Chapter 3

Enhancing sEMG-Based Static Hand Gesture Recognition Using Machine Learning Approach

This chapter primarily investigates the potential of surface electromyography (sEMG) for accurately recognizing static hand gestures. Hand gesture recognition is a crucial aspect of human-computer interaction, enabling intuitive and natural communication between humans and machines. Static hand gesture recognition, in particular, is essential for various applications, including sign language interpretation [39], virtual reality [199], robotics control [200], and assistive technologies for individuals with disabilities [201]. Accurate and reliable static gesture recognition can significantly enhance the usability and accessibility of these systems.

Our focus is on static hand gestures, specifically on American Sign Language (ASL) finger-spelling gestures [67, 68, 73]. ASL is widely used by the deaf and hard-of-hearing community, and accurate recognition of ASL gestures can facilitate more effective communication [41].

We propose an innovative ensemble feature selection technique to improve the precision of existing sEMG-based recognition models. This technique incorporates a feature combiner designed to select the most representative and non-redundant features for the classification model. By optimizing the feature selection process, we aim to address challenges such as signal interference, individual variability, and high-dimensional feature vectors, which can affect the performance of recognition systems [19].

Section 3.1 introduces the chapter, presenting the problem statement, outlining the proposed solution, and the research questions addressed. Section 3.1.2 outlines the main contributions of this chapter. The experimental setup is given in Section 3.2, which covers the materials and methods employed, data preprocessing steps, and the introduction of a novel ensemble feature selection technique. Section 3.3 presents the experimental results and discussion. Finally, Section 3.4 summarizes the key findings and conclusions.

3.1 Introduction

Sign language is an essential communication tool for individuals with disabilities, enabling effective interaction with their surroundings. American Sign Language (ASL) is one of the most widely used sign languages, consisting of various gestures to convey human expression [41]. Most of these gestures, especially fingerspelling, are produced with a single hand [202]. Fingerspelling is a crucial aspect of sign language communication, allowing users to spell out words that cannot be easily represented by standard signs. ASL, like any other natural language, has its own syntax and grammatical rules [203]. Accurate recognition of these static hand gestures is essential for effective communication and has significant implications for enhancing accessibility and usability in various applications.

Most current Sign Language Recognition Systems(SLRS) rely on computer vision techniques, which use cameras and image processing algorithms to interpret

gestures [4]. While these methods have shown promise, they often face significant challenges, including dependency on consistent lighting conditions, background clutter, and occlusions [5]. These limitations can compromise the accuracy and reliability of image-based gesture recognition systems in real-world environments.

To address the limitations of computer vision-based methods, surface electromyography (sEMG) has emerged as a promising alternative for hand gesture recognition [45]. sEMG captures the electrical activity produced by muscle contractions, offering a robust solution to the challenges faced by image-based systems. Unlike computer vision techniques, sEMG is unaffected by lighting conditions and can directly reflect the user's muscle activities and intentions. This makes it effective for precise and reliable gesture recognition even in unfavorable lighting conditions.

This study focuses on the recognition of static ASL finger-spelling gestures using sEMG. To enhance the performance of existing sEMG-based recognition models, we propose an innovative ensemble feature selection technique. This technique integrates multiple feature selection methods to identify the most representative and non-redundant features for classification, addressing challenges such as signal interference, individual variability, and high-dimensional feature vectors.

Given the scarcity of publicly available sensor-based datasets for Sign languages, we have compiled our own datasets, ASL-10 and ASL-24 (details in Section 2.3.3), to support this research. These datasets provide a comprehensive collection of sEMG signals corresponding to various ASL gestures, enabling rigorous evaluation and validation of our proposed techniques. The datasets were recorded using wireless sEMG sensors to capture predefined hand gestures, focusing on the 24 manual alphabets (ASL-24) and ten digits (ASL-10) of American Sign Language (ASL). The data were preprocessed, and approximately 450 well-established features were extracted from each sEMG channel. We applied an ensemble feature selection approach, combining four diverse filter-based methods: ANOVA [191, 192], Chi-square [193], Mutual Information [194, 195], and ReliefF [189]. A novel feature combiner, leveraging feature

feature and feature-class correlation thresholds, was used to integrate the selected features. This process resulted in a reduced and highly representative feature subset, which was subsequently used for classifying ASL gestures.

Ensemble feature selection has been successfully applied to various fields, including network traffic analysis, biomedical signal processing, and pattern recognition, to optimize large feature spaces [204–206]. Using ensemble feature selection helps overcome the biases and local optima associated with individual feature selection methods [207]. By integrating diverse feature selection models, ensemble methods achieve improved performance in classifying a wide range of applications [205, 206]. However, arbitrarily increasing the number of individual feature selection models in an ensemble does not necessarily yield better results. Wang et al. [208] observed that ensembles comprising a few rankers perform comparably or better than those made from multiple or higher numbers of feature ranking techniques.

Olsson and Oard [209] applied ensemble selection to text classification, achieving improved precision and F1 scores. Similarly, Yu et al. [210] reported increased performance using a genetic algorithm-based ensemble method. Zang et al. proposed a cost-sensitive feature selection model using multi-objective particle swarm optimization to maximize classification performance while minimizing feature-related costs. Miften et al. [211] proposed an ensemble feature selection technique to identify the most influential features for classifying six hand grasps using sEMG signals. The authors combined three feature selection methods to create an ensemble and used a ranking combination approach to obtain an integrated feature set. Additionally, the Fisher discriminant ratio was applied to determine the threshold value, which helped identify the most significant feature subset for classification. The proposed ensemble method achieved an average classification rate of 98.5% for five subjects, outperforming many previous research works and yielding about 7% higher classification results than using a single feature selection model.

Our proposed work on ensemble feature selection is inspired by the ensemble

technique used in Miften et al. [211], as described earlier in this section. The authors applied a ranking-based combinatorial approach to the overall feature set of three different feature selection techniques. However, the feature ranking methods used are not considered efficient in managing redundant variables [212] [213] [214], which can lead to suboptimal feature subsets, impacting model performance and computational efficiency. This limitation has motivated us to develop a more effective and robust ensemble feature selection method.

3.1.1 Problem Definition and overview of the solution

This chapter explores the development of a reliable and accurate sEMG-based static hand gesture recognition system, with a specific focus on American Sign Language (ASL) finger-spelling gestures. The overall goal is to address several key challenges in the field of human-computer interaction, particularly in the context of gesture recognition using sEMG signals. We propose an innovative ensemble feature selection technique aimed at improving the precision of existing sEMG-based recognition models.

3.1.1.1 Problem Statement

Let $X \in \mathbb{R}^{N \times T}$ be the raw sEMG data matrix, where N is the number of sensors and T is the number of time samples. The aim is to develop an efficient pipeline for recognizing static hand gestures, specifically American Sign Language (ASL) finger-spelling gestures, using sEMG signals. The primary objective is to enhance feature selection and thus improve classification accuracy through a proposed ensemble feature selection technique.

3.1.1.2 Overview of the Solution

To achieve the objective, the following tasks are organized as a pipeline:

1. **Feature Extraction**: The initial step in the pipeline involves converting the raw sEMG data X into a set of meaningful features suitable for classification.

This transformation is crucial because raw sEMG signals are challenging to interpret directly. The feature extraction function, f_{extract}, processes the raw data to generate a feature vector $\Phi(X) \in \mathbb{R}^M$, where M represents the number of features.

$$\Phi(X) = f_{\text{extract}}(X)$$

This step is essential for converting complex raw data into a more manageable and informative representation.

2. **Feature Selection**: The goal here is to select a subset of features that maximizes classification accuracy. Our ensemble approach applies multiple filter-based algorithms to the extracted features. Each algorithm selects a set of relevant features:

$$\Phi_{\beta_i}(X) = \text{select}_i(\Phi(X))$$

These sets are combined using a feature combiner, ensuring the final set of selected features is representative and non-redundant:

$$\Phi_\beta(X) = \text{combiner}(\{\Phi_{\beta_1}(X), \Phi_{\beta_2}(X), \ldots, \Phi_{\beta_k}(X)\})$$

The combined feature vector $\Phi_\beta(X)$ integrates the most informative features from various methods, enhancing the model's robustness and accuracy. The overall ensemble feature selection function is:

$$\Phi_\beta(X) = \text{select}_{\text{ensemble}}(\Phi(X), \beta)$$

This step improves the model's performance and efficiency by focusing on the most relevant features.

3. **Classification Model**: With the selected features, the next step is to train a classifier to predict the gesture labels. The classifier function h, parameterized by θ, is trained on the selected feature set $\Phi_\beta(X)$ to predict the gesture label

$\hat{Y} \in \{1, 2, \ldots, C\}$.

$$\hat{Y} = h(\Phi_\beta(X); \theta)$$

This task involves learning from the selected features to accurately recognize different static hand gestures.

4. **Loss Function**: During training, the model's performance is quantified using a loss function. The classification loss L measures the discrepancy between the predicted labels $\hat{Y}$ and the true labels Y. Minimizing this loss is essential for improving the model's accuracy. The loss function is defined as:

$$L(\theta, \beta) = \frac{1}{N} \sum_{i=1}^{N} \ell(h(\Phi_\beta(X_i); \theta), Y_i)$$

where ℓ represents the loss function (e.g., cross-entropy loss). This step guides the optimization process, ensuring that the model's predictions become more accurate over time.

5. **Constraints**: To ensure the model's effectiveness, it must satisfy an accuracy constraint. Specifically, the classification accuracy A must meet or exceed a predefined threshold $A_{\min}$ for the selected subsets S, as determined by the proposed ensemble feature selection approach and associated parameters θ.

$$A(S, \theta) \geq A_{\min}$$

This step guarantees that the model's performance is not only optimized but also meets predefined standards.

Hence, when framed as an optimization problem, the aim is to minimize the classification loss:

$$\min_{\beta, \theta} L(\theta, \beta)$$

subject to:

$$A(S, \theta) \geq A_{\min}$$

Figure 3.1 summarized the generalized pipeline proposed for the handwritten character recognition task.

To validate the effectiveness of the proposed model, we formulated the following research questions(RQs):

1. RQ1: *How can an ensemble feature selection technique be designed to enhance the accuracy and efficiency of feature selection for sEMG-based static hand gesture recognition? (Answered in Section 3.2.7)*

2. RQ2: *What level of classification accuracy and robustness can be achieved by the proposed machine learning pipeline for sEMG-based static hand gesture recognition, and how does it compare with existing methods? (Answered in Section 3.3.1)*

3. RQ3: *How scalable is the proposed pipeline to larger datasets for sEMG-based static hand gesture recognitions?(Answered in Section 3.3.2.3)*

4. RQ4: *What is the feasibility of implementing the proposed machine learning pipeline in real-time sEMG-based static hand gesture recognition applications, considering factors such as response time and processing efficiency?(Answered in Section 3.3.5)*

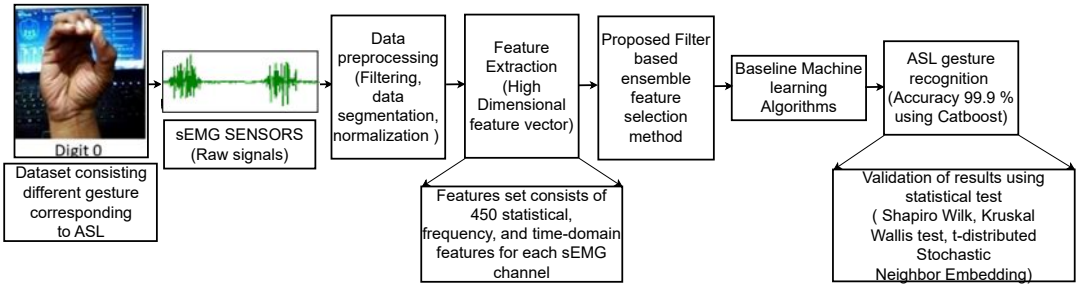

FIGURE 3.1: Generalized pipeline proposed for ASL gesture recognition task

3.1.2 Major contribution of the work

The primary contributions of this chapter are detailed as follows:

1. Two datasets, ASL-10 and ASL-24, were systematically acquired by recording surface electromyography (sEMG) signals corresponding to American Sign Language (ASL) gestures.

2. A comprehensive evaluation of approximately 450 features for each sEMG channel was conducted. A novel ensemble feature selection technique was introduced, which combined multiple feature selection methods to identify the most representative feature subsets for the classification of ASL hand gestures. This innovative approach significantly enhanced the feature selection process, ensuring optimal performance and accuracy.

3. A robust machine learning pipeline was developed for ASL gesture recognition, optimizing the number of sEMG sensors. This pipeline demonstrated exceptional average classification accuracies of 99.99% for ASL-10 using two sEMG channels and 99.91% for ASL-24 using eight sEMG channels.

4. The proposed machine learning pipeline was validated using the benchmark Ninapro dataset (Dataset 5, Exercise A), which includes 12 similar gestures. The pipeline achieved classification accuracy on par with state-of-the-art methods, thereby demonstrating its efficacy.

5. Explainable AI techniques were employed to determine feature importance, enhancing transparency and interpretability. This provides valuable insights into which features most significantly impact model performance while performing sEMG-based classification tasks.

3.2 Materials and Methods

This section provides an overview of the proposed machine learning pipeline for the ASL recognition task, including data collection protocols and a dataset description. It outlines data preprocessing methods to address inconsistencies, noise, and

artifacts in raw sEMG signals and details the extraction of relevant features from each window segment.

Additionally, it details the ensemble feature selection approach used to manage the high-dimensional feature vector. This method combines four filter-based techniques followed by a greedy search to select feature subsets based on ranking and importance heuristics. A novel feature combiner then merges the selected subsets using feature-feature and feature-class correlation thresholds.

3.2.1 Subjects and experimental protocol

The experimental dataset was collected from 20 subjects (15 males and 5 females) aged 22-28 years. Ethical approval was obtained from the Institute of Medical Science, Banaras Hindu University, Varanasi, prior to collecting surface EMG signals. A Myo Armband with an eight-channel wireless sEMG sensor from Thalmic Labs was used to record various American Sign Language (ASL) gestures, including ten digits (0-9) and 24 manual alphabets. Subjects wore the armband on the lower forearm near the elbow, targeting the flexor, extensor, and brachioradialis muscles, which are responsible for wrist and finger movements. Hairs were removed, and the skin was cleaned with alcohol. Sensor placement was maintained at a fixed distance from the elbow for consistency. The experimental setup for acquiring EMG data is shown in Figure 3.2. While the raw sEMG signals for ASL-10 are shown in Figure 3.3

3.2.2 Data Acquisition and Sensor Placement

For each ASL gesture, 5 seconds of continuous EMG data were collected from all participants at a sampling frequency of 200 Hz. The Myo data capture software was used to acquire the EMG data on a PC, while the Myo web-based interface monitored the sEMG signals. To minimize muscle fatigue during data collection, a fixed protocol (Figure 3.4) was followed. Data collection occurred in multiple trials

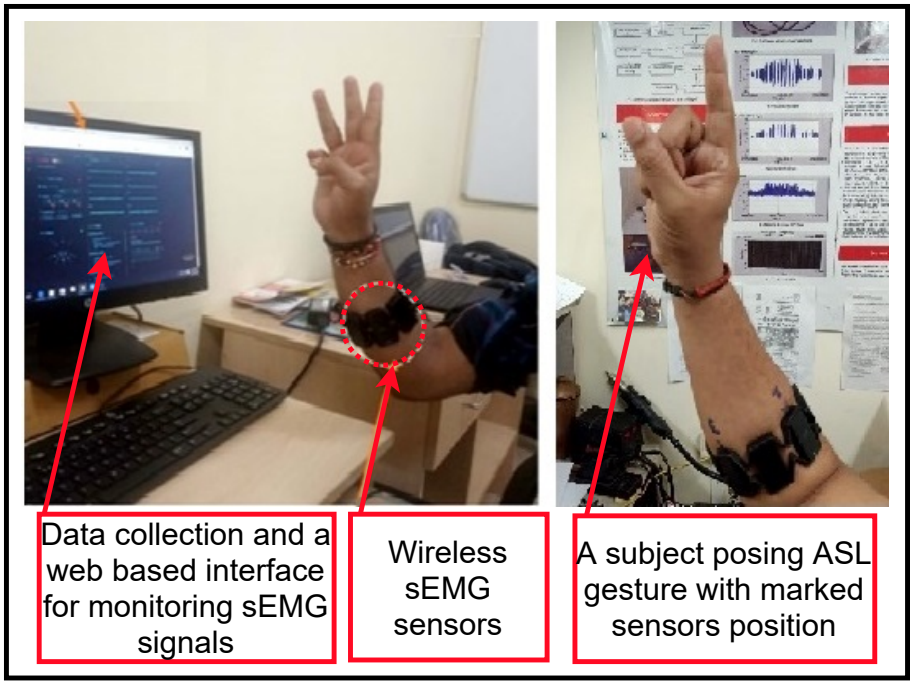

FIGURE 3.2: Data collection setup

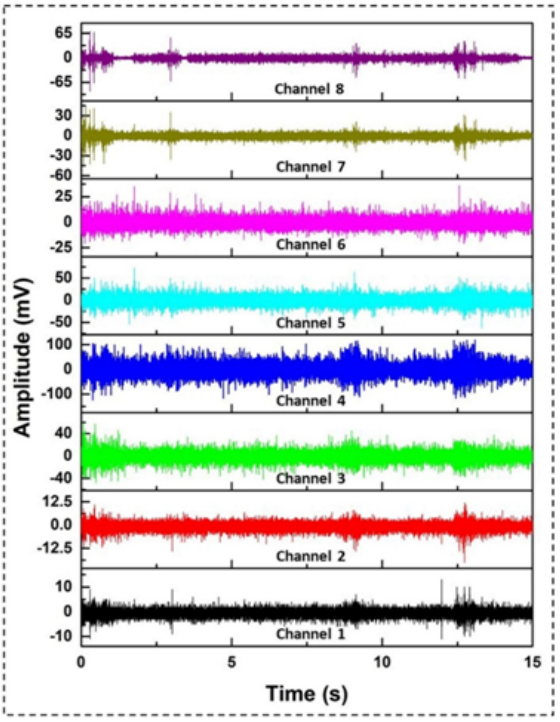

FIGURE 3.3: Raw sEMG signals patterns of a subject for digit 9 of ASL

and sessions, with each session consisting of several 5-second trials for each ASL sign. Appropriate gaps were maintained between successive sessions to reduce fatigue.

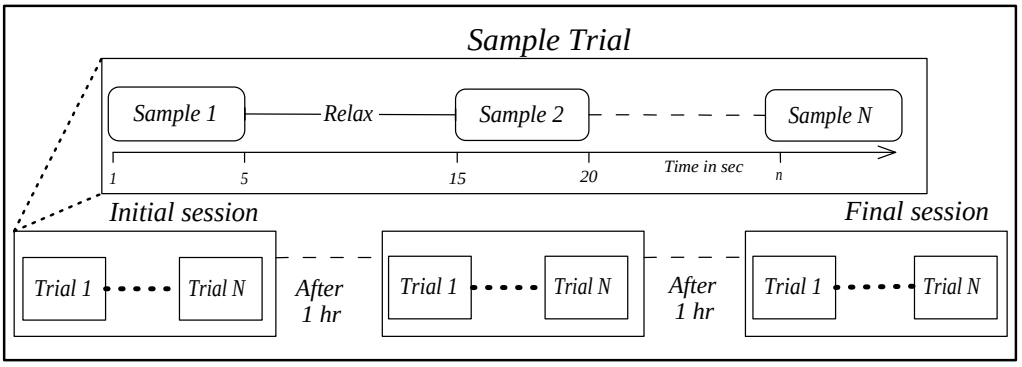

FIGURE 3.4: Data collection protocol

The accuracy of EMG signals depends significantly on sensor placement. The eight sensors of the Myo Armband were identified with marks from 1-8. Efforts were made to place sensor 1 (with a blue light) on the Extensor carpi ulnaris muscle. Additionally, an sEMG sensor was positioned on the Extensor carpi radialis longus muscle by adjusting the armband length. These two muscles were initially identified and marked. The respective sEMG channels were annotated in the dataset based on these placements, following the methodology of Pizzolato et al. [215].

3.2.3 Datasets

The study utilizes three distinct datasets to evaluate the proposed Ensemble Feature Selection algorithm:

- Ninapro reference dataset (Detailed description is provided in Section 2.3.4
- ASL manual alphabet dataset (ASL-24) : (Detailed description is provided in Section 2.3.7.
- ASL digit dataset (ASL-10) (Detailed description is provided in Section 2.3.6

3.2.3.1 Dataset Preparation

For this experiment, we collected two new datasets of raw sEMG signals corresponding to ASL gestures. The signals from various sEMG sensors were simultaneously recorded and stored in CSV file format, representing each sEMG channel as a separate variable in a multivariate time series.

Let $T_s(u) = \{u_1, u_2, u_3, \ldots, u_m\}$ represent a time series corresponding to each sEMG sensor with m data points. Here, s is the total number of sEMG sensors, $W_s(p, s)$ is the window segment of length p (p data points of s sEMG channels), z is the total number of samples, and l_i is the annotation with $0 \leq l_i \leq (C_i - 1)$. C_i represents different classes, each mapped to a unique ASL gesture. The raw sEMG signals collected can be depicted as $D_{\text{raw}} = \{(T_s(u_1), T_s(u_2), T_s(u_3), \ldots, T_s(u_s), l_i)\}$ where $T_s(u) \in \mathbb{R}^{1\times n}$ and $l_i \in \mathbb{R}$.

For each window size $W_s(p, q)$, various relevant features were calculated and stored. The classes C_i can be defined as:

$$C_i = \left\{f_{(j,1)}, f_{(j,2)}, f_{(j,3)}, f_{(j,4)}, \ldots, f_{(j,h)}, l_i\right\};\ C_i \in \mathbb{R}^{N\times(h+1)},\ 1 \leq j \leq N$$

Where $f_{j,k}$ is a feature extracted for window segment $W_s(p, q)$, h is the total number of features, and C_i represents different classes, each mapped to a unique ASL gesture. The values of C_i range from 1 to Z, where Z is 10 and 24 for the ASL-10 and ASL-24 datasets, respectively.

3.2.4 Data Preprocessing

The armband sensors captured EMG signals from various muscle locations on the lower forearm. These raw sEMG signals were recorded in CSV format, with each column corresponding to an individual channel. Initially, basic data processing techniques were employed, including the identification and handling of missing

values, on the raw signals collected for the ASL gestures. Following this, feature extraction was performed, and the extracted feature values were standardized to a range of 0 to 1.

3.2.4.1 Filters

To address noise during data acquisition, a digital third-order Butterworth band-pass filter with a bandwidth of 5-500 Hz was applied to the sEMG signals. Additionally, following recommendations from Savur et al. [73], a configurable Butterworth filter was used to eliminate 50 Hz power line noise.

3.2.5 Data Segmentation

Data segmentation is essential for dividing the sEMG data stream into fixed sizes, known as window sizes, to evaluate features over the data points within these windows. The window acts as the unit size for feature collection in the time series. Selecting an appropriate window size is crucial; a smaller window may not capture a complete gesture, while a larger window may lead to computational overhead and delays in real-time applications.

To determine the optimal window size, classification was performed on various segment lengths. We applied the classification algorithm on windows containing 400, 300, 200, and 60 data samples. Separate analyses were conducted using the same window sizes with a 50% overlap. Balancing computational delay and accuracy, an optimal window size of 60 was found. Data was collected at a sampling rate of 200 Hz, and using a window size of 60 allowed us to maintain a delay within the permissible limit for real-time gesture recognition [216].

3.2.6 Feature Extraction

For the sEMG channels, a wide range of features from the frequency, time, and frequency-time domains were extracted to capture relevant biomedical signal characteristics [217]. Statistical features such as Kurtosis, Mean, Variance, Absolute Energy, Autocorrelation, and Standard Deviation were calculated. Time domain sEMG signals were transformed into the frequency domain using Fast Fourier Transform (FFT). Given that various ASL gestures produce different frequency distributions, 95 FFT coefficients from 5 to 100 Hz were selected as features [67, 218, 219]. Additionally, other features such as Entropy, Benford Correlation, c3 Non-linearity Statistics [220], Complexity-invariant Distance (Complexity Information) [221], Mexican Hat Wavelet [222], Spectral Centroid (absolute), Skewness, Kurtosis of the Fourier Transform, Power Spectral Density based on Welch Method [223], Lempel-Ziv Complexity [224], One-dimensional Matrix Profile [225], Partial Autocorrelation, Root Mean Square, Sample Entropy, Friedrich Coefficients [226], Absolute Sum of Changes (consecutive time series value changes), Langevin Fixed Point (largest point of deterministic dynamics), and Quantiles were extracted for each of the sEMG signals.

The resultant feature set was formed by combining the features from all sEMG channels.

3.2.7 Proposed Ensemble feature selection approach

To classify ASL gestures captured using sEMG signals, we extracted numerous features from the raw signals. This resulted in a high-dimensional feature vector, which can negatively impact classification performance due to the curse of dimensionality and redundancy [227]. To address this, we applied a new ensemble feature selection technique to identify the most representative feature subset from the extensive feature space. We chose filter-based feature selection methods—ANOVA, Chi-Square, Mutual Information, and ReliefF—for the ensemble approach. These

methods are independent of machine learning algorithms and are more computationally efficient compared to wrapper-based methods [228, 229].

We proposed a new approach to combine the feature sets of different individual feature selection methods for our ensemble feature selection method. The proposed combiner aggregates the feature sets to obtain an optimal feature set by incorporating feature-feature and feature-class correlation properties. Feature-class correlation is defined as the relevance of a feature based on the correlation between the feature and the ASL class label. The two concepts, feature-feature and feature-class correlation, help achieve relevant and non-redundant features from the large feature vector obtained in the feature extraction step.

The proposed ensemble selection can be explained in two major steps. In the first step, the filter-based methods (ANOVA, Chi-Square, Mutual Info, ReliefF) provide individual k-best feature sets. The choice of using four filter methods for creating the ensemble was based on the findings of Wang et al. [208].

In the second step of the ensemble, these feature sets are aggregated using the feature-feature and feature-class relationships, providing the optimal feature subset. In our work, an optimal feature set, $final\_selected$, is initially made by selecting distinct features from the four filter-based feature selection methods (ANOVA, CHI2, Mutual Info, ReliefF). In this step, using the greedy search approach, the higher-ranking features common in all four feature sets are directly added to the optimal features set, $final\_selected$. The selection of common features from all four feature sets is based on our heuristics that these features are the highly significant features with efficient discriminatory properties.

Later, the combination of any three individual feature selection methods out of the four methods are listed as groups. The non-redundant and common features in the listed group are identified. The identified features are only added to the optimal set $final\_selected$, if they satisfy the correlation threshold.

In the next step, the distinct features selected in any two of the four feature selection methods are added to the optimal feature set after validating through the feature-class threshold. A correlation test is performed on the optimal feature set, and the features having a correlation above a threshold ($thres_1$) are truncated from this feature set.

All those features not initially selected in the $final\_selected$ are combined and optimized based on the correlation threshold ($thres_2$) and stored as $rem\_features$. Further, for each feature f_i in $rem\_features$, the feature-feature and feature-class correlation with the features in the optimal feature set $final\_selected$ and ASL classes is performed. If the feature f_i satisfies the thresholds, it is added to the minimal feature set. The thresholds $thres_1$, $thres_2$, and $thres_3$ are derived based on the empirical results carried out with the aim to attain higher classification accuracy. We used the thresholds $thres_1$ and $thres_2$ equal to 0.85, while $thres_3$ is 0.90 based on the exhaustive experimental results. The initial number of "k" features is selected based on heuristics and empirical analysis. The ensemble feature selection method is illustrated as Algorithm 5. The proposed feature selection approach results in a feature subset consisting of the best discriminatory features from each sEMG channel.

3.2.7.1 Overall Complexity of the Ensemble Feature Selection Algorithm

To determine the overall complexity of the Ensemble Feature Selection algorithm, we need to combine the complexities of each individual component and step. Here is a detailed breakdown of the complexities and how they add up to the final algorithm complexity.

Initialization and Intersection:

- Initializing the feature selection methods list and sets: $O(1)$
- Computing the intersection of all feature sets: $O(k * n)$

  - k = number of feature sets (e.g., from different selection methods)
  - If each set contains n features (or up to n), and you're intersecting all sets

Pairwise Feature Identification:

- Identifying unique features using *find_features*: $O(k)$ per pairwise comparison
- Total pairwise comparisons: $\binom{4}{2} = 6$
- Complexity: $6 \times O(k) = O(k)$

Correlation Matrix Computation for Initial Features:

- Computing the correlation matrix and filtering $final\_selected$ using $thres_1$: $O(k^2)$

Filtering Remaining Features:

- Computing the correlation matrix and filtering $rem\_features$ using $thres_2$: $O(k^2)$

Evaluation of Features in Remaining Set:

- Evaluating correlation for each feature f_i in $rem\_features$: $O(k^2)$

Summarizing the Complexities: Combining the complexities of each step

- Initialization and Intersection: $O(1) + O(k) = O(k * n)$
- Pairwise Feature Identification: $O(k)$
- Correlation Matrix Computation for Initial Features: $O(k^2)$
- Filtering Remaining Features: $O(k^2)$
- Evaluation of Features in Remaining Set: $O(k^2)$

Total Complexity: Adding up all the individual complexities

$$O(k * n) + O(k) + O(k^2) + O(k^2) + O(k^2) = O(k * n) + 3 \cdot O(k^2)$$

Since $O(k^2)$ dominates $O(k * n)$, as the k is very less compared to n, the overall complexity of the Ensemble Feature Selection algorithm is:

$$\boxed{O(k^2)}$$

The overall complexity of the Ensemble Feature Selection algorithm is $O(k^2)$, indicating it is efficient and manageable for moderate-sized feature sets, making it suitable for practical applications in feature selection for classification tasks. The dominant steps contributing to this complexity are the correlation matrix computations.

Algorithm 5 Ensemble Feature Selection

Input: Feature sets consisting of k features each for ANOVA, Mutual Info, ReliefF, and Chi Square:

- $FS_{\text{chi2}} = \{f_1, f_2, \ldots, f_k\}$ ▷ using Chi-square
- $FS_{\text{mi}} = \{f'_1, f'_2, \ldots, f'_k\}$ ▷ using Mutual Info
- $FS_{\text{anova}} = \{f''_1, f''_2, \ldots, f''_k\}$ ▷ using ANOVA
- $FS_{\text{ReliefF}} = \{f'''_1, f'''_2, \ldots, f'''_k\}$ ▷ using ReliefF

Thresholds: $thres_1, thres_2$ ▷ Feature-features correlation thresholds $thres_3$ ▷ Feature-class correlation threshold

Output: Final selected features subset

1: Initialization
2: $feature\_selection\_methods = [FS_{\text{chi2}}, FS_{\text{anova}}, FS_{\text{ReliefF}}, FS_{\text{mi}}]$
3: $final\_selected = \emptyset$ ▷ Initialize selected features set as empty
4: $rem\_features = \emptyset$ ▷ Initialize remaining features set as empty
5: $final\_selected = FS_{\text{chi2}} \cap FS_{\text{mi}} \cap FS_{\text{anova}} \cap FS_{\text{ReliefF}}$ ▷ Common features in all methods

Algorithm 5 Ensemble Feature Selection (continued)

function FIND_FEATURES(X, Y)
    $features = (X \cap Y) - (FS_{\text{chi2}} \cap FS_{\text{mi}} \cap FS_{\text{anova}} \cap FS_{\text{ReliefF}})$
    return features
end function
function SEL_UNCORR_FEATURES(feature set, threshold)
    // Compute correlation matrix of feature set and drop features based on threshold
    return feature set
end function
for each distinct pair (X, Y) where $X \neq Y$ in $feature\_selection\_methods$ **do**
    $final\_selected = final\_selected \cup$ FIND_FEATURES(X, Y)
end for
$final\_selected =$ SEL_UNCORR_FEATURES$(final_selected, thres_1)$
$rem\_features = (FS_{\text{chi2}} \cup FS_{\text{mi}} \cup FS_{\text{anova}} \cup FS_{\text{ReliefF}}) - final\_selected$
$rem\_features =$ SEL_UNCORR_FEATURES$(rem_features, thres_2)$
for each feature f_i in $rem\_features$ **do**
    // Evaluate correlation coefficient C_i with all features in $final\_selected$
    // Evaluate correlation coefficient D_i with classes
    if ($C_i \leq thres_1$ and $D_i \geq thres_3$) **then**
        add f_i to $final\_selected$
    end if
end for
Final selected features subset: $final\_selected$

3.3 Results and discussion

In this section, we systematically present and analyze the experimental results. The proposed experiments were performed with the objective of attaining higher classification accuracy for sEMG-based gesture recognition tasks and answering the research questions mentioned in Section 3.1.1.2. During the experiments, we utilized mainly three datasets: ASL-10 (10 ASL digit gestures), NinaPro reference dataset (12 hand gestures), and ASL-24 (24 ASL manual alphabet gestures). The proposed pipeline is evaluated on these datasets with different objectives. ASL-10 is used to measure the efficiency of our proposed pipeline using a minimum number of sEMG sensors for 10 ASL signs (0-9 digits). At the same time, the NinaPro reference

dataset is used to study the generalizability of the proposed method on a benchmark dataset. Whereas the ASL-24 dataset is used to validate its scalability.

3.3.1 Performance Measure

We have employed standard evaluation metrics, including average accuracy, MCC score, ROC-AUC curve, and kappa, as performance measures. A detailed discussion of these metrics is provided in Section 2.3.12. These metrics offer a comprehensive assessment of the model's performance across various dimensions.

Initially, we evaluated the efficiency of the proposed pipeline on the mid-size ASL-10 dataset, running the model multiple times with ten-fold cross-validation. The pipeline achieved an impressive overall average classification accuracy of 99.99% using the CatBoost algorithm. Figure 3.5 highlights the classification accuracy achieved for the ASL gestures. This demonstrates the pipeline's effectiveness in recognizing ASL gestures, especially when using advanced ensemble learning techniques like CatBoost.

In addition, we evaluated other traditional machine learning classifiers—LDA, K-NN (K=3), and K-NN (K=5).LDA is a standard classifier and frequently used in sEMG-based classification due to its less complex nature and faster processing speed [230, 231]. Although these models achieved high accuracy and agreement scores, CatBoost's superior performance suggests it is better equipped to handle the complexities of gesture recognition. This advantage is likely due to CatBoost's ability to manage categorical features and capture complex interactions among features more effectively than simpler models like LDA or K-NN. Table 3.1 provides a summary of the performance metrics achieved by the aforementioned classifiers.

Accuracy as metric provides a straightforward measure of the overall correctness of the model's predictions. Achieving higher average accuracy signifies the model's high effectiveness in correctly classifying ASL gestures, indicating its reliability and

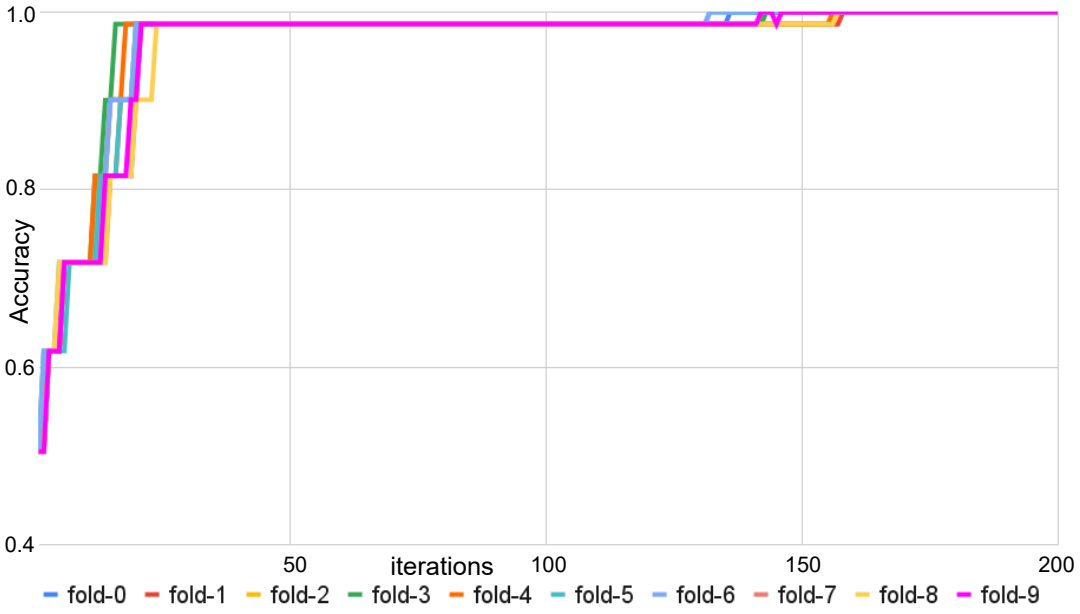

FIGURE 3.5: Classification accuracy achieved for ten-fold cross-validation with respect to the number of iterations of classification algorithm

precision in recognizing static hand gestures from sEMG signals. However, it does not account for the balance between different classes.

To address this, multi-class Matthews correlation coefficient (MCC) scores were calculated. Figure 3.6 shows the MCC score against the number of iterations of the CatBoost algorithm during its ten-fold cross-validation. MCC measures the correlation between the predicted and true values of instances by the classifier, thus validating the results in imbalanced dataset scenarios. Considering the Figure 3.6, in fewer than 50 iterations, the model achieves MCC scores greater than 0.9 and eventually reaches the highest MCC score after 150 iterations. Such High MCC scores indicate that the model performs well with both positive and negative samples of ASL gestures [232]

Meanwhile, Figure 3.7 illustrates the Kappa values obtained for various cross-validation steps. Higher Kappa scores for ASL-10 suggest strong agreement between the predicted and true values of ASL instances, demonstrating the classifier's effectiveness in real-world applications.

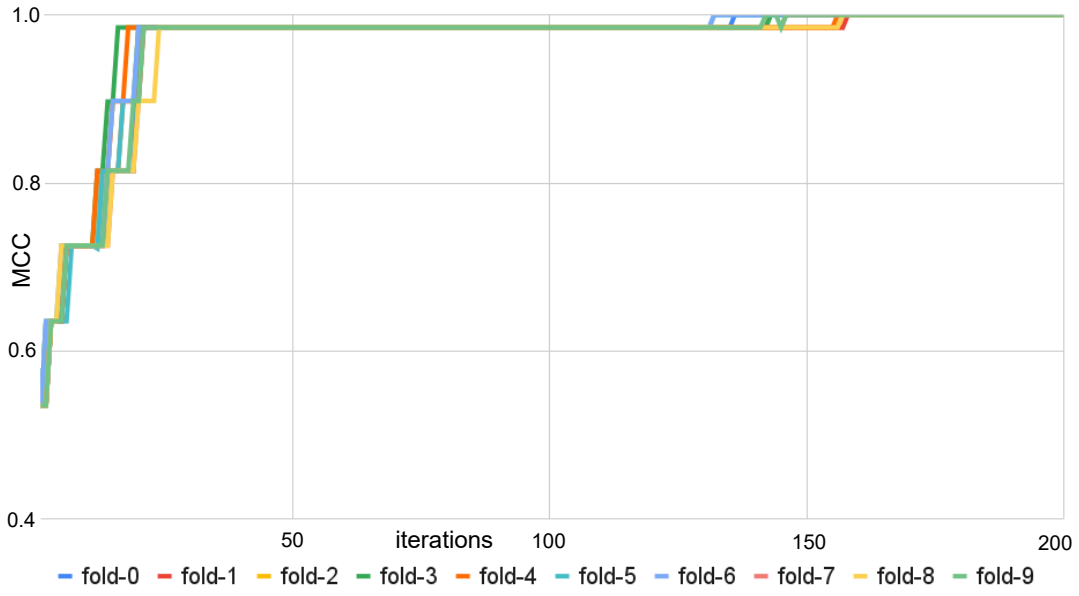

FIGURE 3.6: MCC score plotted against the number of Catboost iterations for each of 10 fold cross validation for ASL-10 dataset

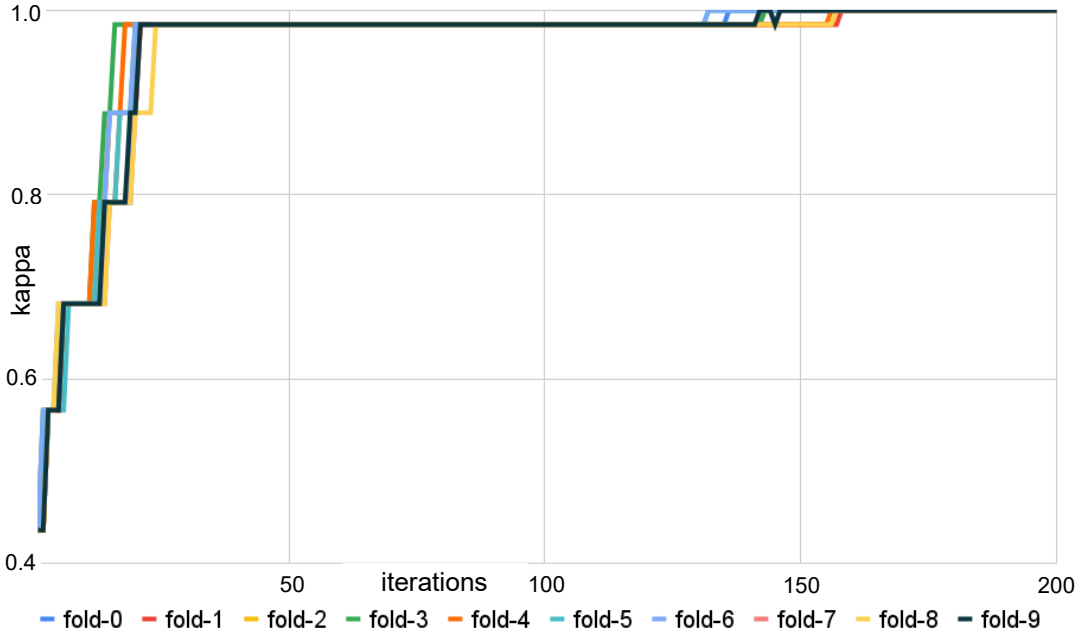

FIGURE 3.7: The Kappa values calculated for multi-classification performed at the ASL-10 dataset

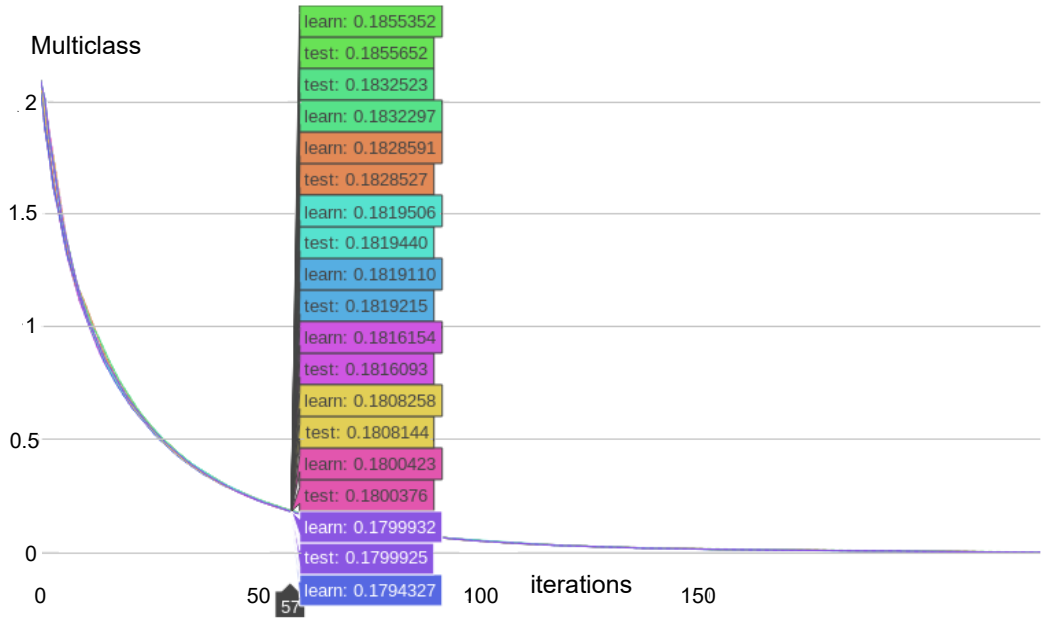

FIGURE 3.8: Figure illustrating the Multiclass loss function using 10 fold cross validation(ASL-10 dataset)

Classifiers	Accuracy	MCC	Kappa
LDA	99.60 ± 0 .13	99.3	99.4
K-nn (K=3)	99.70 ± 0.167	99.65	99.65
K-nn (K=5)	99.70 ± 0 .224	98.08	98.08
Catboost	99.91 ± 0.1	99.77	99.97

TABLE 3.1: Different performance metrics achieved using the baseline classifiers

The "multi-class" loss function was calculated and plotted for each of the ten folds to further investigate the model's performance. Figure 3.8 shows the "multi-class" loss function with respect to the number of iterations of the classification algorithm used. The multi-class loss function measures the error produced by the model when predicting class labels. Considering the Figure 3.8, the model based on our framework demonstrates a low error rate when predicting the ASL gesture class labels, validating the higher average accuracy achieved. The "multi-class" loss function approaches minimal after 150 iterations of the CatBoost algorithm. Additionally, the multi-class AUC [233] for all the ASL classes is plotted in Figure 3.9. For our model, the multi-class AUC reaches a value of 1 in fewer than 50 iterations. High AUC values suggest the model's strong ability to distinguish between positive and negative ASL classes.

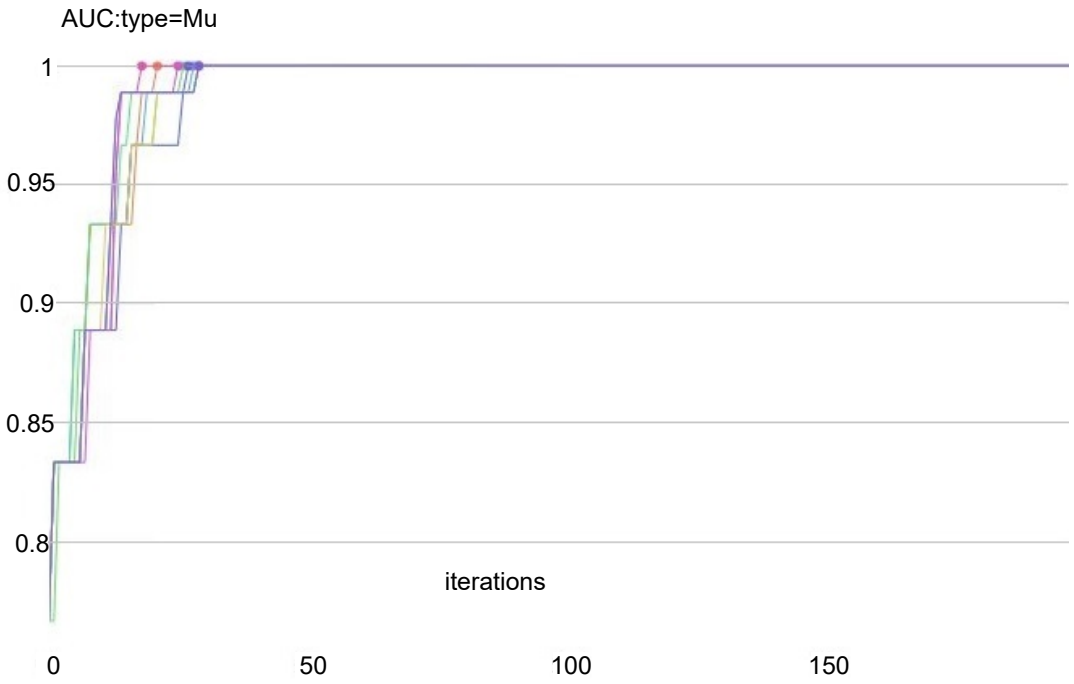

FIGURE 3.9: Multi Class AUC calculated for ten-fold cross-validation calculated for ASL-10 dataset

3.3.2 Validation of proposed framework

3.3.2.1 Statistical Analysis

We conducted a comprehensive statistical analysis of the obtained experimental results to validate their correctness, hence the performance of our proposed machine learning pipeline. Statistical hypothesis tests were applied to ensure that the results obtained were actual and not produced due to statistical fluke. Specifically, these analyses were performed with two primary objectives.

- To demonstrate the statistical significance of the features selected using our proposed ensemble feature selection method through standard statistical tests (Kruskal Wallis [234]).
- Second, to establish the correctness of the classification algorithm results by using the 5*2 cv t-test [235].

Fig. 3.10 shows the schematic diagram of the statistical tests performed to validate the obtained results.

Kruskal Wallis test was applied to statistically substantiate the significance of the features selected by our proposed ensemble method [234]. The Kruskal-Wallis test is a computationally efficient statistical method and has been frequently used in literature to determine features with significant discriminatory ability. Sharma and Pachauri [236] use the Kruskal Wallis statistical test to validate the discriminative ability of their proposed method to classify epileptic seizures and seizure-free signals. In a similar work, Khan et al. [237] applied the Kruskal Wallis test to validate the features with higher discriminative information for the face recognition task. The features with p-values near zero were considered discriminative face features and claimed to produce a high recognition rate. In [157], the author claims to find significant features for the classification task when the p-value lies near 0 (p-value<0.5).

The Kruskal-Wallis test is a non-parametric method used to validate the null hypothesis that the medians of the groups/features are similar, determining whether samples originate from the same distribution. The H-statistics and p-value are used to accept or reject the null hypothesis.

In our analysis, the Shapiro-Wilk test was initially used to validate the distribution of the selected features, supporting the use of the Kruskal-Wallis test. The Shapiro-Wilk test indicated that the data was not normally distributed (p-value ¡ 0.05), justifying the use of non-parametric methods for further analysis. This experiment confirmed that the selected features did not follow a normal distribution, making the Kruskal-Wallis test appropriate. Consequently, the Kruskal-Wallis test rejected the null hypothesis for the ASL digit dataset with a p-value less than 0.001.

However, the Kruskal-Wallis test does not provide detailed information on which groups or features differ. Therefore, a post hoc test is necessary to compare the significance of pairwise features. Since the selected features were non-parametric,

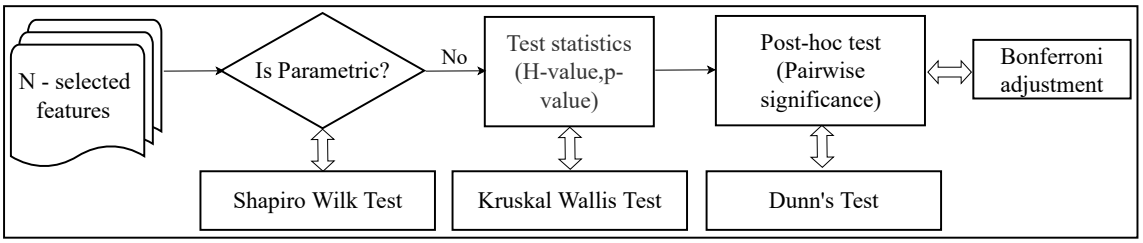

FIGURE 3.10: The schematic diagram describing the statistical test performed to validate the results

we used Dunn's test as the post hoc test [238]. Additionally, to address cumulative Type I error or alpha inflation while using Dunn's test, we applied the Bonferroni adjustment [239]. The Bonferroni adjustment controls Type I error by calculating a new pairwise alpha, ensuring the familywise alpha value remains at the specified level.

For Dunn's test, the resultant p-values for all feature pairs were reported to be less than 0.001. This indicates that the differences between the features are statistically significant, hence the significance of each feature in the subset. Specifically, Table 3.2 shows the pairwise p-values for "emg1 fft coefficient 31 (IMG)" compared to all other features after the Bonferroni adjustment. These results confirm that "emg1 fft coefficient 31 (IMG)" is significantly different from the other features.

The interpretation of these results is that the selected features exhibit significant differences from one another, suggesting their individual discriminative power. This detailed pairwise comparison validates the effectiveness of our ensemble feature selection method, highlighting the importance of these features in improving the accuracy and robustness of the classification task. The statistical significance of these differences supports the reliability of the selected features in distinguishing between different ASL gestures, ensuring that the model is based on robust and meaningful data.

TABLE 3.2: The selected features and the corresponding p-values

Channel 1 Feature	p-value	Channel 2 Feature	p-value
fft coefficient 31 (REAL)	8.372341E-12	fft coefficient 32 (REAL)	0
fft coefficient 36 (REAL)	1.5706702E-101	fft coefficient 35 (REAL)	7.696407E-68
fft coefficient 31(IMG)	1	fft coefficient 36 (REAL)	1.570670E-101
fft coefficient 32 (IMG)	0	fft coefficient 32 (IMG)	0
fft coefficient 34 (IMG)	0.001298	fft coefficient 35 (IMG)	4.8492671E-13
fft coefficient 35 (IMG)	4.849267E-13	fft coefficient 31(ABS)	0
fft coefficient 34 (ABS)	0	fft coefficient 32 (ABS)	0
fft coefficient 35 (ABS)	0	fft coefficient 33 (ABS)	0
fft coefficient 36 (ABS)	0	fft coefficient 35 (ABS)	0
fft coefficient 31 (ANG)	0	fft coefficient 31 (ANG)	0
fft coefficient 32 (ANG)	0	fft coefficient 35 (ANG)	0.000083
fft coefficient 34 (ANG)	0	fft coefficient 36 (ANG)	1.283962E-56
fft coefficient 35 (ANG)	0.000083		

3.3.2.2 5*2 CV t-test

In an another experiment to statistically validate the performance of our classification model, we used the 5*2 CV t-test proposed by Dietterich [235]. This statistical approach applies a t-test on five iterations of 2-fold cross-validation. 5*2 CV t-test can efficiently distinguish the classification algorithm's performance on a single dataset with an acceptable Type I error rate as compared to other statistical methods [235].

The use of the 5*2 cv t-test was based on our heuristics. If two different instances of the classifier, C_1, and C_2, provide statistically similar results on our selected feature set, then such results suggest the correctness and generalizability of our proposed pipeline. The two instances were made by assigning different hyperparameters to the classifier.C_1 corresponds to an instance of the classifier with default hyperparameters. In contrast, C_2 corresponds to the classifier instance with tuned hyper-parameters, which provided the best average classification accuracy on the ASL 10 dataset.

The 5*2 cv t-test randomly partitioned the dataset into two equal-size sets, X_i(train set) and $\bar{X_i}$(test set), repeatedly for five iterations. In every iteration, each of the two classifiers,C_1 and C_2, are trained using the training set X_i and evaluated on the test set $\bar{X_i}$ and vice versa. This training and testing pattern generates the four error estimates $p_{C_1}^{(1)}$, $p_{C_2}^{(1)}$, $p_{C_1}^{(2)}$, $p_{C_2}^{(2)}$ in a single iteration. These error estimates are

used to calculate the two performance measures($p_i^{(1)}$ and $p_i^{(2)}$) which are described as

$$p_i^{(1)} = p_{C_1}^{(1)} - p_{C_2}^{(1)} \tag{3.1}$$

$$p_i^{(2)} = p_{C_1}^{(2)} - p_{C_2}^{(2)} \tag{3.2}$$

With, Eq. 3.1 and Eq. 3.2 the variance can be calculated as $s_i^2 = (p_i^{(1)} - \bar{p}_i)^2 + (p_i^{(2)} - \bar{p}_i)^2$ where, $\bar{p}_i = (p_i^{(1)} + p_i^{(2)})/2$. Using these variables, the 5*2cv t-statistics is described as follows:

$$\bar{t} = \frac{p_1^{(1)}}{\sqrt{\frac{1}{k}\sum_{i=1}^{k} s_i^2}} \tag{3.3}$$

Where, $k = 5$, $p_1^{(1)}$ is the error estimate of the first iteration and s_i^2 is the variance evaluated at i-th iteration. For our experiment, p-value was reported to be 0.96 (p-value>0.05), which suggests the performance of the two instance of the classifier are similar.The 5x2 CV t-test demonstrated that the performance of the classifier is consistent and reliable, regardless of hyperparameter tuning. This suggests that our classification pipeline is robust and generalizable.

3.3.2.3 Model Scalability and Training Sample Sufficiency

A separate analysis for pipeline scalability was performed using the ASL-24 dataset. For scalability, we assumed that if the pipeline maintains consistent performance as provided to ASL-10, it is considered scalable. The ASL gestures for this dataset were deliberately increased to include 24 different gestures, corresponding to the 24 ASL manual alphabets used in finger-spelling ASL words. Using tenfold cross-validation, the model achieved an average classification accuracy of 99.91% and an MCC score of 0.99. Figure 3.11 illustrates the MCC score achieved with respect to the iterations of the CatBoost algorithm. Initially, the MCC scores ranged from

0.25 to 0.5 but reached a maximum within 50 iterations. Additionally, Figure 3.12 shows the model's efficiency by plotting the "Multi-Class" loss function against the iterations performed by the CatBoost algorithm. Both the validation and learning curves reached minimal within 400 iterations. The near-overlapping and smooth nature of these curves for all ten folds suggest low error production by the model when predicting class labels. These experimental results support the scalability of the pipeline, as the performance is similar to that achieved with ASL-10.

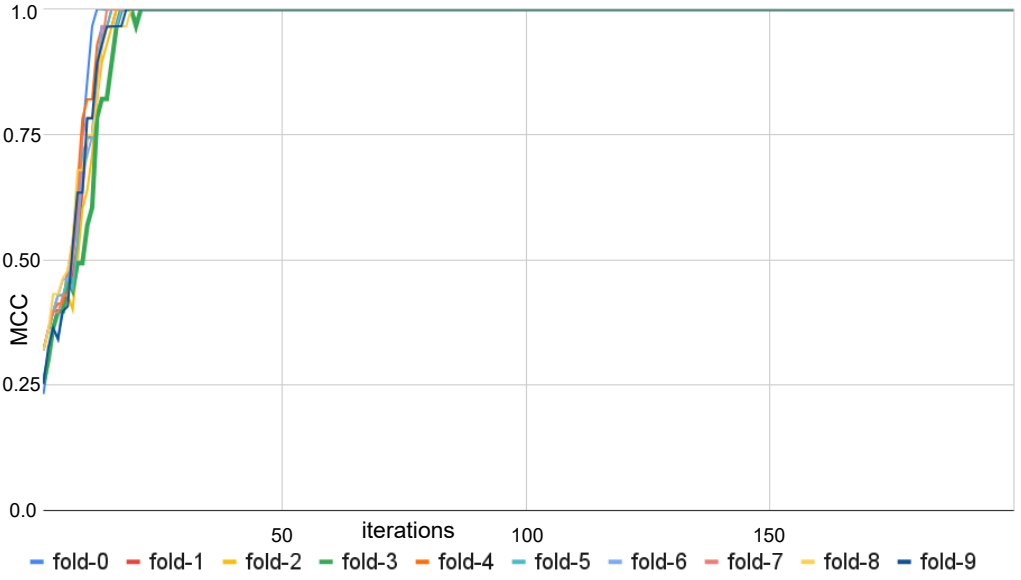

FIGURE 3.11: MCC score plotted against the number of Catboost iterations for each of 10 fold cross validation for ASL-24 dataset

To gain better insight into the trained model, we plotted graphs showing the relationships between training examples, fit_times, and accuracy scores, highlighting the model's scalability and performance. Figure 3.13 illustrates these relationships for the classification model built using our proposed pipeline. The plot on the left displays the time taken (fit_times) to train the Linear Discriminant Analysis (LDA) classifier with different numbers of samples, while the plot on the right shows the accuracy score achieved with respect to the training time (fit_times).

The results from these plots clearly indicate that the model achieves its maximum accuracy with fewer than 10,000 samples. Increasing the number of samples beyond this point does not significantly impact the model's performance.

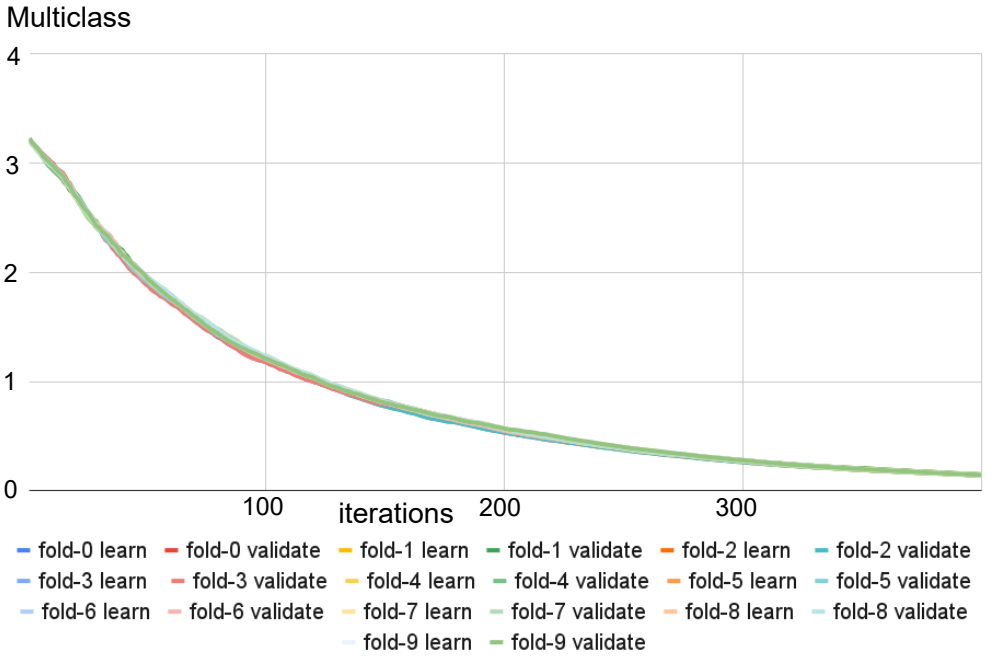

FIGURE 3.12: Figure illustrating the Multiclass loss function using 10 fold cross validation(ASL-24 dataset)

This demonstrates that the model has been trained with a sufficient number of samples, and adding more samples would not substantially improve its accuracy. Additionally, the consistent accuracy scores suggest the robustness of the pipeline, making it reliable for practical applications involving larger datasets.

3.3.2.4 Validation using public dataset: NinaPro [1]

The same pipeline was applied to the benchmark Ninapro database, commonly used for assessing sEMG-based classification methods. Specifically, we used Database 5, which includes 53 various hand movements and gestures grouped into three exercise sets. For better comparison and relevance to our problem, we focused on the raw sEMG signals corresponding to 12 different hand gestures from Exercise A of

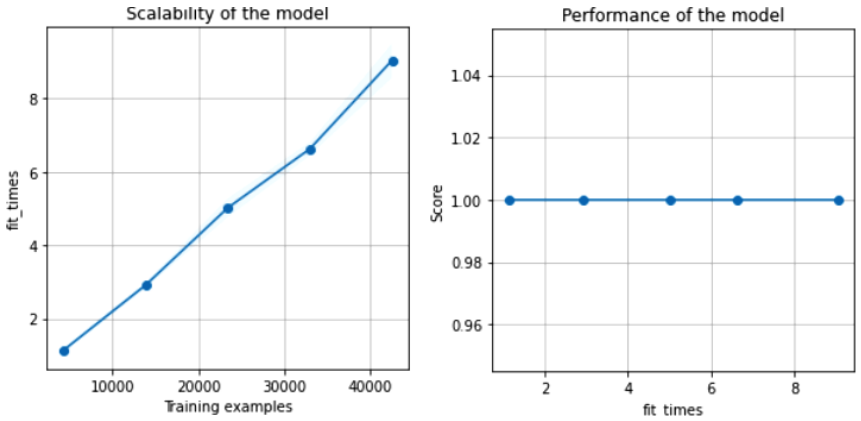

FIGURE 3.13: The figure illustrating the time required to train various numbers of samples using LDA

Ninapro Database 5 (Details provided in Section 2.3.4). The pipeline performed

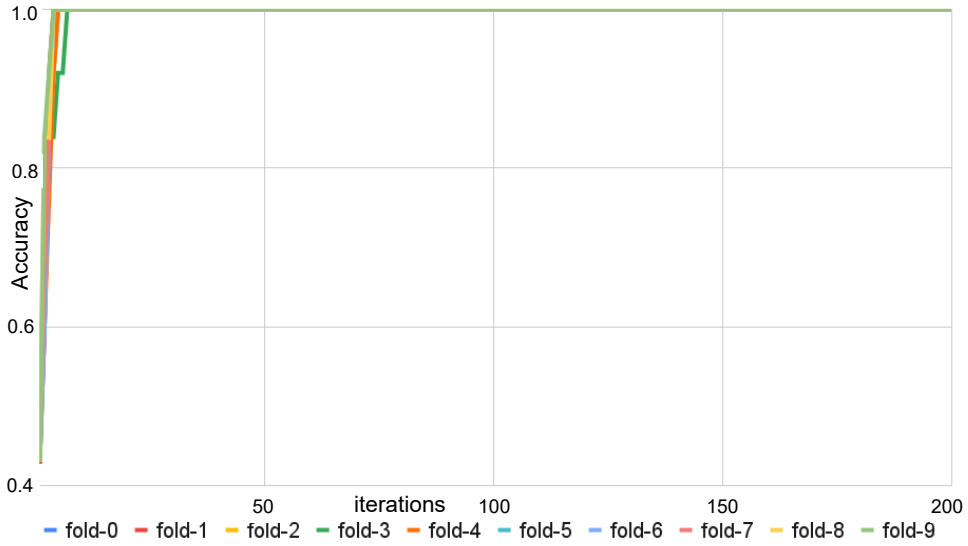

FIGURE 3.14: The accuracy obtained for the Ninapro subset using the proposed pipeline

exceptionally well, achieving an MCC score close to 1 when classifying the sEMG signals for the Ninapro database. Figure 3.14 illustrates the accuracy achieved for each of the ten-fold validations. Compared to the ASL dataset (Figure 3.5), the highest accuracy was achieved with a lower number of iterations by the classification algorithm, demonstrating the pipeline's robustness and effectiveness.

The use of the Ninapro database suggests the generalizability of our pipeline to publicly available datasets. However, it is important to note that this dataset is not specific to ASL, and the gestures in Ninapro are different from those in ASL. Despite this, the high performance of our pipeline on the Ninapro dataset indicates its potential applicability and robustness in similar sEMG-based hand gesture recognition tasks.

3.3.3 Comparison to state-of-the-art methods

In this section, we compare our proposed method with similar state-of-the-art ASL recognition methods. Various machine learning (ML) approaches have been proposed to build accurate and reliable classifiers for sEMG-based ASL gesture recognition. These ML-based approaches primarily involve deploying various sensors, extracting relevant features from sEMG signals, and applying machine learning classification algorithms. Some popular baseline approaches for sEMG-based recognition include Support Vector Machine (SVM), Hidden Markov Model (HMM), Artificial Neural Network (ANN), and Linear Discriminant Analysis (LDA).

Early work in this domain by Savur et al. [67] used an SVM classifier, achieving 61.04% classification accuracy on a multi-user dataset. Fatmi et al. [66] compared three machine learning baseline approaches (ANN, SVM, and HMM) for ASL recognition, with ANN achieving the highest overall accuracy of 93.79% compared to SVM (85.56%) and HMM (85.90%). Wu et al. [70] evaluated Decision Tree, LibSVM, Nearest Neighbour, and Naïve Bayes algorithms for ASL gesture recognition, achieving 95.94% accuracy using an additional inertial sensor.

Dynamic Time Warping (DTW) [240], which measures the similarity between two temporal sequences, has also been commonly used for ASL recognition. Paudyal et al. [69] used DTW to classify 20 ASL gestures, attaining 97.72% accuracy with two additional sensors and sEMG. However, achieving higher recognition accuracy is necessary for building a real-time ASL sign recognition system.

Considering the performance of these baseline approaches, Our proposed pipeline, using the CatBoost algorithm and an ensemble feature selection approach, achieved a classification accuracy of 99.91% for 24 ASL manual alphabets. This reported accuracy highlights the efficiency of our method compared to other baseline approaches.

Table 3.3 compares prominent sEMG-based ASL recognition research with our proposed work. The table summarizes the performance of each research work, highlighting the types of sensors used, the number of subjects involved in data collection, dataset size, number of ASL gestures, number of sensor channels used, and performance metrics.

For any two methods to be compared accurately, all related experiments must be performed and evaluated under similar conditions. However, most datasets used in these studies are not publicly available, preventing such comparative analysis. Despite this, a glance at the evaluation metrics suggests that our proposed pipeline performs better or comparably to state-of-the-art methods for ASL recognition tasks.

Our proposed pipeline, with two sEMG channels, achieved a classification accuracy of 99.99% for 10 ASL gestures. The ensemble feature selection method helped obtain the minimal number of features (25 features) for the classification tasks. Among the various research works listed in Table 3.3, only the authors in [72] used a reduced number of sensors (2 sEMG channels). Compared to this work, our proposed model achieves better classification accuracy and is more robust, as they only used 180 samples for model validation.

Moreover, our proposed pipeline demonstrated efficient scalability while recognizing an increased number of ASL gestures. When applied to a dataset consisting of 24 manual alphabet gestures (ASL-24), it achieved an average classification accuracy of 99.91%. Compared to state-of-the-art methods (Table 3.3), this reported accuracy is the highest for an approach using a minimal number of sEMG sensors. The proposed pipeline, when tested on the ASL-24 dataset, performed better than

the work of Fatmi et al. [66], Paudyal et al. [69], Taylor [68], Kosmidou et al. [72], and Savur et al. [73]. While we cannot directly compare classification accuracy with Wu et al. [71] and Wu et al. [70], as they used a greater number of ASL gestures, our approach is comparable since they used more sensors and validated their models on datasets with fewer samples. Additionally, our model is more generalizable, having been validated on datasets with a larger number of participants. Savur et al. [67] and Savur et al. [73] attained lower classification accuracy with a comparable number of ASL gestures.

Based on the above comparison, we can conclude that our proposed pipeline is better suited for developing cost-effective Sign Language Recognition Systems (SLRS) using a minimal number of sensors.

TABLE 3.3: Summary of the related work for sEMG sensor-based ASL recognition

References	Different types of sensors used	No of samples	No of gestures	No of subjects	No of channels	Feature Explainablility	Classification Accuracy
[67]	1	1040	26	10	8	No	61.04%
[66]	4	26000	13	3	26	No	93.79%
[69]	3	-	20	10	34	No	97.72%
[70]	3	3000	40	4	10	No	95.94%
[71]	3	24000	80	4	10	No	85.24%-96.16%
[72]	1	180	9	-	2	No	97.7%
[73]	1	2080	26	-	8	No	91.1%
[68]	3	-	20	-	15	No	94%-98%
Proposed method (ASL-10)	1	**53000**	10	20	2	Yes	**99.99%**
Proposed method (ASL-24)	1	**83000**	24	10	8	Yes	**99.91%**

3.3.4 Insightful observations on selected features

Among the features extracted from the raw sEMG signals, the majority of FFT coefficients were identified as the most significant by the ensemble feature selection. The selected feature subset, as shown in Table 3.2, consists of various FFT components in the range from coefficient 30 to coefficient 36. In another experiment, by increasing the number of features to the top 250 in the first level of the ensemble

feature selection, we found that the range of selected FFT coefficients extended from coefficients 30-36 to coefficients 30-51.

The importance of individual features was analyzed in separate experiments. To assess the efficiency of each FFT component, only single components of the FFT coefficients (REAL, IMG, ABS, ANG) were used in the selected feature set when applying the pipeline. Table 3.4 highlights the average classification accuracy achieved using various features. The combined FFT coefficients outperformed the individual components.

Considering Table 3.4, FFT features demonstrate the highest classification accuracy across all categories. The individual FFT features such as FFT(ABS), FFT(IMG), FFT(ANG), and FFT(REAL) achieve impressive accuracies of 99.71%, 99.53%, 99.49%, and 99.42%, respectively. These results highlight the robustness of FFT features in capturing the relevant patterns from the sEMG signals. Furthermore, when FFT features are combined, the classification accuracy reaches an optimal 99.99%, indicating that the combination of different FFT components provides a comprehensive and highly discriminative feature set.

Other features, excluding FFT, also perform well, achieving average accuracies of 97.22% for 8 channels and 96.7% for 2 channels. While these accuracies are slightly lower than those of the FFT features, they are still significantly high, suggesting that non-FFT features also contain valuable information for the classification task. However, the slight drop in accuracy when using only 2 channels indicates that having more channels contributes to better feature extraction and, consequently, higher classification accuracy.

In contrast, only time-domain features exhibit the lowest performance among all feature types. With an average accuracy of 51.09% for 8 channels and 43.01% for 2 channels, these features are less effective in capturing the discriminative properties of the sEMG signals. The lower accuracy rates suggest that time domain features

may not adequately represent the complex patterns and variations in the sEMG data, especially when fewer channels are used.

The Friedrich coefficients [226], with an average accuracy of 83.33% for 8 channels, demonstrate a moderate performance. While not as high as the FFT features, they still provide a reasonably good classification accuracy, indicating their potential utility in certain contexts.

The result obtained highlights the application of the Fourier transform as a sufficient feature extraction technique for classifying ASL and other similar hand gestures involving combined or single-finger movements (refer to Figure 2.4 and Figure 2.6 in Section 2.3.3). The FFT coefficients evaluated for sEMG signals are complex numbers and can be decomposed into real (REAL), imaginary (IMAG), absolute (ABS), and angle (ANG) components. These individual components can be treated as a separate feature for recognizing the sEMG-based gestures. The sEMG signals contain frequencies in the range of 0-500 Hz. For our collected ASL gesture dataset, while performing classification, the FFT coefficients corresponding to the lower frequency component of sEMG signals show a more significant contribution than the higher frequency components. When treated as features, these lower frequency FFT components as a group exhibit sufficient information to identify effectively various hand gestures involving finger movements.

TABLE 3.4: Various features and the corresponding performance metrics

Features	Average accuarcy	Features	Average accuarcy
FTT(ABS)	99.71%	FFT combined(8 channels)	99.99%
FFT (IMG)	99.53%	Other features excluding FFT (8 channels)	97.22
FFT(ANG)	99.49%	Time domain features (8 channels)	51.09
FFT (REAL)	99.42%	Friedrich coefficients (8 channels)	83.33
FFT combined	99.99%		
Time domain features (2 channels)	43.01%		
Other features excluding FFT(2 channels)	96.7%		

3.3.4.1 Feature visualization

t-Distributed Stochastic Neighbor Embedding (t-SNE) [241] was used to map the selected feature subset into a lower-dimensional space for better visualization. t-SNE is a nonlinear dimensionality reduction technique that efficiently captures the local structure (patterns) in the original space and represents them in lower dimensions [242]. It ensures that if two points are close in the higher-dimensional space, they remain close to each other in the reduced dimension. Figure 3.15 illustrates the labeled selected feature set for the ASL dataset when visualized through t-SNE. For the t-SNE plot, the hyperparameters "max-iterations" and "perplexity" were set to 1000 and 50, respectively. Figure 3.16 shows the t-SNE plot for the feature subset obtained for the Ninapro data subset, with the same hyperparameter values as used for the ASL dataset. The selected features for the Ninapro dataset also exhibited similar properties to those of the ASL dataset, forming almost non-overlapping clusters for each gesture class. This indicates that the features are highly discriminative, allowing the classification algorithm to distinguish between different gesture classes with minimal effort.

3.3.4.2 Others Insights on Feature using Explainable AI (SHAP [2])

To identify the impact of various features on the classification task, we calculated the Shapley Additive explanations(SHAP values) [2].SHAP is a technique that explains the factors contributing to predictions made by a machine learning model. It includes the calculation of Shapley values based on game theory. Figure 3.17 highlights the global importance of the selected features while classifying different instances of the ASL Classes. In the plot, features with larger Shapley values (absolute) are arranged in decreasing order. The larger the Shapley values the larger is the importance of the features for predicting the class labels. The global importance are obtained by averaging the Shapley values(absolute) per features in the dataset.The FFT coefficients 33 (IMG) have the highest impact on all the 10 classes

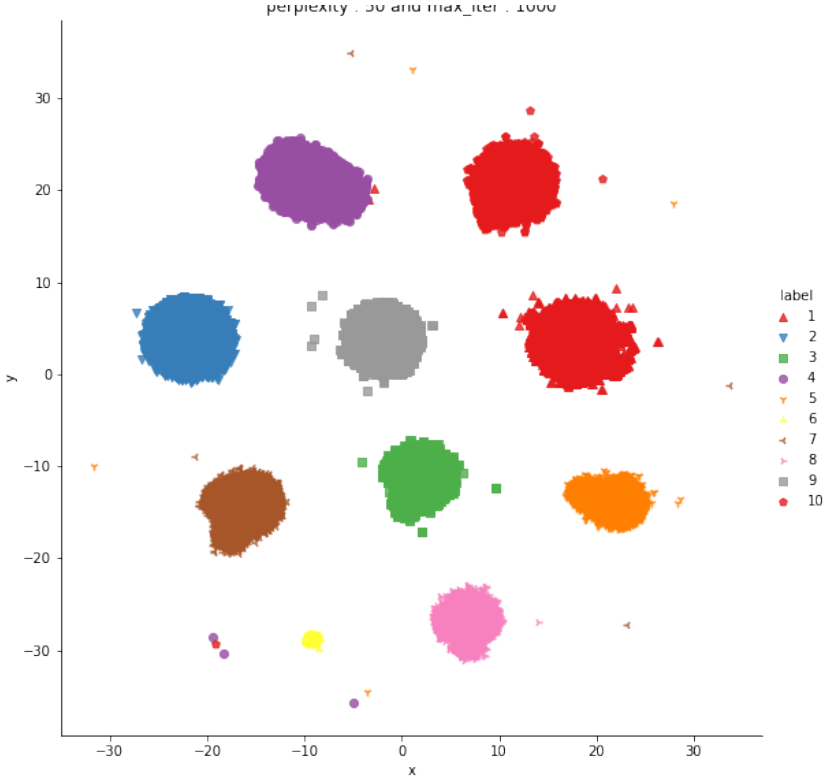

FIGURE 3.15: Features distribution corresponding to ten classes of ASL-10 dataset using t-sne

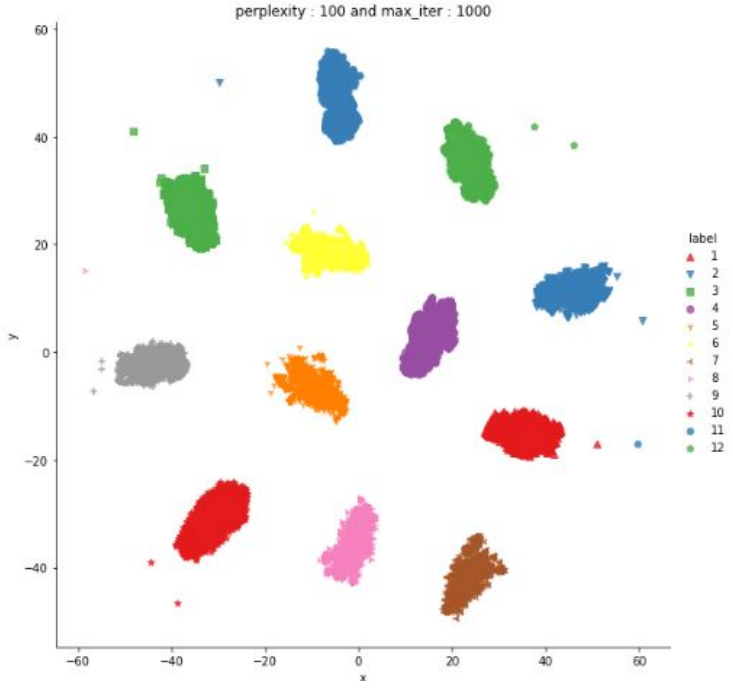

FIGURE 3.16: 2D-classwise features projection using t-sne (Ninapro dataset)

of ASL digits. Similarly, the FFT coefficients 33(ANG) and 35(ANG) are the next influential features having the most significant impact on all the class predictions.

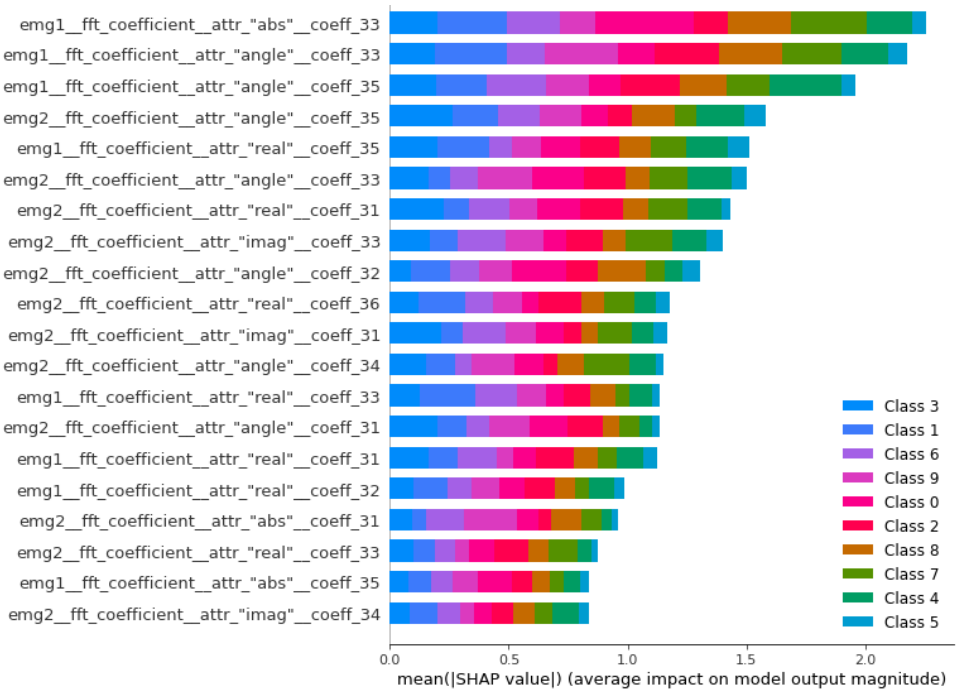

FIGURE 3.17: Classwise feature importance illustrated using SHAP

In addition, Figure 3.18 illustrates the feature importance and the effect of feature value on individual class label prediction. The y-axis of the plot shows the various features involved while classifying the class 1 of the ASL dataset (gesture corresponding to ASL digit 1). At the same time, the values at the x-axis represent the Shapley values obtained from those features. Meanwhile, the colors in the figure reflect the feature's value from high to low. For the FTT coefficient 35(ANG), the higher the value, the higher the impact on the class label prediction. Similarly, the effect of other individual feature values on the class prediction can be derived using the figure.

SHAP value analysis provided additional insights, highlighting specific FFT coefficients such as 33 (IMG), 33 (ANG), and 35 (ANG) as the most impactful for accurate classification across all ASL classes. These findings were visually supported by t-SNE plots, which mapped the selected feature subsets into lower-dimensional

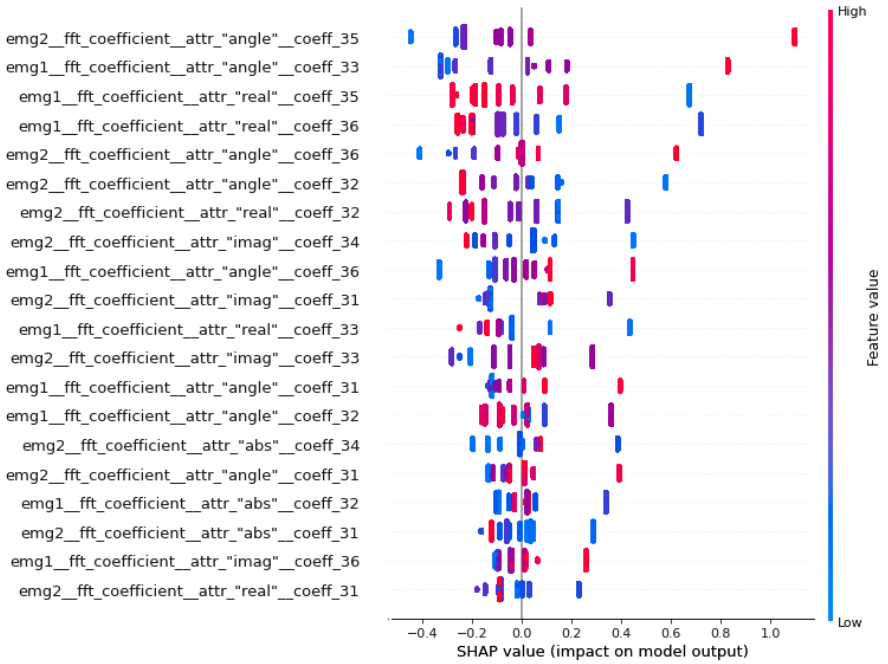

FIGURE 3.18: Figure illustrating the impact of feature values on the model performance while predicting class 1 of the ASL dataset

spaces, forming distinct clusters for each gesture class and reinforcing the robustness of the selected FFT features. The lower frequency FFT components, corresponding to the 0-500 Hz range of sEMG signals, showed a significant contribution to gesture classification, confirming their utility in capturing essential characteristics of hand gestures involving finger movements. Collectively, these results validate the selection and combination of FFT coefficients as effective features for sEMG-based hand gesture recognition systems.

3.3.5 Run Time Analysis

We assessed the module's recognition time, termed as Response Time (T_{Res}) (refer to Section 2.3.12), by measuring the duration required for the trained classifier to predict a single ASL gesture. The total Response Time includes the time delays for window segmentation, feature extraction, and recognition. The overall Response Time was 316 ms, with 300 ms allocated for window segmentation, 7 ms for feature extraction, and 9 ms for recognition. Enhancing the pipeline efficiency, especially

TABLE 3.5: Response time of the proposed pipeline to identify a single ASL gesture

Window segmentation (T_{w_s})	Feature Extraction (T_{fe})	Recognition Time (T_{Recog})	Response Time (T_{w_s}+T_{fe}+T_{Recog})
300 ms	7 ms	9 ms	316 ms

with higher sampling frequency sEMG sensors, could potentially reduce this time further. Table 3.5 outlines the details of each component of the Response Time.

All evaluations were conducted on a workstation equipped with an Intel Core i7 processor (2.9 GHz) and 16 GB of RAM, rendering the 316 ms processing delay suitable for real-time recognition.

3.3.6 Threats to Validity

Despite providing better classification accuracy for the ASL gesture recognition task, several limitations present potential threats to the validity of our findings. These challenges, which may impact the internal, external, and conclusion validity of the study, are discussed below:

Internal Validity

The accuracy of sEMG-based gesture recognition is sensitive to factors such as sensor placement, session timing, muscle fatigue, and skin impedance [243] [244] [245]. We mitigated these effects through controlled experimental protocols, including consistent sensor placement, short intraday sessions with rest periods, and the use of conductive gel. Nonetheless, deviations from these conditions in less controlled environments may reduce internal validity and impact model performance.

External Validity

The generalizability of our results is limited by the study's scope. The model was evaluated on isolated, static ASL gestures, whereas real-world communication involves dynamic, sequential signs and multimodal inputs. Without incorporating

gesture segmentation, temporal modeling, and visual cues such as facial expressions, the system's effectiveness in practical ASL contexts remains uncertain.

Construct Validity

This study defines ASL recognition in terms of static hand gestures derived from sEMG signals. However, ASL is a linguistically rich visual language involving motion, spatial structure, and facial expressions. The model, while effective in its limited domain, does not fully represent the broader construct of ASL, reducing alignment between the task and the concept it aims to model.

Conclusion Validity

The reported accuracy reflects performance under ideal, lab-controlled conditions. The lack of real-time evaluation and exclusion of dynamic or multimodal elements may inflate expectations of real-world performance. Caution is advised when interpreting these results beyond the experimental setting, pending further testing in naturalistic environments.

3.4 Summary

In this chapter, we presented a comprehensive approach for static hand gesture recognition using surface electromyography (sEMG) biosignals, specifically focusing on American Sign Language (ASL) finger-spelling gestures. ASL was chosen due to its widespread use and the current lack of a commercial, cost-effective ASL recognition system despite technological advancements in related fields such as sensors, activity recognition, and deep learning algorithms.

Our objective was to contribute to this field by developing an accurate machine-learning framework for ASL gesture recognition. This involved identifying suitable methods for data collection protocols, preprocessing techniques, feature extraction,

and selecting an appropriate machine learning classifier. Such a framework can serve as a foundation for building effective sign language recognition systems.

Identifying and understanding the features responsible for higher recognition accuracy is crucial in sEMG-based recognition. In this chapter, we employed statistical methods and explainable AI (XAI) to identify the most critical features. Notably, FFT coefficients were found to be highly significant, providing sufficient discriminative information to distinguish between different static ASL gestures.

During our experiments, we found that the proposed pipeline demonstrates scalability to larger and more diverse datasets for sEMG-based static hand gesture recognition while maintaining high performance. It is also suitable for real-time applications, with a total response time of approximately 316 milliseconds—300 milliseconds for window segmentation, 7 milliseconds for feature extraction, and 9 milliseconds for recognition. This quick processing time ensures the feasibility of the pipeline for real-time use.

Furthermore, the findings confirmed that the ensemble filter-based feature selection techniques not only improved classification accuracy but also improved feature interpretability when combined with XAI models.