EMGCipher: Decoding Electromyography for Upper-limb Gesture Classification with Explainable AI for Resource Optimization

Hunmin Lee
Dept. of Computer Science
University of Minnesota
Minneapolis, USA
lee03915@umn.edu

Ming Jiang
Dept. of Computer Science
University of Minnesota
Minneapolis, USA
mjiang@umn.edu

Qi Zhao
Dept. of Computer Science
University of Minnesota
Minneapolis, USA
qzhao@umn.edu

***Abstract*—Assistive limb devices often employ surface electromyography (sEMG) and deep learning (DL) models for gesture classification. While DL models effectively classify diverse upper-limb gestures, their decision-making mechanisms often lack transparency. To address this, we introduce EMGCipher, an interpretable DL framework for upper-limb gesture classification using sEMG. It aims to bridge the gap between interpretability and performance by combining low-level sEMG feature representations with DL model-derived knowledge, quantitatively assessing the probabilistic significance of input sensors and features in gesture classification. Experiments on the Ninapro DB5 dataset demonstrate EMGCipher's effectiveness in sensor-wise and feature-wise interpretation, demonstrating its potential to optimize the usage of sensors and features for improved gesture classification performance and efficiency.**

***Index Terms*—explainable AI, electromyography, interpretability, rehabilitation engineering, resource optimization**

I. Introduction

The global market for rehabilitation and assistive device technologies is anticipated to quadruple by 2030 compared to 2022 [1], with a significant impact on the lives of individuals dealing with conditions like neurological disorders, paralysis, musculoskeletal impairments, aging, or limb loss [2]. Assistive limb devices and prosthetics often utilize surface electromyography (sEMG) and AI technologies, particularly deep learning (DL) models, for gesture classification [3]. The utilization of sEMG gesture classification spans diverse domains such as rehabilitation, healthcare, robotics, and human-computer interaction, facilitating the creation of assistive devices and interfaces governed by muscular signals [2], [4]. It pertains to the identification of limb gestures and motions through the analysis of sEMG signals. The initial stages involve collecting data from electrodes placed upon the skin's surface, followed by meticulous preprocessing to refine the acquired signals. Subsequently, window-based feature extraction techniques are employed to capture relevant patterns within the signal, utilizing DL models for their efficacy in handling non-linear relationships within these features to discern multiple gestures from the sEMG dataset. In particular, DL models have shown efficiency through diverse studies [5] due to their ability to learn complex spatial patterns and representations in sEMG signals, thereby enabling more accurate gesture recognition. Despite advancements in DL-based sEMG upper-limb gesture classification models [4], the decision-making process within these models remains opaque.

Explainable AI (XAI) plays a crucial role in biomedical engineering, facilitating a deeper understanding of AI decisions for personalized rehabilitation tool design, user safety, and trust. In 2021, the US FDA emphasized transparency and a patient-centered approach in their guidelines for future AI applications in the medical field, committing to address concerns about transparency, bias, and robustness. Similar principles have been adopted by other representative organizations globally, including the EU [6], China [7], and adherence to ISO standards (ISO 13485, 14971). Recent studies have explored XAI techniques in the context of sEMG gesture classification [8]–[10]. Gulati *et al.* [8] employed the GradCAM method, scrutinizing the layers within a convolutional neural network (CNN) model for gesture recognition. Similarly, Kang *et al.* [9] employed multiple XAI methodologies to analyze the significance of diverse signal modalities extracted from upper limb gestures, including sEMG, accelerometer, gyroscope, and magnetometer. Further, Lee *et al.* [10] investigated biological interpretation using attention mechanisms, integrating various physiological modalities into sEMG by merging finger joint angles for gesture classification. Despite these methodological advancements in extracting insights from DL models, a fundamental limitation persists: interpretative knowledge does not always directly correlate with performance outcomes, often resulting in subjective and uncertain interpretations.

To address this gap, this study introduces *EMGCipher*, an interpretable DL framework for upper-limb gesture classification using sEMG. EMGCipher aims to bridge the interpretability-performance gap by unraveling decision-making processes in conventional gesture classification models. It combines low-level feature representations with prior knowledge obtained from DL models, evaluating the probabilistic significance of electrodes and features during the feature representation learning phase. The framework's ef-

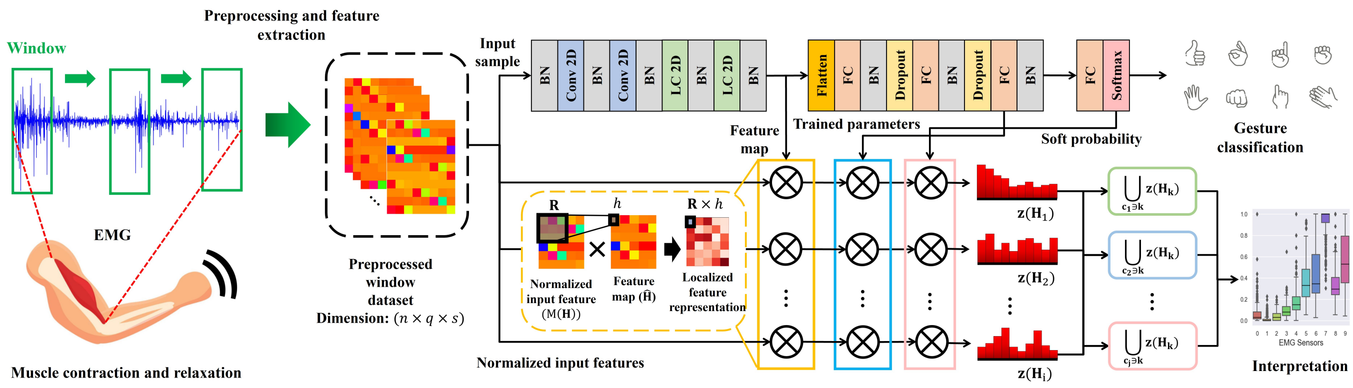

Fig. 1. Overview of the EMGCipher architecture. Normalized input features are integrated with convolutional-layer feature maps, penultimate-layer parameters, and soft output probabilities to interpret the contributions of sensors and feature channels to the gesture classification.

fectiveness is demonstrated through experiments on the Ninapro DB5 dataset, aligning interpretation with performance evaluation. The successful interpretation provides insights into how sEMG sensors and features impact gesture classification performance, enabling more efficient resource utilization by selecting crucial sensors and features. This, in turn, enhances the overall efficiency of the system design.

Our contributions can be outlined as follows:

- We introduce EMGCipher, a novel XAI framework aiming at closing the disparity between the interpretation and performance of sEMG gesture classification models.
- EMGCipher translates knowledge derived from a pre-trained CNN model and integrates multi-channel features of data samples to produce a comprehensive characterization of sensor and feature impacts on model performance.
- We conduct extensive experiments on a public sEMG gesture dataset to demonstrate sensor-wise and feature-wise interpretation and verify them with leave-one-out experimental results, indicating the potential of resource optimization with sensor and feature selection.

II. Methodology

A. Preprocessing and Feature Extraction

Typical sEMG gesture classification methods involve windowing and feature extraction, which transforms raw, high-dimensional signals into more manageable, informative features. The process reduces computational complexity, improves signal characteristics, reduces noise, and enhances generalization capabilities, resulting in accurate gesture classification [11]. In our experiments, data segmentation of 250-millisecond window size and 50-millisecond incremental step were applied to the dataset, following the preprocessing protocol in [12]. We extracted 11 temporal and frequency domain features [13], which include Mean Absolute Value (MAV), Waveform Length (WL), Willison Amplitude (WAMP), Zero Crossing (ZC), MAV Slope (MAVS), four Auto-Regressive (AR) coefficients, Median Frequency (MNF), and Power Spectral Ratio (PSR). Each computed feature was subsequently aligned per sensor, serving as an individual element within the input sample. The preprocessed input samples are two-dimensional arrays structured as $(s \times f)$, where s denotes the number of sensors and f represents the number of features.

B. Baseline Model Architecture for EMGCipher

EMGCipher is based on a pre-trained Convolutional Neural Network (CNN) architecture, with the foundational structure based on the Single View-CNN model (SV-CNN) [12]. It is specifically tailored for gesture recognition tasks using window-frame sEMG data, providing effective representation learning from input features. The network architecture, as depicted in Fig. 1, is organized as follows: Batch Normalization (BN) - Conv2D (64) - BN - Conv2D (64) - BN - Locally-Connected 2D (LC2D) (64) - BN - LC2D (64) - BN - Dropout (0.3) - Flatten - Fully-Connected (FC) (128) - Dropout (0.5) - FC (128) - Dropout (0.5) - FC (q) - FC (*classes*). Here, BN denotes the batch normalization layer, LC2D denotes the locally-connected 2D layer, and FC denotes the fully-connected layer. The values in parentheses indicate the layer units, and those within the Dropout layer specify the respective dropout rate. This network architecture serves as the foundation of our EMGCipher framework.

C. EMGCipher Framework

EMGCipher aims to bridge the gap between the inherent complexity of DL models and the need for transparent decision-making processes in the context of upper-limb gesture classification using sEMG data. By introducing a localized feature map $\hat{\mathbf{H}}$ and evaluating the contributions of sensors and features, EMGCipher provides a nuanced interpretation of the neural network's decision logic. Specifically, based on a pre-trained CNN network, we extract the input data $\mathbf{H}$, a localized feature map $\hat{\mathbf{H}}$, parameters at the penultimate layer $\mathbf{P}$, and the soft probability $\mathrm{O}(\mathbf{H})$ from the final layer. With these data, EMGCipher produces a two-dimensional array characterizing the relative influences of sensors and features on the model performance. This array provides a comprehensive understanding of the neural network's decision-making, shedding light on factors contributing to classifying specific gestures.

As shown in Fig. 1, for an input sample $\mathbf{H}$ with dimensions $(s \times f)$, we generate a cumulative feature map $\hat{\mathbf{H}}$ before the flattening layer. This is done through a series of feature extraction layers with layer-wise Rectified Linear Unit (ReLU) activation functions. Maintaining the size of $\mathbf{H}$ through a padding scheme enables the computation of individual sensor and feature contributions.

Next, each $\hat{h}$ in $\hat{\mathbf{H}}$ undergoes element-wise multiplication with the corresponding region of interest $\mathbf{R} \in \mathbf{H}$, and the resulting values are aggregated. To activate sensor contributions, we fix s while multiplying all elements of f, and to evaluate feature significance, we keep f constant while involving all s. This computation activates crucial spatial information across sensors and features, allowing the evaluation of specific compositions emphasized by pre-trained convolutional and locally connected layers when discerning various gesture types.

Afterward, the absolute value of the penultimate layer $\mathbf{P}$ (*s.t.* $\mathbf{P} \ni (q \times c)$) undergoes an inner product operation with the preceding result. For feature contribution evaluation, q is set as f, and for sensor contribution, q assumes the value of s. This multiplication combines knowledge representation patterns from the model, where the assignment of dimension q in $\mathbf{P}$ facilitates compatible matrix computations. The final value $\mathbf{z}(\mathbf{H})_q$ is obtained by incorporating the output $(\mathrm{O}(\mathbf{H}))$ from the final layer, representing class probabilities after the softmax function.

We summarize this process in equations (1) and (2), where $\mathrm{M}(\cdot)$ denotes min-max normalization, while u, i, and v, j are row and column indices for a two-dimensional matrix. The resulting $\mathbf{z}(\mathbf{H})_q$ is a singular vector characterizing the relative influences of each input dimension, adopting dimensions of $(s \times 1)$ or $(f \times 1)$. These measurements, calculated per sample, are combined into a two-dimensional array spanning dimensions $(n \times q)$, where n signifies the number of samples.

$$\mathbf{X}_q[i,j] = \begin{cases} \sum_{u=-k}^{k} \sum_{u=-k}^{k} \mathrm{M}(\mathbf{H})[u,v] \times \mathbf{R}[i-u,j], & \text{if } q = s \\ \sum_{u=-k}^{k} \sum_{u=-k}^{k} \mathrm{M}(\mathbf{H})[u,v] \times \mathbf{R}[i,j-u], & \text{if } q = f \end{cases} \tag{1}$$

$$\mathbf{z}(\mathbf{H})_q = (\mathbf{X}_q)^{\mathrm{T}} \cdot |\mathbf{P}_q| \cdot \mathrm{O}(\mathbf{H}) \begin{cases} \mathbf{z}(\mathbf{H})_q \in (s \times 1), & \text{if } q = s \\ \mathbf{z}(\mathbf{H})_q \in (f \times 1), & \text{if } q = f \end{cases} \tag{2}$$

III. Experiment and Result

A. Experimental Setting

Our experiments are conducted on Ninapro DB5 [14], a publicly available upper limb sEMG dataset. The dataset comprises sEMG recordings from 10 intact subjects, capturing sEMG signals through 16 electrodes. With 53 unique hand gestures, each repeated six times, we focused our experimentation on exercises B and C, narrowing down the number of gesture classes to 41. The inter-session gesture classification performance was evaluated for each subject, designating the 2nd and 5th repetitions as the test dataset and using the remaining repetitions for training. The models were trained for 200 epochs, using a batch size of 512, the Adam optimizer, and ReLU activation functions for all layers except the final layer, which employed the softmax function. Early stopping, returning the best test accuracy across epochs, and learning rate decay (50% decay in the 15th, 40th, 60th, and 80th epochs, starting from 0.002) were used.

B. Gesture Classification Result

The training process of the SV-CNN gesture classification model serves as the foundation to establish a benchmark for our study. The successful completion of model training and the achievement of classification performance are crucial for reliable interpretation. To analyze the correlation between EMGCipher's interpretation and model performance, we evaluate it using a leave-one-out methodology. This involves removing data related to a specific sensor or feature from the input data and assessing the classification performance across subjects using the modified input. For example, when a sensor is removed, the input data's dimensions will be reduced to $(s-1, f)$. Similarly, removing a particular feature alters the input dimensions to $(s, f-1)$.

The results in Table I provide insights into the test accuracy of gesture classification by the SV-CNN model, comparing the leave-one-out results across sensor and feature dimensions and suggesting the quantifiable impact of each sensor or feature on performance accuracy. With all sensors and features (see the last rows 'None'), we observe consistent performance across subjects with test accuracies of 87.5% $(q = s)$ and 85.7% $(q = f)$, accompanied by standard deviations of 2.2 and 2.5, respectively. These results indicate the model's effectiveness in accurately classifying gestures across different sessions, showcasing its resilience and credibility in inter-session gesture classification. The minimal standard deviations suggest consistent and reliable performance, contributing to the overall reliability of the SV-CNN model in handling diverse upper-limb gestures across various subjects and sessions.

The leave-one-feature-out experiment revealed substantial variations in classification accuracy, particularly when excluding specific features. Notably, the absence of feature 1 resulted in a significant 30% decline, yielding a 52% accuracy, contrasting sharply with models excluding other features. Features 2 and 3 exhibited marginal variations, with approximately a 3.5% decrease and increase, respectively, compared to models without features 4 to 11. Intriguingly, the removal of feature 3 led to a discernible performance enhancement, surpassing the model's average by 2.2% compared to training with all features. This highlights a detrimental contribution of feature 3 to overall accuracy. Beyond these three features, most removed features showed a minor performance degradation of around 2%, emphasizing the significant impact of individual features on the model's accuracy. The findings reveal specific features that significantly influence classification performance, suggesting the importance of feature selection for optimizing models' effectiveness and efficiency.

TABLE I
LEAVE-ONE-OUT PERFORMANCE EVALUATION RESULTS, WHERE THE VALUE INSIDE THE PARENTHESIS DENOTES THE STANDARD DEVIATION.

Leave-one-feature-out performance											
Feature (f)	**Sub. 1**	**Sub. 2**	**Sub. 3**	**Sub. 4**	**Sub. 5**	**Sub. 6**	**Sub. 7**	**Sub. 8**	**Sub. 9**	**Sub. 10**	**Average**
MAV (f1)	55.6%	56.2%	56.9%	42.7%	50.3%	48.9%	50.8%	51.7%	61.1%	51.7%	52.6% (4.6)
WL (f2)	83.6%	83.3%	89.6%	79.3%	82.1%	83.1%	79.1%	81.0%	87.0%	81.4%	82.9% (3.0)
WAMP (f3)	**89.1%**	**89.7%**	**92.6%**	**88.0%**	**91.1%**	**89.1%**	**87.6%**	**89.1%**	**90.9%**	**89.5%**	**89.7% (1.4)**
ZC (f4)	86.7%	85.6%	90.3%	82.5%	85.0%	84.9%	82.2%	83.7%	88.4%	85.0%	85.4% (2.3)
MAVS (f5)	86.1%	85.3%	90.0%	81.1%	83.3%	84.3%	81.1%	83.6%	88.6%	84.8%	84.8% (2.6)
AR1 (f6)	86.5%	86.3%	90.7%	82.4%	84.2%	84.9%	81.9%	84.2%	89.1%	85.4%	85.6% (2.5)
AR2 (f7)	86.6%	85.5%	91.3%	83.3%	84.6%	84.9%	82.8%	84.6%	88.9%	85.5%	85.8% (2.3)
AR3 (f8)	86.2%	86.9%	91.6%	82.6%	85.0%	85.1%	82.3%	84.5%	88.2%	85.5%	85.8% (2.5)
AR4 (f9)	87.1%	86.2%	91.0%	81.8%	85.5%	85.5%	82.3%	84.0%	88.7%	84.7%	85.7% (2.5)
MNF (f10)	87.4%	86.0%	91.2%	83.1%	85.0%	84.4%	82.0%	84.3%	88.5%	85.5%	85.7% (2.4)
PSR (f11)	86.5%	85.6%	91.7%	83.0%	85.2%	85.2%	81.7%	83.7%	89.3%	84.6%	85.6% (2.7)
None	88.0%	87.1%	92.4%	85.4%	86.8%	87.2%	84.3%	86.3%	90.2%	87.1%	87.5% (2.2)
Leave-one-sensor-out performance											
Sensor (s)	**Sub. 1**	**Sub. 2**	**Sub. 3**	**Sub. 4**	**Sub. 5**	**Sub. 6**	**Sub. 7**	**Sub. 8**	**Sub. 9**	**Sub. 10**	**Average**
s1	86.2%	84.5%	90.3%	82.3%	84.3%	83.2%	81.8%	83.1%	87.4%	84.1%	84.7% (2.3)
s2	86.4%	85.0%	90.4%	82.1%	84.5%	84.6%	81.9%	82.5%	88.1%	84.4%	85.0% (2.4)
s3	86.0%	85.4%	90.8%	81.5%	84.1%	84.3%	82.0%	83.9%	88.3%	84.2%	85.1% (2.5)
s4	85.9%	85.6%	**91.3%**	82.7%	84.9%	83.5%	81.9%	83.4%	**89.3%**	85.1%	85.4% (2.7)
s5	**87.2%**	85.3%	90.5%	82.5%	84.7%	83.6%	81.7%	83.0%	88.5%	85.0%	85.2% (2.5)
s6	86.2%	85.7%	89.7%	82.7%	**86.0%**	84.9%	81.9%	**84.6%**	88.4%	85.3%	85.5% (2.1)
s7	86.2%	85.7%	90.5%	81.3%	84.5%	83.6%	81.5%	83.8%	88.9%	83.9%	85.0% (2.7)
s8	86.2%	**86.1%**	89.5%	81.8%	83.1%	85.0%	80.9%	82.8%	87.6%	83.6%	84.7% (2.4)
s9	86.1%	84.9%	90.0%	81.1%	83.1%	84.1%	81.4%	83.2%	88.0%	84.2%	84.6% (2.5)
s10	85.9%	85.0%	90.4%	81.8%	83.8%	84.8%	80.4%	82.3%	87.3%	84.5%	84.6% (2.6)
s11	85.9%	86.0%	90.7%	81.5%	85.2%	85.4%	81.6%	83.4%	88.6%	84.7%	85.3% (2.6)
s12	86.0%	85.9%	90.4%	**83.0%**	84.5%	84.2%	82.2%	83.7%	89.2%	83.3%	85.2% (2.4)
s13	85.5%	85.6%	90.8%	82.6%	84.2%	84.8%	**82.4%**	83.2%	87.8%	84.3%	85.1% (2.3)
s14	86.0%	85.5%	90.6%	82.6%	84.1%	85.1%	80.7%	83.3%	88.5%	**85.5%**	85.2% (2.6)
s15	86.2%	85.7%	90.7%	82.1%	84.3%	84.5%	82.1%	83.9%	88.6%	84.7%	85.3% (2.4)
s16	85.7%	85.5%	90.5%	81.8%	84.3%	**85.7%**	80.8%	83.7%	87.8%	83.1%	84.9% (2.6)
None	87.0%	85.9%	90.8%	82.5%	84.9%	85.1%	82.3%	84.0%	88.7%	85.4%	**85.7% (2.5)**

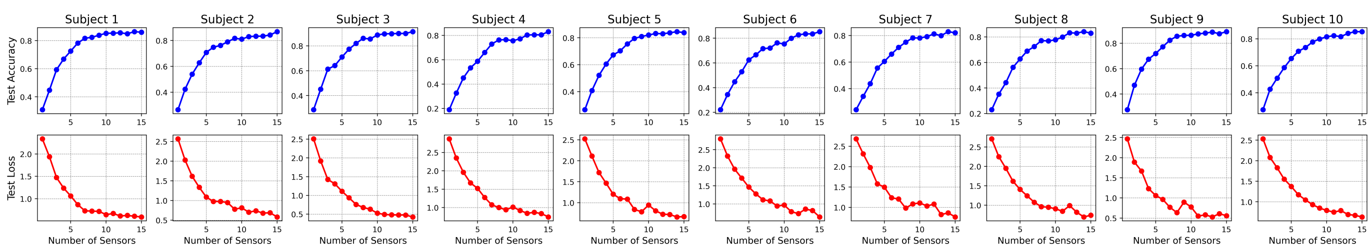

Fig. 2. Graphs representing test accuracy (top row) and test loss (bottom row) varying by a randomly selected number of sensors aligned with each subject.

Contrastingly, the leave-one-sensor-out approach yielded classification accuracies with minimal disparities, suggesting a limited impact of individual sensor absence on performance measures. To further explore the influence of electrode quantity on the model, a random reduction in the sequential number of sensors was executed, allowing the model to adapt and assess classification performance. As depicted in Fig. 2, optimal accuracy and loss performance were consistently observed across all subjects. Notably, consistent patterns emerged, indicating prominent performance deviations when the sensor count ranged between 7 to 10. This observation implies a potential equilibrium between model performance and computational complexity, shedding light on the relationship across sensor quantity, model performance, and computational efficiency, offering valuable insights for optimizing the selection of sensors in practical applications.

C. Model Interpretation

To verify the effectiveness of the proposed EMGCipher framework, in Fig. 3, we visualize the feature and sensor activation levels through box charts in the upper and lower rows, respectively, and compare the interpretations with those insights we observed from the leave-one-out experiments.

As shown in the first row of Fig. 3, certain feature attributes such as Mean Absolute Value (MAV), Waveform Length (WL), and Willison Amplitude (WAMP) consistently exhibit higher significance than other features in the feature activation analysis across all subjects. Particularly, MAV emerges as the most prominent feature across most subjects. The observed consistency in the significance of MAV, WL, and WAMP suggests their crucial role in gesture classification. Such feature-wise interpretation sheds light on the specific features that consistently contribute to the model's decision-making

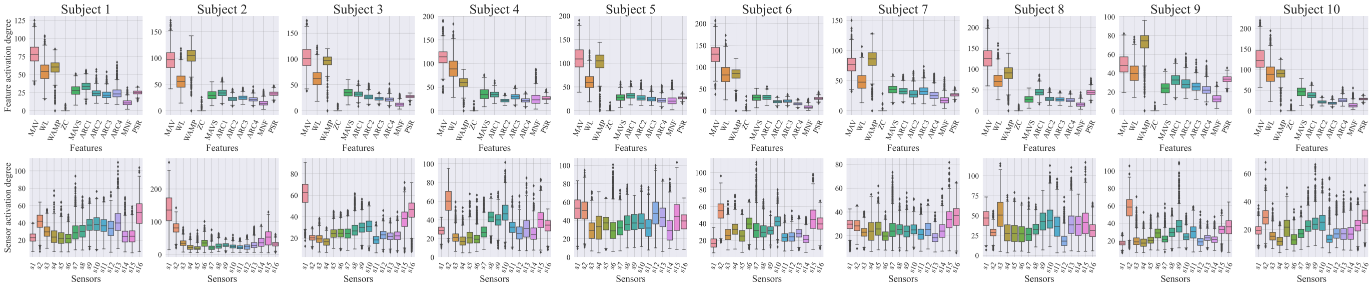

Fig. 3. Interpretation results across all subjects. The first row visualizes feature importance, consistently suggesting significant activation levels within the first three features (*i.e.*, MAV, WL, WAMP). The second row displays the importance of various sensors, showing no specific patterns or consistent trends.

process. The interpretation results consistently align with the leave-one-feature-out experimental results, where excluding these features significantly impacted classification accuracy as well as computational efficiency.

The second row of Fig. 3 shows sensor-wise interpretation results, which reveals a degree of inconsistency of sensor contributions across subjects. The activation levels varied notably among individuals, with specific sensors displaying heightened significance in different cases. For example, sensor 1 showed substantial activation in subjects 2 and 3, sensor 2 stood out in subject 9, and sensor 15 demonstrated significant activation in subject 7. This variability suggests the subject-specific nature of sensor contributions, highlighting the need for personalized considerations in interpreting sensor-level influences on the neural network's decision-making process. These findings also agree with the leave-one-sensor-out experiment results. In the leave-one-out experiments, the absence of individual sensors had minimal impact on the overall classification accuracy, indicating a level of redundancy or robustness in the model to the removal of individual sensors. Here, the sensor-wise interpretation reveals the reason behind such observation, which is subject-specific variations. Such interpretation supplements the general leave-one-out experiments, offering more in-depth analyses and insights on optimizing sensor resource usage.

IV. Conclusion

We have introduced EMGCipher, a novel interpretation framework designed to enhance the transparency and interpretability of DL models for classifying upper-limb gestures using sEMG data. Through extensive analysis on a publicly available sEMG upper-limb gesture dataset, our research establishes a coherent relationship between model interpretation (*e.g.*, contributions of individual features and sensors) and the resulting model performance. This alignment serves to bridge the gap in understanding the workings of black-box AI models in sEMG gesture classification, ultimately contributing to improvements in accuracy and resource efficiency. The interpretability achieved by EMGCipher holds significant implications in the domain of sEMG-based gesture classification. By shedding light on the decision-making processes of AI models, EMGCipher paves the way for more informed and effective advancements in healthcare applications.

References

[1] Verified Market Research. (Apr. 2023) Global robotic rehabilitation and assistive technologies market size by type (assistive robots, rehabilitation robots), by portability (standalone, mobile), by application (post-surgery, cognitive and motor skill therapy), by geographic scope and forecast. Report ID: 40083.

[2] R. J. Varghese, D. Freer, F. Deligianni, J. Liu, G.-Z. Yang, and R. Tong, "Wearable robotics for upper-limb rehabilitation and assistance: A review of the state-of-the-art challenges and future research," *Wearable technology in medicine and health care*, pp. 23–69, 2018.

[3] A. T. Nguyen, M. W. Drealan, D. K. Luu, M. Jiang, J. Xu, J. Cheng, Q. Zhao, E. W. Keefer, and Z. Yang, "A portable, self-contained neuroprosthetic hand with deep learning-based finger control," *Journal of neural engineering*, vol. 18, no. 5, p. 056051, 2021.

[4] D. Xiong, D. Zhang, X. Zhao, and Y. Zhao, "Deep learning for emg-based human-machine interaction: A review," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 3, pp. 512–533, 2021.

[5] M. C. Tosin, J. C. Machado, and A. Balbinot, "Semg-based upper limb movement classifier: Current scenario and upcoming challenges," *Journal of Artificial Intelligence Research*, vol. 75, pp. 83–127, 2022.

[6] K. Lekadir, G. Quaglio, A. T. Garmendia, and C. Gallin, "Artificial intelligence in healthcare-applications, risks, and ethical and societal impacts," *European Parliament*, 2022.

[7] Center for Medical Device Evaluation of NMPA, "Guidelines for the classification and definition of ai-based samd," July 2021, notice No. 47.

[8] P. Gulati, Q. Hu, and S. F. Atashzar, "Toward deep generalization of peripheral emg-based human-robot interfacing: A hybrid explainable solution for neurorobotic systems," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2650–2657, 2021.

[9] P. Kang, J. Li, S. Jiang, and P. B. Shull, "Reduce system redundancy and optimize sensor disposition for emg–imu multimodal fusion human–machine interfaces with xai," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–9, 2022.

[10] H. Lee, D. Kim, and Y.-L. Park, "Explainable deep learning model for emg-based finger angle estimation using attention," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 1877–1886, 2022.

[11] A. Phinyomark, R. N. Khushaba, and E. Scheme, "Feature extraction and selection for myoelectric control based on wearable emg sensors," *Sensors*, vol. 18, no. 5, p. 1615, 2018.

[12] W. Wei, Q. Dai, Y. Wong, Y. Hu, M. Kankanhalli, and W. Geng, "Surface-electromyography-based gesture recognition by multi-view deep learning," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 10, pp. 2964–2973, 2019.

[13] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for emg signal classification," *Expert systems with applications*, vol. 39, no. 8, pp. 7420–7431, 2012.

[14] S. Pizzolato, L. Tagliapietra, M. Cognolato, M. Reggiani, H. Müller, and M. Atzori, "Comparison of six electromyography acquisition setups on hand movement classification tasks," *PloS one*, vol. 12, no. 10, p. e0186132, 2017.