

DEEP HYBRID NEURAL NETWORKS FOR PREDICTING MISSING SEGMENTS IN sEMG TIME SERIES DATA

*Jihane Ben Slimane (1,2)**

⁽¹⁾ Department of Computer Sciences, Faculty of Computing and Information Technology, Northern Border University, Rafha; ⁽²⁾ National Engineering School of Tunis, LR11ES20 Analysis Design and Control of Systems Laboratory, University of Tunis El Manar
Saudi Arabia ⁽¹⁾ & Tunisia ⁽²⁾

* Corresponding Author, e-mail: jehan.saleh@nbu.edu.sa

Abstract: Surface electromyography (sEMG) has illustrated noteworthy findings over different disciplines; however, it suffers from several issues like signal interference, noise, and interruptions. In this research, the Savitzky-Golay filter was first used to extract meaningful data while preserving the overall shape of the data, and then two hybrid neural network models based on Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) were performed to predict the missing sEMG data. Their performance was compared with independent LSTM and GRU models using the coefficient of determination (R-squared), Root Mean Square Error (RMSE), and correlation coefficient (ρ). All models were trained, validated, and tested on extended and limited datasets. In addition, the optimal number of hidden neurons was determined experimentally for each condition. The outcomes indicated that the deep learning architecture based on sequential GRU and LSTM models outperformed all competitors with a prediction accuracy of 99.25% and 98.91% for the long and short training datasets, respectively.

Key words: GRU, LSTM, sEMG, deep learning, time-series prediction.

1. INTRODUCTION

In recent decades, sEMG has shown promising results and proven effective in medical diagnosis, rehabilitation of neurological problems, and sports science. Quite simply, sEMG is user-friendly, controllable, and more comfortable compared to invasive EMG. However, despite its advantages, sEMG faces a great deal of challenges. These may include human tissue impedance, power line noise, ground contact, electronic equipment artifacts, basic wandering, motion artifacts, and crosstalk [1]; This can affect sEMG signals and cause noise, interference, attenuation, or interruption. To overcome these issues, researchers often resort to a software method that contains many digital filters such as the Wiener filter [2], Butterworth filter [3], and Savitsky-Golay filter

which is known for its adjustable nature and ability to reduce noise and high-frequency variations, while preserving data signal patterns [4]. Therefore, in this paper, we propose applying the Savitzky-Golay filtering technique in the pre-processing phase. Then, for handling the prediction of missing sEMG data, Duan et al. [5] proposed a Data Split Reorganization (DSR) and Weight-based Multiple neural network Voting methods (WMV). Furthermore, Akmal et al. [6] introduced mathematical techniques using Tensor-Based Methods (TBM). Subsequently, in the processing phase, we apply two innovative deep learning architectures: the concatenated and sequential Hybrid GRU-LSTM models which aim to predict missing segments of sEMG signals. Following that, the prediction performance of the proposed hybrid models will be discussed, and the results will be compared with the standard LSTM and GRU models and then with previous works.

The performance of all studied models is assessed within long and short training dataset portions. Then, the predictions are evaluated through a set of performance metrics including the R-squared (R^2), RMSE, and ρ to give a rounded evaluation of the prediction models. However, it should be noted that the sequential hybrid model consistently outperforms all models across all evaluation metrics with $R^2=0.7642$, RMSE=0.0256, and $\rho=0.8938$ for the long training dataset, and $R^2=0.6969$, RMSE=0.0321, and $\rho=0.8575$ for the short dataset. The consistent outperformance of the sequential hybrid model in predicting both the extended and limited training datasets indicates a remarkable efficiency of deep learning methodologies, achieving an accuracy rate of 99.25% and 98.91% for long and short training datasets, respectively.

The implementation of the Hybrid GRU-LSTM model in predicting missing sEMG segments guarantees enhanced accuracy and reliability in the prediction measurements. As a result, this research contributes to the advancement of sEMG technology, empowering healthcare professionals to provide personalized and targeted rehabilitation programs with improved efficacy.

2. SIGNAL ACQUISITION AND PRE-PROCESSING

2.1. Electrode placement

The Flexor Carpi Radialis (FCR) muscle was chosen to conduct this research because of its pivotal role in moving the hand. It can perform three main movements of the wrist: the neutral position (NP), radial deviation (RD), and wrist flexion (WF). Additionally, the Median Nerve connects to the muscle in the Innervation Zone (IZ). As shown in Figure 1, the study [7] indicated that the IZ extends from the elbow to the vicinity of the muscle belly. Therefore, noise may be detected if the electrodes are placed there. This issue can be mitigated by attaching the electrodes to the muscle belly.

2.2. Electrode configuration

A closely located bipolar electrode setup was chosen in this work to ensure that the electrodes receive similar levels of interference and to avoid IZ. As shown in Figure 1, the three electrodes forming the bipolar configuration are:

- The Active Electrode 1 should be placed on the FCR belly.

- The Active Electrode 2 should be placed close to the Active Electrode 1 at a distance of 2 or 3 cm towards the elbow, making sure not to include the IZ.
- The Ground (GND) Electrode must be connected to the body in an electrically inactive point such as tendons or bony prominences. Since the dorsum of the wrist is a region of tendon extension, it was designated as a GND electrode location.

The principle of the sEMG bipolar configuration is to capture the signal of the muscle with the least possible noise and minimal neighboring muscle interference. However, this configuration does not cover the overall activity of the studied muscle.

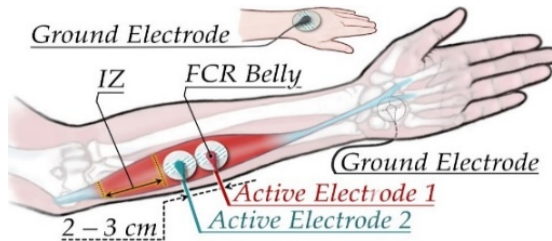


Figure 1. sEMG Bipolar Configuration

2.3. Signal acquisition system

To acquire sEMG data, an electronic circuit was specifically designed to amplify and filter the myoelectric signals with a sampling frequency acquisition of 1kHz. Figure 2a shows an actual view of the acquisition system. Additionally, Figure 2b details the system wiring diagram, it is designed with three input pins (V_1 , V_2 , 5V), an output pin V_3 -out, and the GND. The obtained signals are then transmitted to Arduino Uno via pin A0 and the computer via USB. To perform this communication, the pyFirmata 1.1.0 library was imported to the Python IDE, enabling it to interact with the Arduino Uno through the COM ports. Finally, the analog data is digitized by reading it from pin A0.

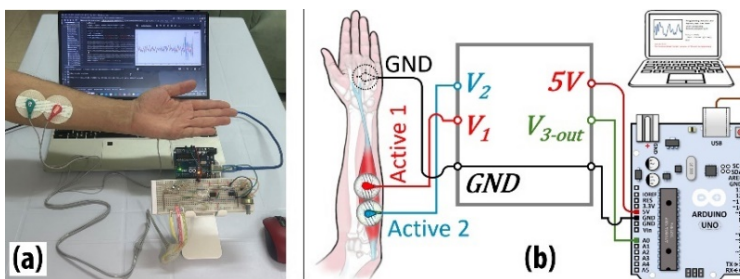


Figure 2. sEMG signal acquisition system: a) Actual view, b) Wiring diagram

2.4. Savitzky-Golay filter

In this work, we applied the Savitzky-Golay filter, an effective signal pre-processing technique that reduces noise and high-frequency fluctuations while preserving signal patterns. It consists of a sliding window approach of width w , where a polynomial of order o is calculated to fit the data points in each window [4]. Figure 3 illustrates the obtained sEMG signals versus the Savitzky-Golay filtered signals.

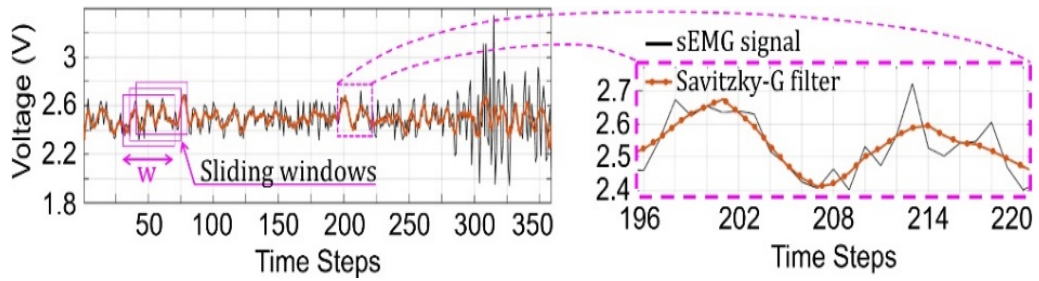


Figure 3. Savitzky-Golay Filtered Signals

The expanded window shows how the selected filter extracts meaningful information by smoothing the signal while maintaining its overall direction. In this study, $w = 25$ and $o = 8$.

3. PROCESSING DATA

In this section, four different Recurrent Neural Network (RNN) architectures are implemented to predict missing sEMG signals: Firstly, we apply the standard LSTM and GRU models according to Figure 4. Secondly, we introduce innovative additions to the LSTM and GRU models in the form of Concatenated Hybrid (Conc.Hyb) as depicted in Figure 5, and Sequential Hybrid (Seq.Hyb) architectures as shown in Figure 6. These two proposed hybrid models aim to leverage the strengths of both LSTM and GRU, providing reliable predictive capabilities for missing sEMG segments. The standard LSTM and GRU architectures are presented in Figure 4a, and Figure 4b, respectively.

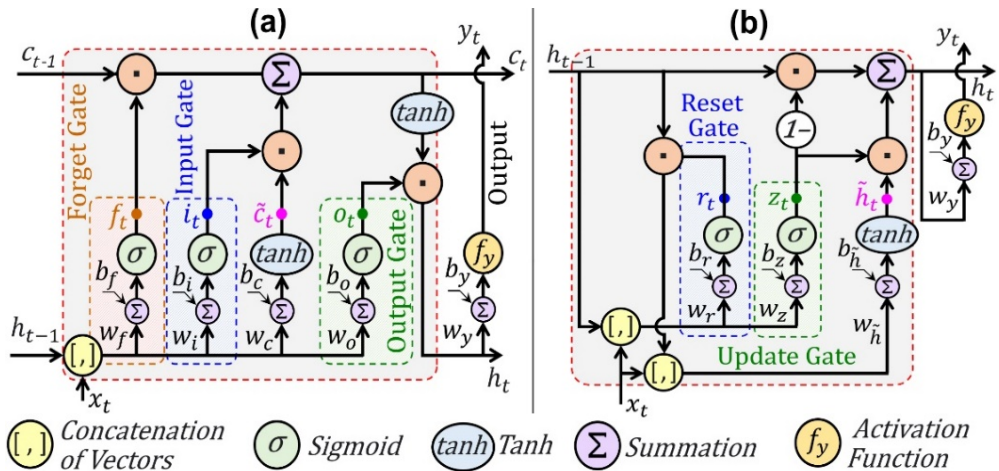


Figure 4. LSTM and GRU: a) Single LSTM cell, b) Single GRU cell

In Figure 4a: t is the current time step, x_t is the current input, h_t is the current hidden state, h_{t-1} is the previous hidden state, C_t is the current cell state, C_{t-1} is the previous cell state, and y_t is the output. w_f , w_i , w_c , w_o , and w_y are respectively the weights of the forget gate, input gate, candidate, output gate, and the output. b_f , b_i , b_c , b_o , and b_y are respectively the biases of the forget gate, input gate, candidate, and the output gate.

In Figure 3b: w_r , w_z , $w_{\tilde{h}}$, and w_y are respectively the weights of the reset gate, update gate, candidate hidden state, and the output. b_r , b_z , $b_{\tilde{h}}$, and b_y are respectively the biases of the reset gate, update gate, candidate hidden state, and the output.

3.1. Processing data via standard LSTM model

In this subsection, we present the architecture of the standard LSTM that comes with greater capabilities to address the limitations of traditional RNNs. It was mainly designed to reduce learning time and alleviate the vanishing gradient problem [8]. Regarding memorization, data flows through the recurrent LSTMs via two states: the hidden state (h_t) and the cell state (C_t), representing short-term memory and long-term memory, respectively. As shown in Figure 4a, the LSTM consists of three gates: the forget gate f_t , the input gate i_t and candidate $\tilde{c}_t$, and the output gate o_t , which are represented by the following equations:

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

$$\tilde{c}_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \quad (2)$$

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$c_t = \tilde{c}_t \odot i_t + c_{t-1} \odot f_t \quad (4)$$

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

$$y_t = f_y(w_y h_t + b_y) \quad (7)$$

Where, $[h_{t-1}, x_t]$ is the concatenation of h_{t-1} and x_t ; f_y is the activation function of the output.

3.2. Processing data via standard GRU model

The GRU implemented in this paper is also a type of RNNs with a gated structure derived from LSTM. Hence it retains the ability to learn long-term sequential data and to address the vanishing gradient problem. Despite its simplified gated structure, GRU demonstrates remarkable applicability for scenarios with limited training data [9]. To evaluate the performance of all models in this study, the predictions are tested on two different datasets: the first consists of a long training dataset of the wrist in NP (280 time steps: Portion A) and the second consists of a short training dataset of the wrist in RD (70 time steps: Portion B). As shown in Figure 4b, the GRU consists of two gates: the reset gate (r_t) with the candidate hidden state $\tilde{h}_t$ and the update gate (z_t), which are represented by the following equations:

$$r_t = \sigma(w_r \cdot [h_{t-1}, x_t] + b_r) \quad (8)$$

$$\tilde{h}_t = \tanh(w_{\tilde{h}} \cdot [h_{t-1} \odot r_t, x_t] + b_{\tilde{h}}) \quad (9)$$

$$z_t = \sigma(w_z \cdot [h_{t-1}, x_t] + b_z) \quad (10)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (11)$$

3.3. Processing data via concatenated GRU-LSTM hybrid model

The proposed architecture in this subsection involves concatenating GRU and LSTM in one hidden layer. This hybrid configuration aims to leverage the strengths of both models, LSTM is intended to excel GRU at memorizing long-term dependencies due to its cell states, while GRU is efficient at quickly updating information. This approach is intended to capture long-term dependencies effectively while maintaining the ability to update and adapt to new information efficiently. Figure 5 shows the proposed concatenated hybrid GRU-LSTM model (Conc.Hyb).

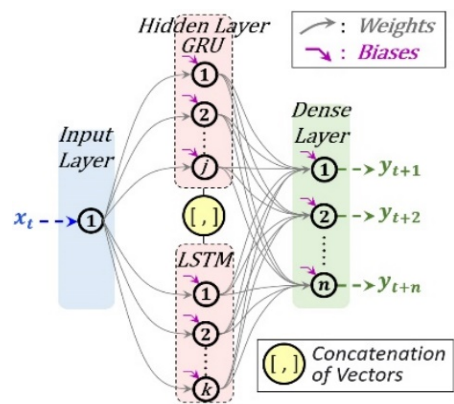


Figure 5. Concatenated hybrid GRU-LSTM model

3.4. Processing data via sequential GRU-LSTM hybrid model

In this experiment, the objective is to create a deep architecture capable of learning more complex patterns by sequentially stacking the GRU and LSTM models, forming two hidden layers. The output of the GRU layer is fed as input to the subsequent LSTM layer for further processing. Figure 6 illustrates the architecture of the sequential hybrid GRU-LSTM model (Seq.Hyb).

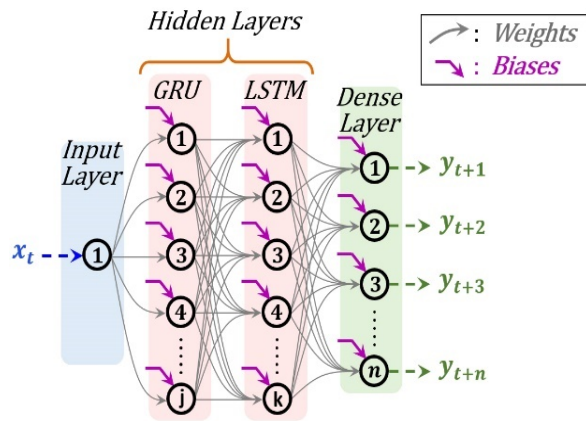


Figure 6. Sequential hybrid GRU-LSTM model

Both Hybrid models are tested on the same portions (A and B) corresponding to the long and short training datasets, respectively.

3.5. Prediction algorithm

The code was developed using Python 3.10.4 and related libraries such as numpy 1.26.4 for numerical computing of arrays and matrices, keras 2.15.0 for building NN models, tensorflow 2.16.1 for training neural networks using backpropagation technique, and matplotlib 3.8.2 for creating visualizations.

The algorithm consists of several steps to train and evaluate a neural network model for making predictions. Here is a detailed description:

- Preparing input-output sequences: The first step involves organizing the data into specific sequences with input and output pairs.
- Building the NN model: The NNs built in this algorithm are GRU, LSTM, or Hybrid models (Sequential or Concatenated).
- Splitting data: 80% for training, 20% for evaluating, and the last 10 time-steps for testing.
- Z- score standardization: To make the data more comparable and reliable, the Z-score standardization process is applied.
- Model training and validation: During the training and validation phase, the algorithm keeps track of the loss history, which represents the errors made by the model during training.
- Reverse standardization and making predictions: Once the model has learned from the training data, the predictions it makes are converted back to their original form using reverse standardization.
- To determine the optimal number of epochs and neurons in the hidden layers, a series of experiments were performed within the range of 20-360 neurons. Specifically, the relevant neuron numbers for this study ranged between 200 and 340, allowing for a focused exploration of their effects on the model's performance.

4. RESULTS AND DISCUSSION

To determine the optimal number of epochs and neurons in the hidden layers, a series of experiments were performed within the range of 20-360 neurons. Specifically, the relevant neuron numbers for this study ranged between 200 and 340, allowing for a focused exploration of their effects on the model's performance.

4.1. Training and validation results

The training algorithm starts by splitting the dataset into three distinct subsets: the testing set (the last 10 time steps), the training set (80% of the dataset), and the validation set (20% of the remaining data). These subsets are then standardized using z-score standardization expressed as follows [10]:

$$z_i = \frac{x - \bar{x}}{\sqrt{\frac{\sum_{i=1}^N x_i^2 - \bar{x}^2}{N}}} \quad (12)$$

Where, z is the z-score of the dataset X with N data points; x_i is an element of X ; and $\bar{X}$ is the mean of X .

During the NN model training and validation, the loss is calculated using the Mean Square Error (MSE) function. Figure 7 depicts the training loss and validation loss of all the proposed models using both long and short training datasets. The optimal number of epochs is obtained when the validation loss stops decreasing.

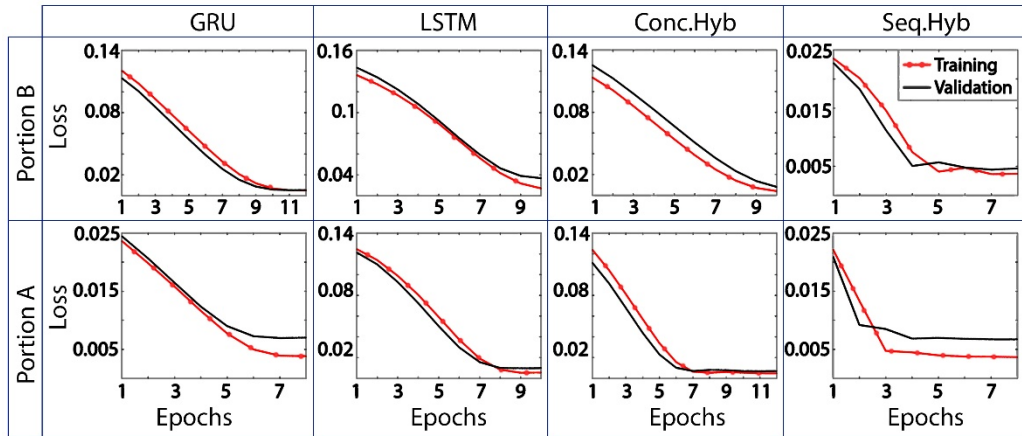


Figure 7. Training and Validation Loss of All NN Models Under Study

4.2. Performance analysis of NN models

To evaluate the performance of LSTM, GRU, and hybrid models in the predicting process, we resort to RMSE, R^2 , and ρ due to their efficiency in determining RNN hyperparameters such as the training and validation dataset sizes, the number of epochs, and the number of hidden neurons. The RMSE, R^2 [11], and ρ [12] are expressed as follows:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2} \quad (13)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (14)$$

$$\rho = \frac{\sum_{i=1}^N ((x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\sum_{i=1}^N ((x_i - \bar{x})^2 (y_i - \bar{y})^2)}} \quad (15)$$

Where, N is the number of observations, $\hat{y}_i$ is the predicted output, y_i is the observed output, $\bar{y}$ is the mean of the observed outputs, x_i is the predicted output, and $\bar{x}$ is the mean of the predicted outputs.

To determine the optimal number of hidden neurons for each proposed NN model, the average R^2 and RMSE of 50 consecutive simulations were calculated for each incremented number of hidden neurons. (Table 1) displays the variations in R^2 and RMSE as the number of hidden neurons increases for all proposed NN models.

Table 1. Error metrics of the proposed NN models for the prediction of portion A and B.

Model	Portion	Error Metrics	Number of Neurons							
			200	220	240	260	280	300	320	340
GRU	A	R^2	0.3553	0.6626	0.5579	0.5526	0.4090	0.5937	0.5945	0.5328
		RMSE	0.0424	0.0307	0.0351	0.0353	0.0406	0.0337	0.0336	0.0361
	B	R^2	-0.4104	-0.1092	0.3253	0.6760	0.2352	-0.0772	-0.5884	0.0106
		RMSE	0.0693	0.0614	0.0479	0.0332	0.0510	0.0605	0.0735	0.0580
LSTM	A	R^2	0.6181	0.5850	0.6960	0.4716	0.6463	0.6498	0.5551	0.2706
		RMSE	0.0326	0.0340	0.0291	0.0384	0.0314	0.0313	0.0352	0.0451
	B	R^2	0.1356	-0.3293	0.0063	-0.1202	-0.0159	0.3073	0.3826	-0.1197
		RMSE	0.0542	0.0673	0.0581	0.0617	0.0588	0.0485	0.0458	0.0617
Hybrid Conc. GRU LSTM	A	R^2	0.7530	0.7584	0.4164	0.5315	0.6430	0.7043	0.1388	0.6885
		RMSE	0.0262	0.0260	0.0404	0.0362	0.0316	0.0267	0.0491	0.0295
	B	R^2	-0.9186	-0.8854	-1.3142	-1.2645	-0.0462	-0.5800	-1.3355	-0.9802
		RMSE	0.0808	0.0801	0.0888	0.0878	0.0597	0.0733	0.0892	0.0821
Hybrid Seq. GRU LSTM	A	R^2	0.3237	0.1201	0.7642	0.7112	0.2739	0.3233	-0.0899	-0.1342
		RMSE	0.0435	0.0496	0.0256	0.0284	0.0450	0.0435	0.0552	0.0563
	B	R^2	-0.9741	-0.5620	0.3185	0.4653	-0.1465	0.6969	-0.8939	-1.0804
		RMSE	0.0820	0.0729	0.0481	0.0426	0.0625	0.0321	0.0803	0.0842

To better understand (Table 1), the best R^2 and RMSE metrics were extracted and presented in (Table 2) for long training dataset and (Table 3) for short training dataset. The best R^2 metrics are the closest to 1, indicating a higher level of accuracy in following patterns, and the best RMSE results are the closest to 0, signifying lower prediction errors. ρ close to 1 indicates a strong positive linear relationship between the observed and predicted data. Based on the R^2 and RMSE results, when examining the portion (A) of the long training dataset, it is evident that the LSTM outperforms the GRU model as ($R^2_{LSTM} = 0.696$) > ($R^2_{GRU} = 0.6626$). Additionally, ($RMSE_{LSTM} = 0.0291$) < ($RMSE_{GRU} = 0.0307$). By the same reasoning, we notice that the sequential hybrid model represents the best predicting architecture for portion (A), followed by the concatenated hybrid model, then LSTM, and finally GRU. However, ρ provided different, but not distant, conclusions. Contrary to what the R^2 has provided, ($\rho_{Conc.Hyb} = 0.90089$) > ($\rho_{Seq.Hyb} = 0.89382$). Then, we can say that these two hybrid models demonstrate a comparable level of performance in an extended training dataset. Therefore, it is crucial to employ various metrics and datasets to assess the performance of NN models before making decisions.

Table 2. Optimal NN model hyperparameters for predicting portion A (long training dataset)

Model	Neurons	Epochs	R^2	RMSE	ρ	Accuracy (%)
GRU	220	12	0.6626	0.0307	0.85442	98.98
LSTM	240	10	0.6960	0.0291	0.85271	99.08
Conc.Hyb	220	10	0.7584	0.0260	0.90089	99.22
Seq.Hyb	240	8	0.7642	0.0256	0.89382	99.25

In the same way, (Table 3) showcases the optimal R^2 , RMSE, and ρ metrics for portion (B) of the limited training dataset. In contrast to (Table 2), all metrics indicate

that during a short training dataset, GRU demonstrates superior performance compared to LSTM. However, it is important to note that the Conc.hyb model shows a low value of ρ and a negative R^2 value, indicating poor adherence to the short training patterns of the moving wrist signals. Consequently, calculating accuracy for this model would not yield meaningful results given its inadequate adherence to the data patterns. Moreover, it is worth mentioning that the Seq.Hyb model consistently outperforms other models in all instances.

Table 3. Optimal NN model hyperparameters for predicting portion B (short training dataset)

Model	Neurons	Epochs	R^2	RMSE	ρ	Accuracy (%)
GRU	260	8	0.6760	0.0332	0.84536	98.87
LSTM	320	10	0.3826	0.0458	0.68891	98.40
Conc.Hyb	280	12	-0.0462	0.0597	0.24915	N/A
Seq.Hyb	300	8	0.6969	0.0321	0.85759	98.91

4.3. Prediction and testing results

Using the optimal hyperparameters obtained for all proposed NN models, the prediction of 10 time steps of signal interruption, was tested and visualized in Figure 8.

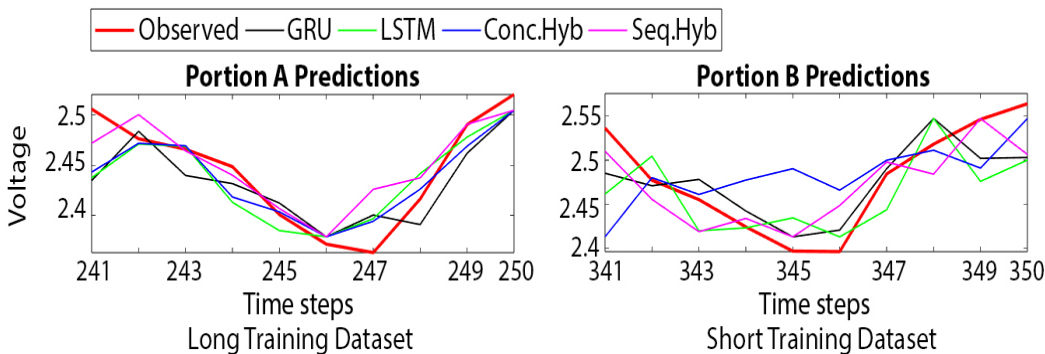


Figure 8. Prediction Results and Testing for All NN Models Under Study

Based on the visualization above, it is evident that there is a strong agreement between the predicted and observed values, particularly in the long training dataset of the non-moving wrist (portion A). This observation suggests that all of the prediction models, particularly the Seq.Hyb model, exhibit superior performance in capturing patterns and accurately predicting missing data when trained on longer datasets.

4.4. Comparisons with other methods

In order to evaluate the performance of our proposed prediction approaches, we compared our results with the findings of previous studies in the field of sEMG prediction. Specifically, we compared our results with the DSR and TBM methods proposed by Duan et al. [5] and Akmal et al. [6], respectively. The results shown in (Table 4) demonstrate the superior accuracy of the LSTM, GRU models compared to the DSR and TBM methods, highlighting the effectiveness of the proposed Hybrid architectures, in particular, the Seq.Hyb model.

Table 4. Comparison of the prediction accuracy with previous works

Model	DSR	TBM	LSTM	GRU	Conc.Hyb	Seq.Hyb
Accuracy (%)	93.66	95	98.40	98.87	98.19	98.91

5. CONCLUSION AND FUTURE WORK

Through the prediction analysis of missing sEMG signals within the static wrist portion utilizing an extensive training dataset, contrasting findings have been observed. The value of R^2 and the error rate RMSE in the LSTM model appeared slightly better in comparison to GRU; however, ρ showcased a reversed scenario. This infers a nearly equivalent performance between the GRU and LSTM models. Similar outcomes were noticed for both the concatenated and sequentially structured hybrid models. It stands to reason that over a larger training dataset, hybrid models exhibit enhanced performance over singular model types. To discern the more effective model, examining their functionality over a smaller training dataset revealed distinct disparities. The concatenation-based hybrid model displayed a negative R^2 value, signifying a considerable divergence from the expected training patterns. Conversely, the sequential hybrid model surpassed all competing models, proving its exceptional prowess in grasping and predicting sEMG signals during wrist activities, even with constrained dataset size. Thus, it could be stated that the sequential hybrid model blending GRU and LSTM architectures, through the deep learning framework, excelled in efficiency within both segments (A and B). This model stands superior among the rest, highlighting its proficiency in the accurate capture and prediction of sEMG signals under diverse setups. However, there are limitations to consider. The research specifically focused on the FCR muscle to investigate the performance of the proposed hybrid models. In particular, the concatenated hybrid model exhibited low performance when trained on a short dataset with dynamic wrist positions. For future research, it is recommended to explore the generalizability and scalability of the proposed models across different muscles and activities. Furthermore, exploring alternative deep learning architectures and incorporating additional physiological features may further enhance the accuracy of sEMG signal prediction.

ACKNOWLEDGEMENT

The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number “NBU-FFR-2024-2099-04”.

REFERENCES

[1] Gabriel DA. Teaching Essential EMG Theory to Kinesiologists and Physical Therapists Using Analogies Visual Descriptions, and Qualitative Analysis of Biophysical Concepts. *Sensors*, vol.22, no.17, 2022, art.6555. DOI: 10.3390/s22176555

[2] Boyer M, Bouyer L, Roy J-S, Campeau-Lecours A. Reducing Noise, Artifacts and Interference in Single-Channel EMG Signals: A Review. *Sensors*, vol.23, no.6, 2023, art.2927. DOI: 10.3390/s23062927

- [3] De Luca CJ, Donald Gilmore L, Kuznetsov M, Roy SH. Filtering the surface EMG signal: Movement artifact and baseline noise contamination. *Journal of Biomechanics*, vol. 43, no.8, 2010, 1573-9. DOI: 10.1016/j.jbiomech.2010.01.027
- [4] Ochieng PJ, Maróti Z, Dombi J, Krész M, Békési J, Kalmár T. Adaptive Savitzky–Golay Filters for Analysis of Copy Number Variation Peaks from Whole-Exome Sequencing Data. *Information*. Vol. 14, no.2, 2023, 128. DOI: 10.3390/info14020128
- [5] Duan F, Yang Y. Recognizing Missing Electromyography Signal by Data Split Reorganization Strategy and Weight-Based Multiple Neural Network Voting Method. *IEEE Transactions on Neural Networks and Learning Systems*. 2021, pp.1-10. DOI: 10.1109/TNNLS.2021.3105595
- [6] Akmal M, Zubair S, Jochumsen M, Kamavuako EN, Niazi I. A Tensor-Based Method for Completion of Missing Electromyography Data. *IEEE Access*. 2019; pp.1. DOI: 10.1109/ACCESS.2019.2931371
- [7] Beretta Piccoli M, Rainoldi A, Heitz C, Wüthrich M, Boccia G, Tomasoni E, et al. Innervation zone locations in 43 superficial muscles: toward a standardization of electrode positioning. *Muscle Nerve*. Vol. 49, no.3, 2014, pp.413-21. DOI: 10.1002/mus.23934
- [8] Le X-H, Ho HV, Lee G, Jung S. Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. *Water*, vol.11, no.7, 2019, 1387. DOI: 10.3390/w11071387
- [9] Li X, Ma X, Xiao F, Wang F, Zhang S. Application of Gated Recurrent Unit (GRU) Neural Network for Smart Batch Production Prediction. *Energies*, vol.3, no.3, 2020, 6121. DOI: doi:10.3390/en13226121
- [10] Shahumyan H, Williams B, Petrov L, Foley W. Regional Development Scenario Evaluation through Land Use Modelling and Opportunity Mapping. *Land*, vol.3, no. 3, 2014, 1180-213. DOI: 10.3390/land3031180
- [11] Jierula A, Wang S, OH T-M, Wang P. Study on Accuracy Metrics for Evaluating the Predictions of Damage Locations in Deep Piles Using Artificial Neural Networks with Acoustic Emission Data. *Applied Sciences*, vol.11, no.5, 2021, 2314. DOI: 10.3390/app11052314
- [12] Burgund D, Nikolovski S, Galić D, Maravić N. Pearson Correlation in Determination of Quality of Current Transformers. *Sensors*, vol.12, no.5, 2023, 2704. DOI: 10.3390/s23052704

Information about the author:

Dr. Jihane Ben Slimane is currently, an Assistant Professor in the Department of Computer Sciences, Faculty of Computing and Information Technology, Northern Border University, Rafha, Kingdom of Saudi Arabia. Her areas of expertise and research interests include Artificial Intelligence, Machine Learning, Neural Networks, Fault Diagnosis, and Cybersecurity.

Manuscript received on 17 May 2024