101669
18. Karabulut D, Faruk O, Yunus ZA, Mehmet AA (2017) Comparative evaluation of EMG signal features for myoelectric controlled human arm prosthetics. *Biocybern Biomed Eng* 37(2):326–335
19. Souit C, Coelho DS, Szylit M, Camargo-Junior F, Junior Milton PC, Forner-Cordero A (2016) Design of a lower limb exoskeleton for experimental research on gait control. In: 2016 6th IEEE international conference on biomedical robotics and biomechatronics (BioRob), pp 1098–1103. IEEE
20. Xie H, Li Z, Li F (2020) Bionics design of artificial leg and experimental modeling research of pneumatic artificial muscles. *J Robot*
21. Erkaymaz O, Şenyer İ, Uzun R (2017) Detection of knee abnormality from surface EMG signals by artificial neural networks. pp 1–4
22. Miller JD, Beazer MS, Hahn ME (2013) Myoelectric walking mode classification for transtibial amputees. *IEEE Trans Biomed Eng* 60(10):2745–2750
23. Naik GR, Easter Selvan S, Arjunan SP, Acharyya A, Kumar DK, Ramanujam A, Nguyen HT (2018) An ICA-EBM-based sEMG classifier for recognizing lower limb movements in individuals with and without knee pathology. *IEEE Trans Neural Syst Rehabil Eng* 26(3):675–686
24. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
25. LeCun Y, Bengio Y et al (1995) Convolutional networks for images, speech, and time series. *Handbook Brain Theory Neural Netw* 3361(10):1995
26. Ullah I, Hussain M, Aboalsamh H et al (2018) An automated system for epilepsy detection using EEG brain signals based on deep learning approach. *Expert Syst Appl* 107:61–71
27. Gautam A, Panwar M, Biswas D, Acharyya A (2020) Myonet: a transfer-learning-based IRCN for lower limb movement recognition and knee joint angle prediction for remote monitoring of rehabilitation progress from sEMG. *IEEE J Transl Eng Health Med* 8:1–10
28. Sanchez OFA, Sotelo JLR, Gonzales MH, Hernandez GAM (2014) EMG dataset in lower limb data set. *UCI Mach Learn Repos* pp 2014–02
29. Vijayvargiya A, Kumar R, Dey N, Tavares JMR. (2020) Comparative analysis of machine learning techniques for the classification of knee abnormality. pp 1–6
30. Chowdhury R, Reaz M, Ali M, Bakar A, Chellappan K, Chang T (2013) Surface electromyography signal processing and classification techniques. *Sensors* 13(9):12431–12466
31. Phinyomark A, Phukpattaranont P, Limsakul C (2011) Wavelet-based denoising algorithm for robust EMG pattern recognition. *Fluct Noise Lett* 10(02):157–167
32. Graps A (1995) An introduction to wavelets. *IEEE Comput Sci Eng* 2(2):50–61
33. Jing-Yi L, Hong L, Dong Y, Yan-Sheng Z (2016) A new wavelet threshold function and denoising application. *Math Prob Eng*
34. Phinyomark A, Limsakul C, Phukpattaranont P (2011) Application of wavelet analysis in EMG feature extraction for pattern classification. *Meas Sci Rev* 11(2):45–52
35. Oskoei MA, Hu H (2008) Support vector machine-based classification scheme for myoelectric control applied to upper limb. *IEEE Trans Biomed Eng* 55(8):1956–1965
36. Kiranyaz S, Ince T, Gabbouj M (2015) Real-time patient-specific ECG classification by 1-d convolutional neural networks. *IEEE Trans Biomed Eng* 63(3):664–675
37. Ullah I, Petrosino A (2016) About pyramid structure in convolutional neural networks. pp 1318–1324
38. Kingma DP, Adam JB (2014) A method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
39. Mehta S, Paunwala C, Vaidya B (2019) CNN based traffic sign classification using Adam optimizer. pp 1293–1298
40. Vijayvargiya A, Gupta V, Kumar R, Dey N, Tavares JMR (2021) A hybrid WD-EEMD sEMG feature extraction technique for lower limb activity recognition. *IEEE Sensors J* 21:20431–20439
41. Herrera-González M, Martínez-Hernández GA, Rodríguez-Sotelo JL, Avilés-Sánchez OF (2015) Knee functional state classification using surface electromyographic and goniometric signals by means of artificial neural networks. *Ing Univ* 19(1):51–66
42. Zhang Y, Peng X, Li P, Duan K, Wen Y, Yang Q, Zhang T, Yao D (2017) Noise-assisted multivariate empirical mode decomposition for multichannel EMG signals. *Biomed Eng Online* 16(1):107