Machine and Deep Learning Model for EMG Signal Classification: A New Performance-Cost Analysis Across CPU and GPU Architectures

Toka Fathi
Department of Computer and Information Engineering, Ninevah University
Iraq
toka.fathi@uoninevah.edu.iq

Mohammed A M Abdullah
Department of Computer and Information Engineering, Ninevah University
Iraq
mohammed.abdulmuttaleb@uoninevah.edu.iq

Basil Shukr
Department of Computer Engineering, University of Mosul
Iraq
basil.mahmood@uomosul.edu.iq

Abstract: *Electromyographic (EMG) signal classification paves the road for many human-machine interface applications where EMG sensors capture muscle activity for further processing and application. In this work, the performance of several machine learning algorithms is tested, including Decision Trees, Random Forests (RF), K-Nearest Neighbors (KNN), Extreme Gradient Boosting (XGBoost), Long Short-Term Memory (LSTM), and Convolutional Neural Network-Long Short Term Memory (CNN-LSTM). These models are evaluated and compared using EMG data collected from approximately 30 subjects performing six distinct gestures. In addition, a hybrid CNN-LSTM model is proposed to achieve an accurate yet low-cost EMG classification. Moreover, the performance of these algorithms is compared on both Central Processing Unit (CPU) and Graphics Processing Unit (GPU) platforms in terms of accuracy, training/testing time, and hardware utilization. Results show that the RF algorithm gives the best performance with an accuracy of 98.16%. On the other hand, the decision tree algorithm gives a trade-off between the accuracy and computational efficiency, with 95.46% accuracy, and approximately 17.6 times faster compared to RF, making it more suitable for real-time applications and limited resources environments. On the other hand, deep learning models acquired noticeably higher computational time compared to classical algorithms. This research demonstrates the importance of selecting a suitable classification algorithm and hardware platform to achieve an efficient EMG classification.*

Keywords: *EMG, GPU, decision trees, random forests, KNN, XGboost, LSTM, and CNN-LSTM.*

Received May 6, 2025; accepted October 6, 2025
https://doi.org/10.34028/iajit/23/2/11

1. Introduction

Universally, the number of amputations has increased significantly, with cases rising from 370.25 million in 1990 to 552.45 million in 2019, reflecting a growth of 67% [32]. This increment has driven a growing demand for advanced prosthetic devices able of restoring lost functionality, resulting in utilizing Electromyographic (EMG) signals in prosthetic control and rehabilitation. Bioelectrical signals, particularly EMG signals which result from muscle activity is controlled by neural impulses. EMG sensors detect these signals and capture essential information relating to muscle states and movements. This enabled EMG signals to have broad applications, ranging from medical diagnostics and rehabilitation to assistive technologies such as gesture recognition [16].

Accurate classification of EMG signals plays a crucial role in enhancing the aforementioned applications as the ability to distinguish between different muscle activities contributes to the development of advanced Human-Computer Interaction (HCI) systems. EMG sensors can be broadly divided into two types: the first one is the Surface Electromyographic (sEMG) and the second is the needle Intramuscular Electromyographic (iEMG) [15]. sEMG is a non-invasive technique that uses electrodes applied directly to the skin to measure muscle activity, this makes it widely used for rehabilitation and gesture recognition systems. sEMG signals have been broadly used in many man-machine interfaces [15]. On the other hand, iEMG need to be inserted into the muscle tissues directly which make it able to record deep muscles activities. Although iEMG sensors generate accurate and pure signal, they are invasive and not suitable for long term usage. On the contrary, sEMG sensors are non-invasive and more suitable for long-term usage, as it allows for the detection of muscle activity without penetrating the skin. However, the disadvantage of using sEMG sensors is the reduced accuracy due to skin impedance and cross-talk from the adjacent muscles' signals [26].

The development of a robotic limb that could be controlled via muscle contractions in late of 1950s indicated the starting point of myoelectric control systems [6]. Soon enough Eddy *et al*. [3] noticed that EMG signals might be used to operate prosthetic devices after they are collected from an amputee's residual limb. However, traditional EMG classification methods, such as threshold-based or rule-based techniques, often struggle with accuracy due to the complexity of EMG signals as these signals are noisy, influenced by factors such as electrode placement and muscle strength. Also, traditional classification techniques are sensitive to noise and hardly capture complex patterns. To address these limitations, Artificial Intelligence (AI)-based classification methods have emerged as a robust alternative. Deep learning and machine learning algorithms, particularly Convolutional Neural Network (CNN) and decision tree have demonstrated a good performance in recognizing EMG signal patterns [25].

In this context, researchers have explored many deep and machine learning algorithms to improve the accuracy and robustness of EMG signal classification. For example, Ozdemir *et al*. [23] recorded EMG signals from 30 participants while performing 7 different hand gestures namely: extension, flexion, open hand, punch, radial deviation, rest, and ulnar deviation. The data were collected from 4-channels and converted into spectrogram images using Short-Time Fourier Transform (STFT) which are passed to A 50-layer CNN based on Residual Networks (ResNet) architecture. The method achieved a test accuracy of 99.59% and the authors reported that deep learning outperformed conventional techniques in terms of prediction speed and accuracy. Wahid *et al*. [29] compare the effectiveness of five machine learning algorithms namely: Naïve Bayes, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Discriminant Analysis, and Random Forest (RF) for EMG classification. It was found that the RF algorithm gave the highest classification accuracy (96.38%) when using EMG features normalized to RMS value.

Waris *et al*. [30] examined the longitudinal performance of several techniques for classifying hand movements based on EMG. Over the period of seven days, ten normal people and twelve amputees participated in the study. The classification methods included KNN, SVM, Artificial Neural Network (ANN), Linear Discriminant Analysis (LDA), and decision trees. an intelligent framework has been developed by Khan *et al*. [11] to classify four EMG hand gestures; the EMG data is collected using the Myo armband then classified using SVM. The cubic-support vector machine was trained on four diverse hand gestures. The supervised support vector machine model reached a cumulative classification accuracy of 98.9%. A hybrid architecture consisting of CNN and Bidirectional Long Short-Term Memory (Bi-LSTM) is used by Karnam *et al*. [10] for the classification of hand activities. The obtained results showed that using two layers LSTM and CNN gives better results compared to Recurrent Neural Network (RRN) for EMG classification. Jabbari *et al*. [9] employed an LSTM-based Neural Network (NN) to recognize EMG patterns. The proposed model consists of multilayer LSTM with a concatenated SoftMax layer. The LSTM layer learns the long-term dependency and the classification is performed by the SoftMax layer.

Researchers have examined the implementation of machine and deep learning algorithms for EMG classification of different hardware. For example, Senagi and Jouandeau [28] compared the parallel and sequential implementation of RF algorithm on Central Processing Unit (CPU) and Graphics Processing Unit (GPU). The results showed an improved average speedup of 1.62 is achieved using parallelized version of RF on GPUs and an average speed up of 3.57 compared to the CPU implementation using dynamic parallelism RF on GPU. Wen *et al*. [31], demonstrated that GPU implementations of RF and gradient boosted decision trees achieve significant speedups compared to their CPU implementation, mainly for large datasets. The Extreme Gradient Boosting (XGBoost) showed substantially reduced training time on the GPU compared to the CPU, underscoring the efficiency of parallel processing for these algorithms. Despite these advances, only a few studies have systematically compared a broad range of EMG classification algorithms under similar conditions-especially considering both accuracy and computational efficiency (e.g., real-time constraints or hardware differences) [1]. Similarly, few works take into account the deployment of such AI algorithms on different hardware platforms for EMG signal classification [18], highlighting a gap in the literature.

A critical factor in selecting the most appropriate algorithm for EMG data classification is the cost, which is related to both computational time and hardware requirements. In this work, various machine learning and deep learning algorithms were evaluated for classifying EMG data corresponding to seven gestures. To determine the most suitable approach, these algorithms were implemented using both CPU and GPU, and a combination of them. This work investigates performance evaluation based on key performance of several machine learning and deep learning methods metrics such as number of parameters and training parameters, as well as training and testing time, to identify the most efficient classification method.

The following research questions are addressed in the study:

- Which machine learning and deep learning algorithms gives the highest classification accuracy for the given sEMG dataset?
- What is the comparative performance, in terms of training and testing durations, of these algorithms when executed on CPU versus GPU architectures?
- Which classification strategy yeild the best trade-off between classification accuracy and computational cost.

The paper is organized as follows. The next section presents employed machine learning models used for

EMG classification in addition to the proposed models. Section 3 discusses the results and compares the accuracy and computational efficiency of various models across CPU and GPU. Finaly section 4 concludes this work.

2. Methodology

2.1. Machine Learning Models

Machine learning strategies are widely used for classifying EMG signals. In literature, several machine learning techniques has been used to classify the EMG signal. The next subsection gives brief overview of the employed methods.

2.1.1. Decision Trees

One popular technique for classifying data is Decision Tree Classifier (DTC). It is one of the famous supervised data mining tools. The most important aspect of DTC is its capability to simplify complex decision-making issues into steps that lead to a more understandable and explainable result [18].

The DTC creates categorization models in the shape of trees, and generated tree can be used as a decision-making tool. There are two steps to DT generation [19]: the first step involves tree generation where every data tuple is at the root node. After applying a splitting criterion, the optimal splitting attribute for the following level is chosen, Figure 1 shows a boolean function example and decision tree representing it. For the node, the number of branches is determined by the value of the best-splitting property. When a full tree is produced, the accuracy of the training data cannot be increased by adding more leaf nodes. Tree pruning is performed in the second step. This is performed by eliminating the sub-tree that reflects noise and outliers, to reduce the size of the tree [19].

Classification trees and regression trees are decision trees that forecast numerical or nominal variables, respectively. In the context of machine learning, one crucial characteristic of decision trees is that the prediction is the outcome of a straightforward and understandable calculation [20].

X1	X_1	X_3	X	Y
True	True	True	True	True
True	True	True	True	False
False	True	False	True	False
True	True	True	True	False
False	True	False	True	False
False	True	True	False	False
False	True	False	False	False

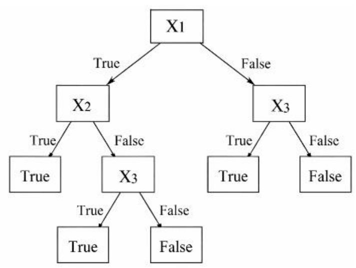

Figure 1. The boolean function Y=X1∧X2∨X3, and decision tree representing it.

2.1.2. Random Forest (RF)

One machine-learning technique that may be applied to both regression and classification problems is the RF model. The decision tree structure and the RF regression model's operational logic are similar. Many decision trees are produced by dividing the data set randomly by continually repeating the learning process and combining the predictions of the several learnt trees to create a meta-model, as illustrated in Figure 2; Ensemble techniques reduce the impact of random artifacts in the training set or learning procedure [21].

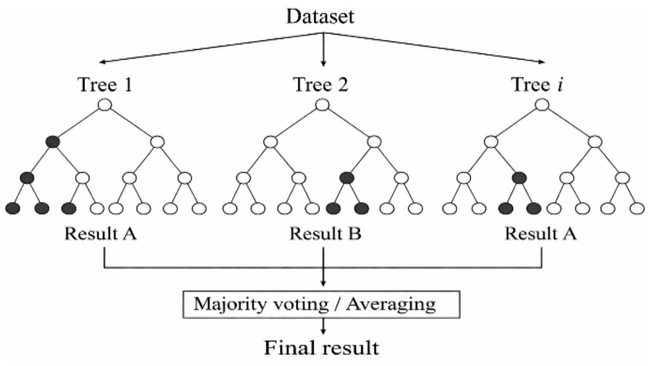

Figure 2. RF classifier uses majority voting of the predictions made by randomly created decision trees to make the final prediction [7].

2.1.3. K-Nearest Neighbor Classifiers (KNN)

One non-parametric supervised learning approach is KNN. An individual or new data point's categorization is based on the distance of the point to a certain number of neighbors, as shown in Figure 3. Euclidean distance and negative cosine similarity are two common distance functions utilized in KNN to compute the distance [12].

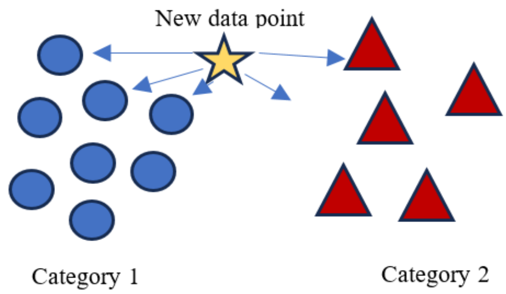

Figure 3. Visualization of KNN classes.

2.1.4. Explainable Extreme Gradient Boosting (XGBOOST)

This machine learning algorithm combines gradient boosting and regularization techniques. It is based on the decision tree method and implemented to enhance the performance of different data mining tasks. Figure 4 depicts the block diagram of XGBOOST. The algorithm aims to determine the connection between an output (Y) and an input (X={x1, x2, ⋯, xn}) about certain samples. The main concept is to integrate several weak learners to create a strong learner. The Classification and Regression Trees (CART) create a tree model for each weak learner. The XGBoost algorithm continuously splits feature to train a new function that suits the residual of the prior prediction [33].

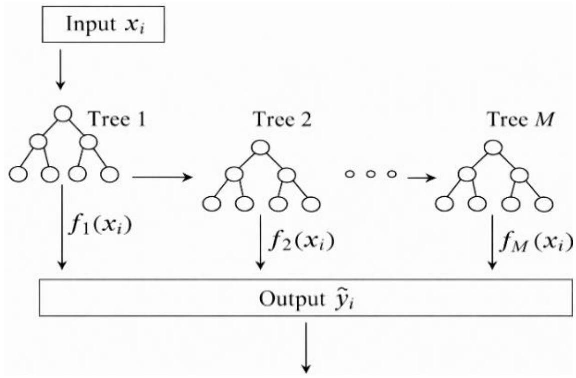

Figure 4. The schematic diagram of the XGBoost algorithm [20].

2.2. Deep Learning Models

Deep learning model are also widely used in the classification of the EMG signals; in this work two deep learning models are employed: LSTM model and CNN-LSTM model.

2.2.1. Long Short-Term Memory (LSTM)

A Feed Forward Neural Network (FFNN) with loops in the hidden layers is extended into a Recurrent Neural Network (RNN). This enables the model to identify temporal relationships between data by using a sequence of samples as input. Nevertheless, it has been shown that individual nodes struggle to establish long-term relationships. The LSTM network solves the problem by giving the hidden node loops extra parameters in order to get states depending on the input data [17]. Figure 5 shows the structure of a single LSTM unit [22].

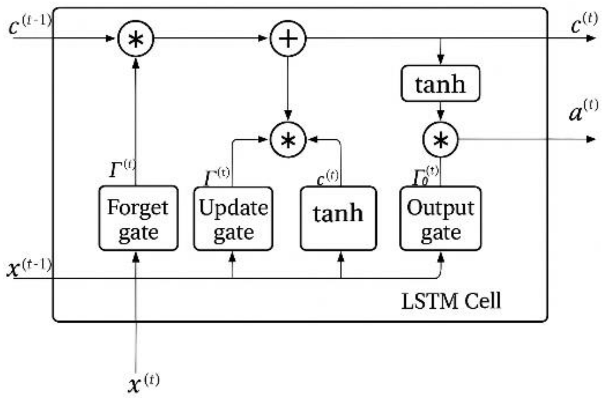

Figure 5. LSTM cell.

2.2.2. CNN-LSTM

CNNs are used extensively in artificial intelligence applications such as computer vision, image processing, and natural language processing, among many others. The main building blocks of the CNN are the pooling and convolutional layers. Each channel in a convolutional layer has a set of parameters called a convolution kernel that links to a small portion of the prior layer's fixed-size input data. CNNs constantly contain some convolutional layers and pooling layers; subsequently, fully connected layers are added [8].

2.3. Proposed Method

CNNs combined with recurrent networks, such as LSTMs, have shown improved performance by capturing both spatial features (across EMG channels) and temporal dynamics in the signal [19]. A hybrid CNN-LSTM model can support this interaction: CNN layers act as feature extractors from raw EMG signals, while LSTM layers model the time dependencies, which is effective for data that involves time dependency and spatial relationships. So, to better capture complex EMG patterns, we propose a hybrid deep learning model that combines both LSTM and a CNN for accurate EMG signal classification.

The proposed model has two parts: CNN and LSTM. In the first part, CNN is used for extracting spatial feature of raw EMG signals by identifying local patterns and time-based dependencies within short segments. In the second part, LSTM accomplishes the modeling of long-term temporal dependencies in the EMG. The proposed CNN-LSTM model is carefully designed to achieve a compromise between computational cost, generalization, and accuracy. This use of the hybrid architecture improves classification performance on multi-class EMG gesture datasets by utilizing the advantages of both CNNs and LSTMs.

Table 1. The hybrid CNN-LSTM model architecture.

Layer	Details
Input layer	EMG data input
1st convolutional layer	64 filters, ReLU activation, kernel size=3, padding=1
MaxPooling layer	Kernel size=2
2nd convolutional layer	128 filters, ReLU activation, kernel size=3, padding=1
MaxPooling layer	Kernel
LSTM layer	LSTM with 128 hidden nodes
Dropout layer	Dropout rate=0.4
Fully connected layer 1	**128 neurons, ReLU activation**
Dropout layer	Dropout rate=**0.4**
Output layer	Fully connected, number of classes (7)

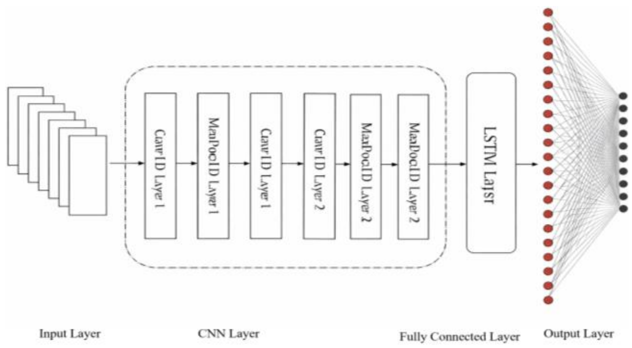

Figure 6. CNN LSTM model architecture.

The architecture of the proposed hybrid CNN-LSTM model is illustrated in Figure 6. The proposed model begins with two convolutional layers responsible for high-level features extraction from the input data. These two layers help in learning local temporal patterns from the EMG signal segments. Each convolutional layer is followed by Max-pooling layers to reduce dimensionality and retain dominant features. The output from the CNN section is reshaped and passed to an LSTM layer consisting of 128 hidden units, which captures sequential dependencies and long-term patterns in the data. The dropout layer helps prevent overfitting and improve generalization, which makes the model more robust when applied to new data. Finally, a dense (fully connected) layer with ReLU activation is applied, followed by a dropout layer to prevent overfitting. The final layer is a SoftMax output layer that classifies the signal into one of the seven predefined gesture classes. The layers of the proposed hybrid model are shown in Table 1. The pseudo code of the CNN-LSTM model training is shown on Algorithm (1). Figure 7 shows the training process of the CNN-LSTM model for EMG classification.

Algorithm 1: CNN-LSTM model training for EMG signal classification.

Input: Raw EMG dataset with 8-channel signals and class labels
Output: Trained CNN-LSTM model for gesture classification

1. *Load EMG data from file*
2. *Normalize features using StandardScaler*
3. *Split data into train/val/test sets (70%,15%,15%)*
4. *Reshape data into [samples, 8, 1] format*
5. *Wrap data in PyTorch Dataset and DataLoader*
6. *Define CNN-LSTM model with:*
   -Conv1D→ReLU→MaxPool
   -Conv1D→ReLU→MaxPool
   -LSTM (hidden_size=128, num_layers=3)
   -Fully connected→Dropout→Softmax
7. *Initialize model, optimizer (Adam), and loss function*
8. *For each epoch from 1 to N:*
   a. Train model on batches:
-Forward pass
      -Compute loss
      -Backpropagate
      -Update weights
   b. Evaluate on validation set:
      -Compute accuracy and F1-score
   c. Save model if validation accuracy improves
9. Load best model
10. Evaluate on test set: report Accuracy, F1, Precision, Recall

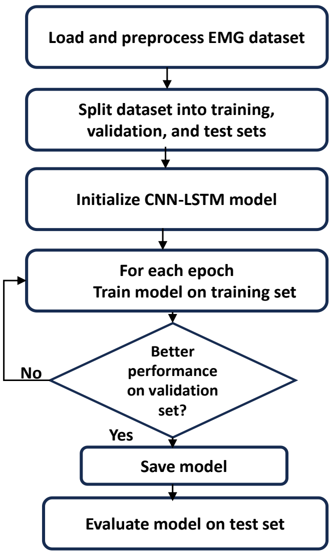

Figure 7. Training process of the CNN-LSTM model for EMG classification.

2.4. Performance Metrics

Performance metrics are crucial to assess the model's performance after it has been trained. A model's performance may be measured using different metrics, the majority of which are based on the confusion matrix, which includes variables such as True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) [22].

2.4.1. Confusion Matrix

A confusion matrix is a tabular representation that compares the real and predicted class labels. The common representation is to display occurrences of a real class in each row and a predicted class in each column. Statistics on actual and anticipated data classifications are included in confusion matrix. These metrics are produced by the classifier algorithms. The matrix's data is used to evaluate the systems' performance [14].

2.4.2. Accuracy

The accuracy of the classification gives good idea about the model performance which is based on TP, TN, FP and FN and is shown in Equation (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

where the *FP* is false positive, *TP* is true positive, *FN* is false negative and *TN* is true negative [21].

2.4.3. F-Measure (F1-Score)

F-score is the measure of the predictive performance. It represents the balanced performance of the precision and recall. F1-score is given by:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (2)$$

3. Experiments and Results

The machine learning and deep learning models are executed on a personal computer with Intel Core (TM) i7-9850H CPU, 2.60GHz with 32.0 GB RAM. In order to accelerate the performance NVIDIA Tesla T4 GPU, provided via Kaggle is used. Algorithms that need GPU were implemented in a Kaggle cloud-based environment to reduce the training time. The Scikit-learn and PyTorch Python libraries are also used to build the models and generate performance metrics.

3.1. Dataset

The dataset is obtained from UC Irvine machine learning repository. This dataset is collected by N. Krilova *et al*. [13] and licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license. To collect EMG signals, the users were asked to wear the MYO Thalmic bracelet, shown in Figure 8, on their forearm, while a personal computer is used for signal processing and connected to the bracelet via Bluetooth. The bracelet contains eight sensors which are equally spread around the forearm.

Figure 8. Thalmic MYO armband.

Thirty-six subjects' raw EMG data were obtained while performing a sequence of static hand gestures. Each subject makes two sequences, each of which involves seven basic gestures. There was a gap of three seconds between each gesture, and each gesture was executed for a duration of three seconds.

At a rate of 200 samples per second, these signals are sampled with eight-bit precision. The bracelet represents the amplified value acquired from each of the EMG sensors by encoding the potential produced by muscular action to integer numbers with a range of 0 to 1023 [5]. The data set contains a total of 40,000 to 50,000 recordings in each column, with 30,000 of those recordings being guaranteed. The EMG data was recorded while the subjects perform six basic tasks as shown in Figure 9, the first class represents unmarked data unmarked data, class 1 for the hand at rest, class 2 for a clenched fist, class 3 for wrist flexion, class 4 for wrist extension, class 5 for radial deviation, class 6 for ulnar deviation, and finally class 7 for an extended palm (which was not performed by all subjects). This dataset allows extensive examinations of a wide range of hand and wrist actions, making it an excellent resource for gesture classification.

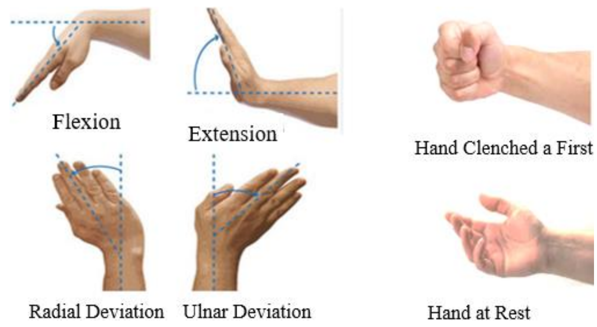

Figure 9. The six basic tasks.

3.2. Results

Four machine learning models are implemented, namely: the decision tree, the RF, the K-NN, the XGBoost model, and two deep learning models are also built in addition to the proposed model based on, the LSTM and the CNN-LSTM model. A comparison Between among these models is given in Table 2.

Table 2. Performance metrics and complexity of machine learning and deep learning models.

Model	Accuracy %	Training time (CPU) seconds	Training time (GPU) seconds	Testing time seconds	F1-score	Trainable params
Decision tree	95.46	46.64	0.76	1.31	0.9285	2.42×10^5
RF	98.16	1296.10 515.23 (parallel)	353.42	23.09	0.9751	354.01×10^5
kNN	97.84	11.93	9.18	365.46	0.9536	33.90×10^5
XGBOOST	97.6	405.65	150.72	152.85	0.87	0.32×10^5
LSTM	83.80	-	19853.81	16.8059	0.6925	3.52×10^5
CNN-LSTM	92.01	-	25183.8278	20.28	0.8824	4.39×10^5

As shown in Table 2, the highest classification accuracy was obtained using the RF classifier. Additionally, Table 2 indicates that the decision tree algorithm achieved the shortest training and testing time. This efficiency can be attributed to the algorithm's structure, where data follows in a single path from the root node to a terminal leaf, minimizing computational complexity. Despite the high number of training parameters, only a single path of the tree is chosen, which reduces the computation overhead.

The RF algorithm contains multiple decision trees, and combining their outputs determines the final classification. Although this approach requires higher computational resources due to the increased number of parameters, it achieved the best performance. The parallel implementation of this algorithm enhances efficiency. When executed on a GPU, the training time is significantly reduced, making the algorithm well-suited for GPU-based computation. A paired t-test was performed on the accuracy scores obtained from 5-fold cross-validation to statistically compare the performance of the decision tree and RF classifiers. The test yielded a t-statistic of 67.00 and a p-value less than 0.0001, confirming that the superior performance of the RF model.

Similarly, the XGBoost algorithm utilizes gradient boosting to optimize training by reducing the number of parameters, thereby decreasing both training and testing times.

The low training time of the KNN algorithm is attributed to its non-parametric nature, as it stores the dataset without performing a learning process during the training phase. However, this results in a higher testing time since classification requires computing distances between the query point and all stored data points. Despite this limitation, KNN has the advantage of being simple and easy to implement. Figures 10 and 11 show the confusion matrix of the DT and RF, respectively. The learning curve of the DT model is shown in Figure 10. On the other hand, the two deep learning models employed in this study, LSTM and CNN-LSTM, demonstrate notably extended training durations when compared with conventional machine learning models. This is due to the fact that CNN architectures incorporate a greater number of parameters with a higher number of computational steps for each parameter Figure 11.

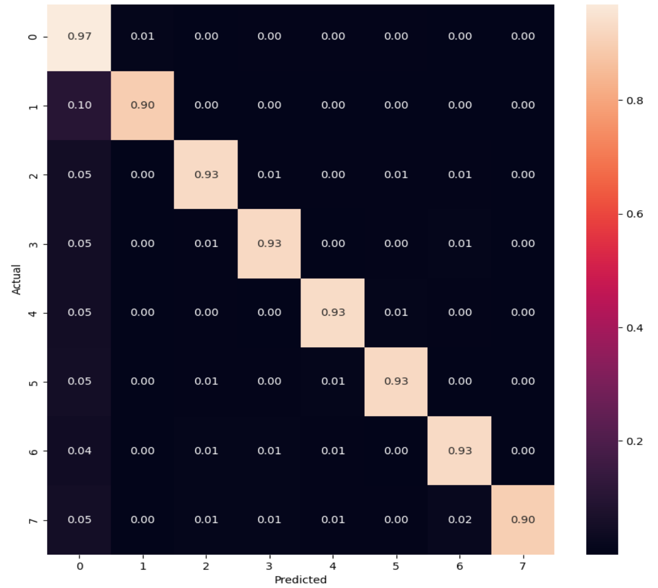

Figure 10. The confusion matrix for the decision tree model.

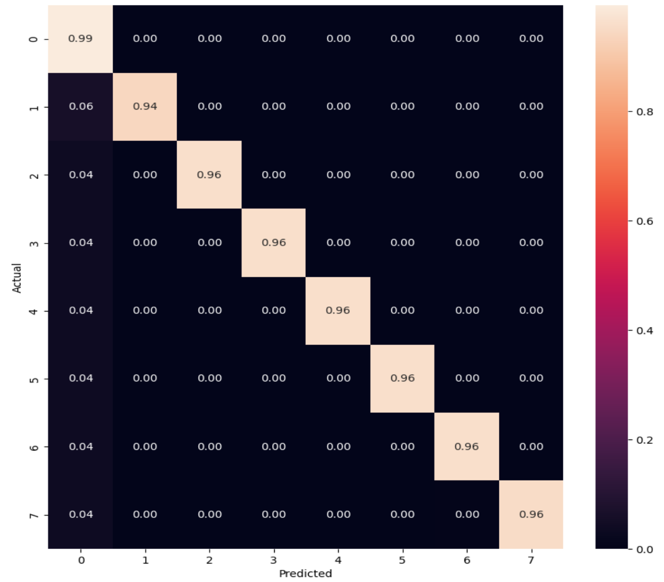

Figure 11. The confusion matrix for the RF model.

Figure 12 presents the learning curve of the RF classifier, when plotting training accuracy, testing accuracy, and the train-test gap as a function of the training set size. Both training and testing accuracies are consistently high (close to 1.0), indicating that the model generalizes well to unseen data.

The findings indicate that selecting the best model depends on the application of EMG. In the context of real-time systems, decision trees are favored because of their minimal computational cost and rapid testing time. In scenarios where accuracy is important, ensemble methods and deep learning techniques are more appropriate, although they require longer training and testing durations.

In terms of hardware setup, the use of the GPU reduces the computation time due to the parallel processing. This is true for complex models such as CNN and NN. On the other hand, for lightweight models, such as DTs, KNN, and RF, CPU can be utilized to save power and cost. Figure 13 shows a comparison among different models' training and testing time. Figure 13 shows the training and testing time using CPU and GPU for different classifiers. Figure 14 demonstrates the RF training and testing accuracies, when plotting training accuracy, testing accuracy, and the train-test gap as a function of the training set size. Both training and testing accuracies are consistently high (close to 1.0), indicating that the model generalizes well to unseen data.

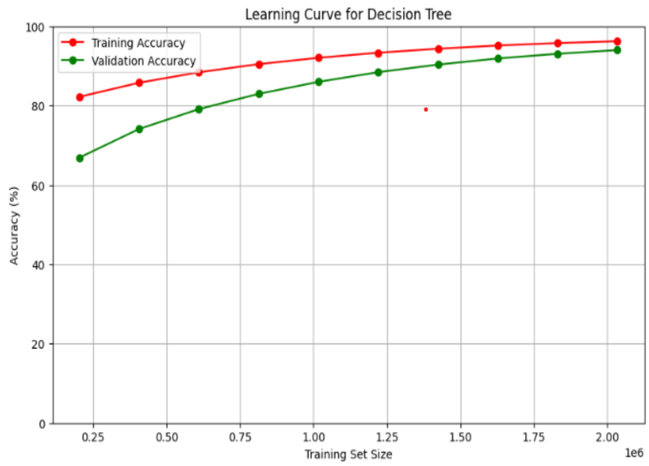

Figure 12. Decision tree learning curve.

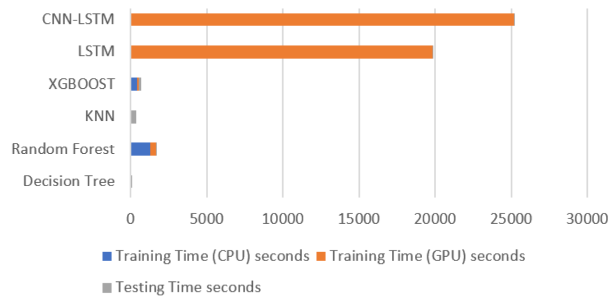

Figure 13. Training and testing time using CPU and GPU.

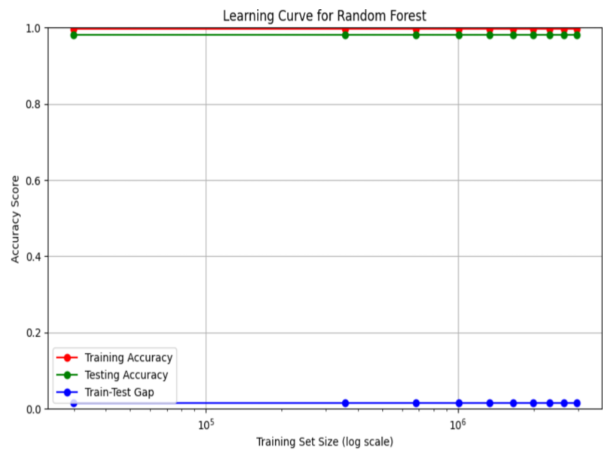

Figure 14. RF training and testing accuracy.

4. Conclusions

This study explored the use of various machine and deep learning algorithms to classify EMG signals. The study compares the algorithms with respect to their accuracy, computational cost, training, and testing times on two platforms, CPU and GPU. The results show the necessity to trade-off between EMG classification accuracy and resource cost and give guidelines to choose the best classification algorithm and platform depending on the application needed.

The RF algorithm gave the best classification accuracy of 98.16% but consumed the maximum number of training nodes, while the decision tree algorithm is simpler and needed less training time, testing time, and training nodes, and gave good accuracy of 95.46%. Deep learning models such as LSTM and CNN-LSTM need iterative optimization processes and sequential computations and require much more time compared to decision tree-based algorithms, so they perform well on the GPU platform, where the training time is reduced significantly.

Results show that the choice of the best algorithm depends on the application of EMG classification. For real-time applications, the decision trees are more suitable. On the other hand, the RF and deep learning models are more suitable in offline applications that require high accuracy. This research contributes to the understanding of choosing the best hardware for machine learning in the field of biomedical and EMG classification.

5. Limitations and Future Work

The study is applied on a single available dataset with fixed acquisition protocol and fixed acquisition device. The performance of the classification system may vary with real-world data from different acquisition devices and different noise levels. The machine and deep learning models were trained and validated on non-streaming data, which may not reflect the true performance in dynamic or wearable settings.

For future work, it is recommended to deploy the models on embedded platforms such as Jetson Nano, Raspberry Pi, and FPGAs for real-time testing. The dataset could be expanded with data collected from diverse subjects and devices. Power consumption and latency to validate real-time feasibility of different models could be examined.

References

[1] Abbaspour S., Naber A., Catalan M., GholamHosseini H., and Linden M., "Real-Time and Offline Evaluation of Myoelectric Pattern Recognition for the Decoding of Hand Movements," *Sensors*, vol. 21, no. 16, pp. 1-17, 2021. https://doi.org/10.3390/s21165677

[2] Blockeel H., Devos L., Frenay B., Nanfack G., and Nijssen S., "Decision Trees: from Efficient Prediction to Responsible AI," *Frontiers in Artificial Intelligence*, vol. 6, pp. 1-17, 2023. https://doi.org/10.3389/frai.2023.1124553

[3] Eddy E., Campbell E., Bateman S., and Scheme E., "Big Data in Myoelectric Control: Large Multi-User Models Enable Robust Zero-Shot EMG-based Discrete Gesture Recognition," *Frontiers in Bioengineering and Biotechnology*, vol. 12, pp. 1-24, 2024. https://doi.org/10.3389/fbioe.2024.1463377

[4] Ersin C. and Yaz M., "Comparison of KNN and Random Forest Algorithms in Classifying EMG Signal," *Avrupa Bilim ve Teknoloji Dergisi*, vol. 51, pp. 209-216, 2023. https://doi.org/10.31590/ejosat.1285176

[5] Espinoza D. and Velasco L., "Comparison of EMG Signal Classification Algorithms for the Control of an Upper Limb Prosthesis Prototype," *in Proceedings of the 17th International Conference Electrical Engineering, Computing Science and Automatic Control*, Mexico City, pp. 1-4, 2020. https://doi.org/10.1109/CCE50788.2020.9299208

[6] Fleming A., Stafford N., Huang S., Hu X., and et al., "Myoelectric Control of Robotic Lower Limb Prostheses: A Review of Electromyography Interfaces, Control Paradigms, Challenges and Future Directions," *Journal of Neural Engineering*, vol. 18, no. 4, pp. 1741-2552, 2021. https://doi.org/10.1088/1741-2552/ac1176

[7] Ghosh S., "Comparing Regular Random Forest Model with Weighted Random Forest Model for Classification Problem," *International Journal of Statistics and Applications*, vol. 14, no. 1, pp. 7-12, 2024. DOI:10.5923/j.statistics.20241401.02

[8] Huang D. and Chen B., "Surface EMG Decoding for Hand Gestures Based on Spectrogram and CNN-LSTM," *in Proceedings of the 2nd China Symposium on Cognitive Computing and Hybrid Intelligence*, Xi'an, pp. 123-126, 2019. https://doi.org/10.1109/CCHI.2019.8901936

[9] Jabbari M., Khushaba R., and Nazarpour K., "EMG-Based Hand Gesture Classification with Long Short-Term Memory Deep Recurrent Neural Networks," *in Proceedings of the 42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Montreal, pp. 3302-3305, 2020. https://doi.org/10.1109/EMBC44109.2020.9175279

[10] Karnam N., Dubey S., Turlapaty A., and Gokaraju B., "EMGHandNet: A Hybrid CNN and Bi-LSTM Architecture for Hand Activity Classification Using Surface EMG Signals," *Biocybernetics and Biomedical Engineering*, vol. 42, pp. 325-340, 2022. https://doi.org/10.1016/j.bbe.2022.02.005

[11] Khan M., Khan H., Muneeb M., Abbasi Z., and et al., "Supervised Machine Learning-based Fast

Hand Gesture Recognition and Classification Using Electromyography (EMG) Signals," *in Proceedings of the International Conference on Applied and Engineering Mathematics*, Taxila, pp. 1-6, 2021. https://doi.org/10.1109/ICAEM53552.2021.9547148

[12] Kok C., Ho C., Tan F., and Koh Y., "Machine Learning-based Feature Extraction and Classification of EMG Signals for Intuitive Prosthetic Control," *Applied Sciences*, vol. 14, no. 13, pp. 1-29, 2024. https://doi.org/10.3390/app14135784

[13] Krilova N., Kastalskiy I., Kazantsev V., Makarov V., and Lobov S., *EMG Data for Gestures*, UCI Machine Learning Repository, https://doi.org/10.24432/C5ZP5C, Last Visited, 2025.

[14] Krstinic D., Braovic M., Seric L., and Stulic D., "Multi-Label Classifier Performance Evaluation with Confusion Matrix," *Computer Science and Information Technology*, vol. 10, no. 8, pp. 1-14, 2020. https://csitcp.org/abstract/10/108csit01

[15] Kundu B. and Naidu D., "Classification and Feature Extraction of Different Hand Movements from EMG Signal Using Machine Learning-based Algorithms," *in Proceedings of the International Conference on Electrical, Communication, and Computer Engineering*, Kuala Lumpur, pp. 1-66, 2021. https://doi.org/10.1109/ICECCE52056.2021.9514134

[16] Lee K., Min J., and Byun S., "Electromyogram-based Classification of Hand and Finger Gestures Using Artificial Neural Networks," *Sensors*, vol. 22, no. 1, pp. 1-20, 2021. https://doi.org/10.3390/s22010225

[17] Liu Y., Breceda A., Karoly P., Grayden D., and et al., "Brain Model State Space Reconstruction Using an LSTM Neural Network," *Journal of Neural Engineering*, vol. 20, no. 3, pp. 1-14, 2023. DOI: 10.1088/1741-2552/acd871

[18] Lopez J., Aviles M., Perez D., Socarras I., and Resendiz J., "Electromyography Signals in Embedded Systems: A Review of Processing and Classification Techniques," *Biomimetics*, vol. 10, no. 3, pp. 1-25, 2025. https://doi.org/10.3390/biomimetics10030166

[19] Lopez L., Ferri F., Zea J., Caraguay A., and Benalcazar M., "CNN-LSTM and Post-Processing for EMG-based Hand Gesture Recognition," *Intelligent Systems with Applications*, vol. 22, pp. 1-12, 2024. https://doi.org/10.1016/j.iswa.2024.200352

[20] Maged A., Elshaboury N., and Akanbi L., "Data-Driven Prediction of Construction and Demolition Waste Generation Using Limited Datasets in Developing Countries: An Optimized Extreme Gradient Boosting Approach," *Environment, Development and Sustainability*, vol. 27, pp. 26865-26889, 2024. https://doi.org/10.1007/s10668-024-04814-z

[21] Omar S., Kimwele M., Olowolayemo A., and Kaburu D., "Enhancing EEG Signals Classification Using LSTM-CNN Architecture," *Engineering Reports*, vol. 6, no. 9, pp. 1-23, 2023. https://doi.org/10.1002/eng2.12827

[22] Ossaba A., Tigreros J., Tejada J., Pena A., and et al., "LSTM Recurrent Neural Network for Hand Gesture Recognition Using EMG Signals," *Applied Sciences*, vol. 12, no. 19, pp. 1-21, 2022. https://www.mdpi.com/2076-3417/12/19/9700#

[23] Ozdemir M., Kisa D., Guren O., and Akan A., "Hand Gesture Classification Using Time-Frequency Images and Transfer Learning Based on CNN," *Biomedical Signal Processing and Control*, vol. 77, pp. 103787, 2022. https://doi.org/10.1016/j.bspc.2022.103787

[24] Priyanka. and Kumar D., "Decision Tree Classifier: A Detailed Survey," *International Journal of Information and Decision Sciences*, vol. 12, no. 3, pp. 246-269, 2020. https://doi.org/10.1504/IJIDS.2020.108141

[25] Rajapriya R., Rajeswari K., and Thiruvengadam S., "Deep Learning and Machine Learning Techniques to Improve Hand Movement Classification in Myoelectric Control System," *Biocybernetics and Biomedical Engineering*, vol. 41, no. 2, pp. 554-571, 2021. https://doi.org/10.1016/j.bbe.2021.03.006

[26] Rodrigues C., Fernandez M., Megia A., Comino N., and et al., "Comparison of Intramuscular and Surface Electromyography Recordings Towards the Control of Wearable Robots for Incomplete Spinal Cord Injury Rehabilitation," *in Proceedings of the 8th IEEE RAS/EMBSs International Conference for Biomedical Robotics and Biomechatronics*, New York, pp. 564-569, 2020. DOI: 10.1109/BioRob49111.2020.9224361

[27] Saurav Z., Mitu M., Ritu N., Hasan M., and et al., "A New Method for Learning Decision Tree Classifier," *in Proceedings of the IEEE International Conference Electrical, Computer and Communication Engineering*, Chittagong, pp. 1-6, 2023. https://doi.org/10.1109/ECCE57851.2023.10101557

[28] Senagi K. and Jouandeau N., "Parallel Construction of Random Forest on GPU," *Journal of Supercomputing*, vol. 78, no. 8, pp. 10480-10500, 2022. https://doi.org/10.1007/s11227-021-04290-6

[29] Wahid F., Tafreshi R., Al-Sowaidi M., and Langari R., "Subject-Independent Hand Gesture Recognition Using Normalization and Machine Learning Algorithms," *Journal of Computational*

Science, vol. 27, pp. 69-76, 2018. https://doi.org/10.1016/j.jocs.2018.04.019

[30] Waris A., Niazi I., Jamil M., Englehart K., and et al., "Multiday Evaluation of Techniques for EMG-based Classification of Hand Motios," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 4, pp. 1526-1534, 2019. https://doi.org/10.1109/JBHI.2018.2864335

[31] Wen W., Shi J., He B., Li Q., and Cui B., "ThunderGBM: Fast GBDTs and Random Forests on GPUs," *Journal of Machine Learning Research*, vol. 21, no. 108, pp. 1-5, 2020. https://www.jmlr.org/papers/v21/19-095.html

[32] Yuan B., Hu D., Gu S., Xiao S., and Song F., "The Global Burden of Traumatic Amputation in 204 Countries and Territories," *Frontiers in Public Health*, vol. 11, pp. 1-15, 2023. https://doi.org/10.3389/fpubh.2023.1258853

[33] Zhao J., She J., Wang D., and Wang F., "Patient Classification Based on sEMG Signals Using Extreme Gradient Boosting Algorithm," *in Proceedings of the 7th International Workshop on Advanced Computational Intelligence and Intelligent Informatics*, Beijing, pp. 1-5, 2021. https://iwaciii2021.bit.edu.cn/docs/2021-12/bf17766b3bd04ef98dc5faf3706d14ce.pdf

Toka Fathi is a Ph.D. candidate in Computer Engineering at University of Mosul and a Lecturer in the Computer and Information Engineering Department at the Ninevah University. She received his M.Sc. degrees in Computer Engineering from University of Mosul. She is currently pursuing his Ph.D. degree in Computer Engineering at University of Mosul, Iraq. Her research interests include Digital Signal Processing, Artificial Intelligence, and Embedded Systems.

Mohammed A M Abdullah received the PhD degree from the school of Electrical and Electronic Engineering at Newcastle University, UK in 2017. He obtained the BSc and MSc degrees in Computer Engineering in 2008 and 2010, respectively from Iraq. He is currently working as the Head of the Computer and Information Engineering Department at Ninevah University, Iraq. His research interests are in the fields of Artificial Intelligence, Pattern Recognition and Image Processing. Dr Abdullah is a Senior Member of the IEEE, BMVA and an Associate Fellow of the Higher Education Academy (AFHEA) in UK.

Basil Shukr is a Senior Professor in Computer Engineering, he was born in Mosul, Iraq, on 1953. He received the B.S. and M.S. degrees from the Department of Electrical Engineering, University of Mosul, Iraq, in 1976 and 1979, respectively. He received his Ph.D. degree in 1996 in Processor Architecture for Fractal Signals. His researches are in the fields: Real Time Systems, Computer Architecture, Operating System, Image Processing and Embedded Systems and other related subjects.