Evaluating the Impact of Model Complexity and Input Representations on EMG-Based Hand Gesture Classification

Anna Carpenter, August Hao
Duke University

Abstract

As prosthetic control technology advances, the use of sEMG signals to drive gesture-based control has become a popular avenue. These signals are a direct reflection of muscle activation patterns-even in individuals without a limb-and can be used to interpret intended movements. Existing research has analyzed the robustness of various CNN, LSTM, and CNN-LSTM hybrid architectures. Most studies investigating the implementation of such techniques only analyze model performance on one subset of hand gestures, leading to uncertainty in generalizing findings to tasks of different complexities. This project aims to determine the effectiveness of popular techniques for increasing model robustness over varying numbers of hand gestures to determine their performance across tasks of varying complexity. In addition, this study investigates the impact of numerous input representations on model performance. 3 CNN models developed with varying numbers of convolution and fully connected layers were compared on tasks of 10, 25, and 53 hand gestures. CNN, LSTM, CNN-LSTM hybrid models were also compared across various task complexities. Multilayer attention (channel and temporal attention) was implemented in both CNN and LSTM models. Finally, CNN models with various domain transformations, including time domain (TD), fast Fourier transform (FFT), discrete wavelet transform (DWT), and a time domain - Fast Fourier transform (TD-FFT) multibranch model, were compared on a baseline CNN architecture. The baseline CNN model (middle complexity) resulted in higher testing accuracy than simpler and more complex models for every classification task, with the simpler model declining in performance and the more complex model improving over an increased number of gestures. The CNN model outperformed the LSTM and CNN-LSTM models in every task complexity. Attention implementation was beneficial in the LSTM architecture but not CNN. The time domain-fast Fourier transform multibranch input representation consistently outperformed the other representations tested.

1 Introduction

Surface electromyography (sEMG) is a technique that records the electrical activity produced by muscle contractions. By detecting action potentials through surface electrodes, sEMG enables the inference of muscle intent even in individuals without intact limbs, making it a valuable tool in the development of myoelectric prosthetic control systems. These systems aim to translate sEMG signals into reliable prosthetic hand gestures, enhancing functionality and user experience.

Recent research has explored various approaches to improve the classification of sEMG signals for prosthetic control. Techniques include the application of signal transformations to extract meaningful features, the development of deep learning architectures such as convolutional neural networks (CNNs), long short-term memory (LSTM) networks, and hybrid CNN-LSTM networks. Attention mechanisms have also been incorporated into model designs to better capture spatial and temporal

dependencies in sEMG signals. While many studies have demonstrated improvements in classification performance, challenges remain in optimizing models for both accuracy and scalability across varying task complexities.

The main contributions of this paper are summarized as follows:

1. We proposed the implementation of various techniques to improve model performance: varying model complexities (implemented on CNN models), changing model type (CNN, LSTM, CNN-LSTM hybrid), and the implementation of multi-layer attention on CNN and LSTM models.
2. Due to the complicated pattern of sEMG signals, we will assess the performance of a benchmark CNN model using input representations representing various time and frequency domain characteristics. These transforms include raw (time domain) data, the discrete wavelet transform, fast Fourier transform, and a time domain-fast Fourier transform multi-branch model.
3. Motivated by the desire to see how the proposed changes to model architectures impact tasks of different complexities, we will test the proposed models and input representations over a range of model complexities. Specifically, we will implement the variations listed above on tasks with 10, 25, and 53 different hand gestures to determine their scalability.

In this paper, we will first discuss related research involving sEMG-based gesture recognition methods, including current research involving the classification of sEMG signals across different domains and various CNN-LSTM hybrid models. We then introduce the proposed models. Thirdly, we will show the experimental results for model complexity, model type, input representation, and attention implementation. Lastly, we draw conclusions and discuss potential future work.

1.1 Background

Surface electromyography measures the electrical activity generated by muscle action potentials. These signals are detectable in individuals with and without a hand, making sEMG especially valuable for prosthetic development, where signals are used to interpret intended movements [11]. However, analyzing and classification sEMG signals can be difficult due to their complex and variable patterns, especially during motion [11]. Prior research has explored various methodologies to improve the analysis of sEMG signals. Signal transformations and feature extraction techniques are commonly used in muscle analysis, particularly for assessing muscle fatigue. For example, Dantas, et al. and Costa, et al. implement both Short Time Fourier Transforms and Continuous Wavelet Transforms to assess muscle fatigue in isometric/dynamic exercise and maximal constant load dynamic exercise, respectively [4][3]. Additionally, Too, et al. evaluates both time and frequency domain characteristics using linear discriminant analysis to classify gestures. Their findings suggest that frequency domain features yield the best performance in EMG signal classification applications [12].

sEMG data contains both temporal and spatial components. Temporal information enables prosthetic hand control systems to predict movements more accurately and in real time by capturing the dynamics, sequence, and evolution of signals that correspond to human hand movements. This is crucial to improve the precision, flexibility, and functionality of prosthetics, allowing smoother and more natural control. Spatial feature extraction is crucial for interpreting the specific location and intensity of muscle activations, allowing a prosthetic hand system to understand and predict movements more accurately. By capturing spatial information, systems can improve gesture recognition, provide finer control, reduce signal ambiguity, and make the prosthetic hand more intuitive and responsive to the user.

2 Related Work

Model complexity has been shown to significantly impact the performance of gesture classification models, particularly when working with sEMG signals that contain rich spatial and temporal patterns. Shi, et al. compare various CNN, RNN, and CNN-RNN architectures with different numbers of filters to study the effects of model complexity on EMG-based gesture classification [10].

CNNs, LSTMs, and CNN-LSTMs are among the most widely used architectures for EMG gesture classification. Bao, et al. proposed a CNN-LSTM hybrid model to extract spatial-temporal features

from sEMG data [2]. This model outperformed traditional CNN and LSTM models in a hand gesture regression setting [2]. Similarly, Koch, et al. explored the regression of hand movements from sEMG signals using RNNs, single and stacked LSTM networks, and ConvLSTM models, highlighting the effectiveness of sequential modeling approaches [9].

Incorporating attention mechanisms into deep learning models has been shown to further enhance the ability to focus on the most informative features, improving classification performance in EMG applications. Shi, et al. proposed an attention-based hybrid CNN-RNN architecture for gesture recognition [10].

3 Methods

We utilized the Ninapro DB1[1] dataset, containing sEMG data for 27 subjects for 53 hand motions (Appendix B)[1]. This data was acquired using 10 Otto Bock MyoBock 13E200 electrodes, where each subject was asked to perform 10 repetitions of 52 different movements. Electrodes 1-8 are equally spaced around the forearm. 9 and 10 are placed on the flexor and extensor digitorum superficialis muscles. A cyberglove is used to record subjects' hand movements (Exercise D). The gestures are divided into:

- **Exercise 1:** basic finger movements
- **Exercise 2:** isometric and isotonic hand configurations; basic wrist actions
- **Exercise 3:** grasping and functional movements

The Ninapro DB1 dataset preprocesses the sEMG signals by segmenting raw EMG signals into fixed-length windows to isolate each movement event. We then sorted the signal data, transforming them into four input domains: raw time domain (TD), discrete wavelet transform (DWT), fast Fourier transform (FFT) and a hybrid TD-DWT representation (Appendix F). We evaluated three model architectures–CNN, LSTM, and a CNN-LSTM hybrid (Appendix C). Metrics were averaged across subjects to compare performance. (See Appendix A for model implementation code.)

3.1 Model Architectures

Throughout this project, we systematically evaluated the performance of three types of model architectures: CNN, LSTM, and CNN-LSTM. These models were selected based on the nature of sEMG signals, which exhibit both spatial and temporal features-well-suited to the strengths of CNNs and LSTMs, respectively. Additionally, we aimed to leverage both types of feature extraction through a hybrid CNN-LSTM model, a decision informed by insights from existing literature. To investigate how model complexity influence performance across different classification tasks, we designed 3 CNN models of varying complexity. We compared these model architectures against a baseline CNN trained on time-domain (TD) inputs from the raw data format provided by the Ninapro DB1 dataset. These models were evaluated on their ability to classify 10, 25, and 53 different hand gestures.

1. **Baseline CNN:** Our baseline CNN consists of 3 convolution layers, combining for a total of 224 filters, and 2 fully connected (FC) layers. For 10 gestures, this model consists of 133,514 parameters.
2. **Simple CNN:** The simple CNN model only consists of 1 convolution layer, with 32 filters, and 1 FC layer. For 10 gestures, this model consists of 2,762 total parameters.
3. **Complex CNN:** The complex CNN model consists of 4 convolution layers, for a total of 480 total filters, and 2 FC layers.
4. **LSTM:** Our LSTM model utilized 2 bidirectional LSTM layers, each with hidden size of 64, followed by a global average pool, and 1 FC layer. We don't perform any pooling between the two LSTM layers in order to preserve performance, since LSTMs are good at extracting temporal relations, and pooling decreases dimension, which would degrade the ability to extract features.
5. **CNN-LSTM:** Our CNN-LSTM model combines the 3 convolution layers of the baseline CNN model with the 2 bidirectional LSTM layers of the LSTM model, followed by 1 FC layer. This hybrid is able to extract features similarly to the plain CNN and LSTM

models. We permute the output of the 3 convolution layers to fit into the LSTM layers. Furthermore, we decided to limit pooling between convolution layers in order to preserve the time dimension for LSTM feature extraction.

3.2 Input Representations

We compared different input representations against a baseline CNN trained on TD inputs from the raw data provided by the Ninapro DB1 dataset. Each representation was trained on the same CNN architecture, and was tested on tasks involving 10, 25, and 53 gestures.

1. **Discrete Wavelet Transform:** A mathematical process that quantifies how much energy is contained within specific frequency bands at particular times within a signal–gains information about the original signal and retains information about when particular frequencies occur [5]. (See Appendix D.)

2. **Fast Fourier Transform:** A fast algorithm for computing the discrete Fourier transform [7]. Returns only the non-negative frequency terms– the FFT of a real signal is symmetric [7]. Our implementation computes the 1D n-point discrete fourier transform for real input (Appendix E).

3. **TD-FFT Multibranch:** Processes 2 different feature extraction models in parallel (one in TD, the other with FFT). Concatenates results and passes output through fully connected layers for classification (see Appendix F).

3.3 Attention

Attention allows models to focus on the relevant parts of the input data when making predictions. We used single-head, multilayer attention to pay attention to channel and temporal features in our CNN and LSTM models.

We apply the first layer of our attention - channel attention - after the first convolution block and first LSTM block, respectively, for the CNN and LSTM to dynamically recalibrate the importance of specific electrode channels. Channel attention is computed based off the idea of a Squeeze-and-Excitation (SE) Network (Appendix H), from the works of Hu, et al.[8] We first "squeeze" by performing a global average and max pool, capturing peaks and averages of activation per electrode channel. We then perform "excitation" by passing both pool branches through a multi-layer perceptron (MLP) model consisting of one FC layer compressing dimensionality by r=8 (good trade-off between complexity and cost), followed by ReLU, and another FC layer projecting back to the original number of channels. We add the two branches and use sigmoid to obtain useful channel attention weights, which we use to re-emphasize important channels and suppress weak ones.

We then apply temporal attention after the second convolution or LSTM layer to allow the model to emphasize when in time key muscle activations occur, and downplay more irrelevant times. In our CNN and LSTM models, we calculate temporal attention in two separate ways. For the CNN, we calculate "temporal" attention based off the spatial attention calculation model (Appendix I) from Han B., et al. [8]. We perform a global average and max pool across the channels, concatenate the features, and perform a 1-D Convolution, transforming into focusing on time series data. We apply sigmoid for attention weights. For the LSTM, we calculate temporal attention by implementing transformer style attention (Appendix J). Through linear transformations on the input, we project query (Q), key (K), and value (V). Attention is calculated with dot product and scaling, then activated using softmax.

4 Experiments and Results

Below are the experiments we performed and results we gathered from our methods discussed above. The data preprocessed from Ninapro DB1 was randomly split, ensuring representation of the overall dataset and to reduce bias, into 70% training, 20% validation, and 10% testing.

4.1 Model Complexity on Test Accuracy

From our results, we observed that all 3 complexities of the CNN model performed exceptionally well on the 10-gesture classification task, each achieving sur-90% accuracy. However, as the number of gestures increased, the simple model exhibited a notable decline in accuracy, with the performance gap between the simple mode and the more advanced architectures widening. This suggests the simple model increasingly underfit tasks with a larger number of classes.

The baseline model and the complex model achieved comparable accuracies across all gesture tasks, with the baseline maintaining a slight edge, consistently achieving the highest performance. The discrepancy between performance of the baseline model and the more complex model decreased as task complexity increased, indicating a reduction of overfitting in the complex model as the number of gesture classes grew.

Table 1: Model Complexity on Test Accuracy

Complexity	10 Gestures	25 Gestures	53 Gestures
Baseline	0.914098277	0.823539641	0.802871778
Complex	0.911437942	0.823488067	0.801829472
Simple	0.907696838	0.807082526	0.782498971

4.2 Model Type on Test Accuracy

CNN models demonstrated the highest performance, followed by CNN-LSTM models and then LSTM models. Accuracy gaps between models widened as task complexity increased, suggesting the CNN-based feature extraction was more effective than LSTM-based feature extraction for this sEMG classification task. While the CNN-LSTM model performed well on simpler tasks, its relative performance declined as the number of gesture classes grew.

Despite these results, all 3 model types are widely used in prior research on EMG feature extraction and gesture classification, and performance can vary significantly depending on specific architecture implementations. In our case, the CNN model achieved the best results, highlighting the importance of capturing spatial and frequency relationships inherent in muscle activations. This ranged from subtle differences between similar gestures (e.g., holding up 1 vs. 2 fingers) to larger signal variations during more intensive movements (e.g., going from rest to holding up all 5 fingers).

The relatively weaker performance of the LSTM and CNN-LSTM models may be attributed to their reliance on sequential feature modeling, which may not fully exploit the frequency and spatial patterns present in sEMG signals without additional engineered temporal structures.

Table 2: Model Type on Test Accuracy

Model Type	10 Gestures	25 Gestures	53 Gestures
CNN	0.914098277	0.823539641	0.802871778
CNN-LSTM	0.909577476	0.798339052	0.744392837
LSTM	0.894768554	0.784168564	0.774416893

4.3 Input Representation on Test Accuracy

The TD-FFT model achieved the highest accuracy across all classification tasks. The TD and FFT models trailed slightly behind, with similar accuracies, while the DWT model performed slightly worse. However, all representations demonstrated strong performance. These results are consistent with expectations, as the TD-FFT model combines both time-domain and frequency-domain information, effectively capturing key features leveraged individually by the TD and FFT models.

The comparatively lower performance of the DWT model suggests that this representation may not capture critical characteristics of sEMG signals as effectively as other representations. The high accuracies of both TD and FFT models indicate that informative features exist in both the

time and frequency domains, which aid in differentiating between different muscle activations, and, by extension, different hand gestures. Additionally, the FFT model consistently maintained a slight edge over the TD model, hinting at the CNN's ability to better exploit frequency-domain relationships–particularly for intensive movements associated with sharp peaks in frequency.

Table 3: Input Representation on Test Accuracy

Representation	10 Gestures	25 Gestures	53 Gestures
TD	0.914098277	0.823539641	0.802871778
DWT	0.901889148	0.817289286	0.799054874
FFT	0.918085416	0.822701068	0.805612053
TD-FFT	0.920268383	0.846415344	0.833529396

4.4 Attention on Test Accuracy

We found that adding channel and temporal attention to both the CNN and LSTM models improved accuracy on the 10-gesture classification task. For the LSTM model, this improvement persisted even as the number of gestures increased. This result aligns with expectations: by applying temporal attention after two bidirectional LSTM layers, the model is able to focus on the most distinctive temporal patterns for classification, having already captured complex temporal dependencies through the bidirectional processing.

However, as task difficulty increased, the CNN with attention began to perform worse than its baseline counterpart. We hypothesize that this decline stems from the type of attention applied. Specifically, while temporal attention benefits models focused on sequence modeling, 1-D CNNs may have benefited more from true spatial attention. Implementing spatial attention—potentially through 2-D CNNs—could allow the CNN to focus on the most informative spatial regions of the input sEMG feature maps, rather than treating all regions equally or emphasizing temporal relationships over spatial ones.

Table 4: Attention Implementation on Test Accuracy

Complexity	10 Gestures	25 Gestures	53 Gestures
CNN	0.914098277	0.823539641	0.802871778
CNN with attention	0.918444529	0.820230369	0.795832341
LSTM	0.894768554	0.784168564	0.774416893
LSTM with attention	0.909410189	0.818265543	0.796638935

5 Conclusion

This study revealed how model complexity and input domain representation affect hand gesture classification across tasks of increasing difficulty. The TD-FFT Multi-Branch model consistently achieved the highest test accuracy. Notably, simpler models' accuracy degraded with increasing task complexity while more complex models improved. Multi-layer attention was notably more effective in LSTM than CNN models, as we performed temporal attention, more beneficial towards sequentially based architectures such as the LSTM. We believe that if we were to apply spatial attention upon the CNN model, we would see improvements, aiding the model in selectively focusing on and processing specific features within visual space, such as peaks or dips within sEMG signals for specific gestures. Although the CNN-LSTM implementation performed well on simpler tasks, it underfit as task complexity grew. These findings highlight the importance of aligning model capacity with differing task demands. Future work could investigate how specific strategies for adjusting model complexity can be tailored to signal characteristics, movement patterns, or subject-specific physiological features including gender, age, height, weight, muscle size, and dominant hand/hand tested, to improve classification performance.

References

[1] Atzori M., Gijsberts A., Castellini C., Caputo B., Hager A.-G. M., Elsig S., et al. (2014). Ninapro - DB1 Dataset. https://ninapro.hevs.ch/instructions/DB1.html.

[2] Bao, T., Zaidi, S., Xie, S., Yang, P., & Zhang, Z.-Q. (2020, November 9). A CNN-LSTM hybrid model for wrist kinematics estimation using surface electromyography. White Rose Research Online. https://eprints.whiterose.ac.uk/167868/.

[3] Costa, M. V., Pereira, L. A., Oliveira, R. S., Pedro, R. E., Camata, T. V., Abrao, T., Brunetto, M. A. C., & Altimari, L. R. (2010). Fourier and wavelet spectral analysis of EMG signals in maximal constant load dynamic exercise. Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference. https://pubmed.ncbi.nlm.nih.gov/21096232/.

[4] Dantas, J. L., Camata, T. V., Brunetto, M. A. C., Moraes, A. C., Abrão, T., & Altimari, L. R. (2010). Fourier and wavelet spectral analysis of EMG signals in isometric and dynamic maximal effort exercise. Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference. https://pubmed.ncbi.nlm.nih.gov/21097104/.

[5] Discrete Wavelet Transform. (n.d.). https://www.st-andrews.ac.uk/ wjh/dataview/tutorials/dwt.html#dwt.

[6] Han B., Xing S., Wang J., Zhang Z. (2023). A new multichannel deep adaptive adversarial network for cross-domain fault diagnosis. https://www.researchgate.net/publication/368499454_A_new_multichannel_deep_adaptive_adversarial_network_for_cross-domain_fault_diagnosis.

[7] Heckbert, Paul (2014). Fast Fourier Transforms and the Fast Fourier Transform (FFT) Algorithm. https://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/2001/pub/www/notes/fourier/fourier.pdf.

[8] Hu J., Shen L., Albanie, Sun G., Wu E. (2017). Squeeze-and-Excitation Networks. https://arxiv.org/abs/1709.01507.

[9] Koch P. et al., "Regression of Hand Movements from sEMG Data with Recurrent Neural Networks," 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 2020, pp. 3783-3787, doi: 10.1109/EMBC44109.2020.9176278.

[10] Shi, X., Wang, T., Wang, L., Liu, H. and Yan, N. (2019). "Hybrid Convolutional Recurrent Neural Networks Outperform CNN and RNN in Task-state EEG Detection for Parkinson's Disease," 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 2019, pp. 939-944, doi: 10.1109/APSIPAASC47483.2019.9023190.

[11] Thorup, Katya (2021). Introduction to sEMG Signals. https://sites.tufts.edu/eeseniordesignhandbook/files/2021/05/Thorup_sEMGSignals.pdf.

[12] Too, J., Abdullah, A.R., Zawawi, T.N.S., & Musa, H. (2017). (PDF) classification of EMG Signal based on time domain and frequency domain features. https://www.researchgate.net/publication/329011789_Classification_of_EMG_Signal_Based_on_Time_Domain_and_Frequency__Domain_Features.

Appendix

A GitHub Repository

https://github.com/augusthao6/Model-Complexity-and-Input-Representations-on-EMG-Based-Hand-Gesture-Classification

B Hand Gesture Classifications

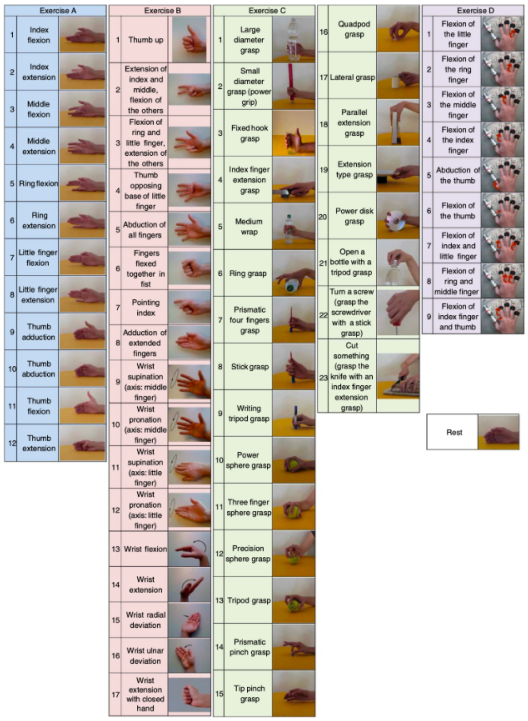

Figure 1: Hand Gesture Classifications

C CNN-LSTM Model Architecture

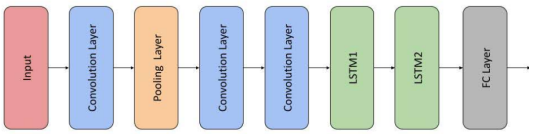

Figure 2: CNN-LSTM Model Architecture

D Discrete Wavelet Transform Calculation

Given an input window $W \in \mathbb{R}^{C \times T}$, where C is the number of channels and T is the number of time samples:

1. For each channel $c \in \{1, \ldots, C\}$:

$$\text{DWT}_c = \text{DWT}^{(L)}(W_c) \tag{1}$$

where:

- $W_c \in \mathbb{R}^T$ is the time series data for channel c,
- $\text{DWT}^{(L)}$ denotes a multilevel discrete wavelet transform up to level L,
- $\text{DWT}_c = \{A_L, D_L, D_{L-1}, \ldots, D_1\}$ are the approximation and detail coefficients.

2. Concatenate the coefficients to form a single feature vector for channel c:

$$f_c = \text{concat}(A_L, D_L, D_{L-1}, \ldots, D_1) \tag{2}$$

3. Stack the feature vectors from all channels to form the DWT feature representation for the window:

$$F = \{f_1, f_2, \ldots, f_C\} \in \mathbb{R}^{C \times \text{len}(f_c)} \tag{3}$$

E Fast Fourier Transform Calculation

Given an input window $W \in \mathbb{R}^{C \times T}$, where C is the number of channels and T is the number of time samples:

1. For each channel $c \in \{1, \ldots, C\}$:

$$F_c = |\mathcal{F}(W_c)| \tag{4}$$

where:

- $W_c \in \mathbb{R}^T$ is the time series data for channel c,
- $\mathcal{F}(W_c)$ denotes the real-valued Fast Fourier Transform (FFT) of W_c,
- $|\cdot|$ denotes taking the magnitude (absolute value) of the complex FFT output.

2. Stack the per-channel FFT magnitudes to form the FFT feature matrix:

$$F = \{F_1, F_2, \ldots, F_C\} \quad \text{where} \quad F \in \mathbb{R}^{C \times F'} \tag{5}$$

with F' being the number of frequency bins.

F Multi-Branch (TD-FFT) Model

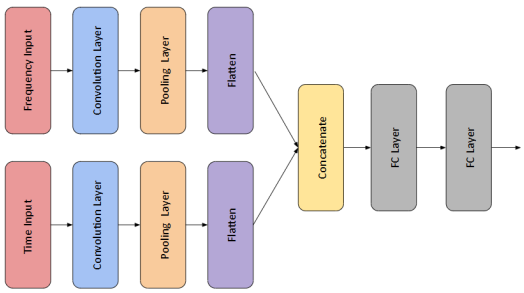

Figure 3: Multi-Branch (TD-FFT) Model

G Model Architectures

Model	Hyper-parameter	Value
middleCNN	1st convolution	Kernel size 3, Stride 1, Padding 1, 32 channels
	2nd convolution	Kernel size 3, Stride 1, Padding 1, 64 channels
	3rd convolution	Kernel size 3, Stride 1, Padding 1, 128 channels
	Max Pooling	Kernel size 2, Stride 2
	Global Avg Pool	AdaptiveAvgPool1d(1)
	Fully connected layers	128 -> 256 -> 256 -> num_classes
	Dropout	0.2, 0.3, 0.3
simpleCNN	1st convolution	Kernel size 3, Stride 1, Padding 1, 32 channels
	Max Pooling	Kernel size 2, Stride 2
	Global Avg Pool	AdaptiveAvgPool1d(1)
	Fully connected layers	32 -> 64 -> num_classes
	Dropout	0.2, 0.3
complexCNN	1st convolution	Kernel size 3, Stride 1, Padding 1, 32 channels
	2nd convolution	Kernel size 3, Stride 1, Padding 1, 64 channels
	3rd convolution	Kernel size 3, Stride 1, Padding 1, 128 channels
	4th convolution	Kernel size 3, Stride 1, Padding 1, 256 channels
	Max Pooling	Kernel size 2, Stride 2
	Global Avg Pool	AdaptiveAvgPool1d(1)
	Fully connected layers	256 -> 512 -> 512 -> num_classes
	Dropout	0.2, 0.3, 0.3
middleLSTM	LSTM 1	Input size: input_channel, Hidden size: 64, Bidirectional
	LSTM 2	Input size: 128, Hidden size: 64, Bidirectional
	Global Avg Pool	AdaptiveAvgPool1d(1)
	Fully connected layers	128 -> 128 -> num_classes
	Dropout	0.3, 0.3
middleCNNLSTM	1st convolution	Kernel size 3, Stride 1, Padding 1, 32 channels
	2nd convolution	Kernel size 3, Stride 1, Padding 1, 64 channels
	3rd convolution	Kernel size 3, Stride 1, Padding 1, 128 channels
	Max Pooling	Kernel size 2, Stride 2
	LSTM 1	Input size: 128, Hidden size: 64, Bidirectional
	LSTM 2	Input size: 128, Hidden size: 64, Bidirectional
	Fully connected layer	128 -> num_classes
	Dropout	0.3, 0.3

Table 5: Model Architectures

H Channel Attention Calculation for CNN and LSTM

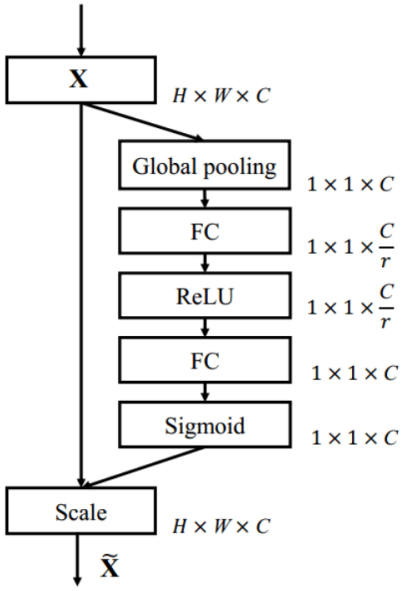

Figure 4: Channel Attention Calculation for CNN and LSTM

I Spatial Attention Calculation for CNN

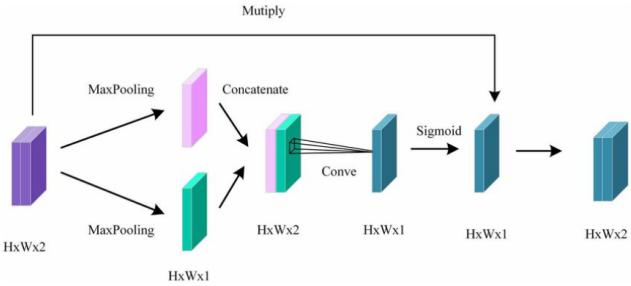

Figure 5: Spatial Attention Calculation for CNN

J Temporal Attention Calculation for LSTM

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Figure 6: Temporal Attention Calculation for LSTM