This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3126348, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Neural Network Based on the Johnson S_{U} Translation System and Related Application to Electromyogram Classification

HIDEAKI HAYASHI[1], **(Member, IEEE), TARO SHIBANOKI**[2], **(Member, IEEE), AND TOSHIO TSUJI**[3], **(Member, IEEE)**

[1]Department of Advanced Information Technology, Kyushu University, Fukuoka, Japan (e-mail: hayashi@ait.kyushu-u.ac.jp)
[2]College of Engineering, Ibaraki University, Hitachi, Japan
[3]Department of System Cybernetics, Graduate School of Engineering, Hiroshima University, Higashi-hiroshima, Japan

Corresponding author: Hideaki Hayashi (e-mail: hayashi@ait.kyushu-u.ac.jp).

This work was supported in part by JSPS KAKENHI Grant Number JP17K12752 and JST ACT-I Grant Number JPMJPR18UO.

ABSTRACT Electromyogram (EMG) classification is a key technique in EMG-based control systems. Existing EMG classification methods, which do not consider EMG features that have distribution with skewness and kurtosis, have limitations such as the requirement to tune hyperparameters. In this paper, we propose a neural network based on the Johnson S_{U} translation system that is capable of representing distributions with skewness and kurtosis. The Johnson system is a normalizing translation that transforms non-normal distribution data into normal distribution data, thereby enabling the representation of a wide range of distributions. In this study, a discriminative model based on the multivariate Johnson S_{U} translation system is transformed into a linear combination of coefficients and input vectors using log-linearization; then, it is incorporated into a neural network structure. This allows the calculation of the posterior probability of each class given the input vectors and the determination of model parameters as weight coefficients of the network. The uniqueness of convergence of the network learning is theoretically guaranteed. In the experiments, the suitability of the proposed network for distributions including skewness and kurtosis was evaluated using artificially generated data. Its applicability to real biological data was also evaluated via EMG classification experiments. The results showed that the proposed network achieved high classification performance (e.g., 99.973% accuracy using Khushaba's dataset) without the need for hyperparameter optimization.

INDEX TERMS Biomedical signal processing, electromyography, Johnson distribution, neural networks, pattern recognition

I. INTRODUCTION

Biosignals such as electroencephalograms (EEGs), electrocardiograms (ECGs), and electromyograms (EMGs) are strong indicators of a human's internal state and intentions, and therefore, have been applied to human–machine interfaces and diagnosis [1]–[4]. In particular, EMG-based control systems have been widely studied because EMGs can be voluntarily controlled [5]–[12]. Many practical applications have been developed, typified by myoelectric prosthetics, which are prosthetic hands that can be controlled using surface EMGs [13], [14].

According to Oskoei and Hu [15], EMG-based control systems include four main stages: data segmentation, feature extraction, classification, and control. The raw data are first segmented and then converted into feature vectors. These vectors are classified into predefined categories. Based on the classification results, the controller transmits output commands to the device.

To realize highly intuitive and dexterous control, it is critical to achieve a high level of classification performance in terms of accuracy and training/prediction speed. Classifiers such as the support vector machine (SVM) [16], multilayer perceptron (MLP) with backpropagation learning [17], and k-nearest neighbors algorithm (k-NN) [18] have been widely used. These popular techniques, however, are not always the most suitable for EMG classification in spite of their high classification abilities. For example, SVMs are computationally expensive for hyperparameter optimization, and MLP requires lengthy training time. Similarly, it is difficult to use k-NN in real-time applications because of the large computational cost for prediction.

To improve the classification ability for a particular purpose, stochastic models can be incorporated into the structure of the classifier as long as prior knowledge of the input signals exists [19]–[23]. For instance, Tsuji *et al.* [22] proposed a Gaussian mixture model-based neural network, known as the log-linearized Gaussian mixture network (LLGMN), by assuming that the input signals obey a Gaussian mixture model.

We consider the following assumptions to be the *prior knowledge* of the feature vectors obtained from EMG signals:

- The distribution of input signals for each class is unimodal.
- The distribution has skewness and kurtosis greater than zero.

The derivations of these assumptions are explained in Section III.

To satisfy the above assumptions, we use a flexible distribution known as the Johnson distribution [24], which represents the mean, variance, skewness, and kurtosis using four parameters. Its extension to higher dimensions is enabled by the multivariate Johnson translation system [25]–[27]. A classifier that was constructed incorporating the multivariate Johnson distribution in its structure would be applicable to EMG classification and EMG-based systems.

This paper proposes a neural network (NN) based on the Johnson S_{U} translation system. The proposed NN is able to represent a flexible distribution by including a discriminative model based on the multivariate Johnson S_{U} translation system, thereby supporting the accurate classification of data with skewness and kurtosis. The parameters of the model can be determined as weight coefficients of the proposed NN via learning.

The contributions of this study are as follows:

- **Novel NN.** We proposed an NN based on the Johnson S_{U} translation system. The proposed NN is capable of representing distributions with skewness and kurtosis.
- **Usefulness in EMG classification.** We showed that employing the Johnson S_{U} is effective for EMG classification. The proposed NN is a relatively fast and accurate classifier. The proposed NN can be trained without hyperparameter optimization, and the training converges to a unique solution.
- **Connection of NN and probabilistic model.** We developed a methodology to incorporate a model that includes domain-dependent knowledge into an NN structure. We revealed that the capability of an NN can be improved by introducing prior knowledge about the domain.

This paper is related to our previous workshop paper [28]. Our previous work was preliminary, and had the following drawbacks:

- The network structure did not correctly represent the Johnson distribution due to the lack of the Jacobian.
- The training algorithm did not guarantee the uniqueness of convergence.
- The dataset variety and comparisons were limited in the EMG classification experiments.

In this paper, the above problems were addressed.

The rest of this paper is organized as follows: Related studies and their characteristic comparisons are described in Section II. Section III explains the derivation of the above assumptions, and describes a discriminative model based on the multivariate Johnson translation system and its transformation to linear combinations of weight coefficients and input vectors via log-linearization. The structure and learning algorithms of the proposed NN are presented in Section IV, and the results of a simulation experiment using artificial data are described in Section V. Section VI outlines the application potential for biosignal classification based on the EMG classification experiment. Finally, Section VII concludes the paper.

II. RELATED WORK

This section summarizes popular algorithms for EMG classification and compares each of their characteristics. The algorithms compared in this section are:

- SVM [16]
- LLGMN [22]
- MLP [17]
- Linear logistic regression (LLR) [29]
- k-NN [18]
- Random forests [30]

These algorithms are compared in terms of the following significant factors for EMG classification and EMG-based control systems: (non-)requirement of hyperparameter optimization, training speed, uniqueness of solutions, prediction speed, nonlinearity, and computability of posterior probabilities. The first three factors are associated with the effort needed to construct a classifier. Hyperparameter optimization is conducted before training, and is time consuming; hence, the usability of a system is enhanced if this step is not required. Fast training and a unique solution are also desirable to avoid effort and uncertainty in the training of the classifier. Online systems require fast prediction, and nonlinearity and computability of posterior probabilities are related to the accuracy of classification. Although EMG classification problems are unlikely to be linear, their nonlinearity is not overly complex because each class of EMG signals can be clustered to some extent. The ability to calculate posterior probabilities can assist to minimize risk, reject options, compensate for class priors, and combine models [29].

Table II summarizes the characteristics of the classification algorithms. The SVM is a distinguished classifier that

TABLE 1. Characteristics of classification algorithms

Algorithm	Hyperparameter-free	Fast training	Unique solution	Fast prediction	Nonlinearity	Posterior probability
SVM	×	✓	✓	×	✓	×
LLGMN	×	×	×	✓	✓	✓
MLP	×	×	×	✓	✓	×
LLR	✓	✓	✓	✓	×	✓
k-NN	×	n/a	n/a	×	✓	×
Random forest	×	✓	×	×	✓	×
Proposed method	✓	×*	✓	✓	✓	✓

* Depends on the input dimensionality. Faster than MLP and LLGMN when the input dimensionality is small (see Section VI).

realizes fast training and a unique solution. However, its limitations include the optimization of two hyperparameters, and lengthy multi-class classification times because it is originally developed as a binary classifier.

The LLGMN is a discriminative model that incorporates Gaussian mixture models into an NN structure. This allows for accurate calculation of the posterior probability. The number of Gaussian distributions used in the model should be carefully considered because the classification ability of the LLGMN for data following a non-Gaussian distribution decreases when there are too few components.

The MLP is generally more compact than an SVM and hence gives faster predictions. However, training has a large computational cost. The number of layers and units should be determined as hyperparameters.

The LLR is a probabilistic discriminative model that can be trained using Newton's method. The structural limit of the LLR is its inability to solve nonlinear separation problems.

The k-NN algorithm is a very simple algorithm that does not require training. However, predictions entail a large computational expense since k-NN compares the distance between the input vector and every training vector. The the number of nearest neighbors, k, used in the voting is user-defined, and selected based on heuristic techniques.

Random forests are an ensemble learning method that attempts to decorrelate the base learners by learning trees using randomly chosen subsets of input variables and training instances. Random forests can quickly train an accurate nonlinear classifier using an ensemble of simple decision trees; however, the size of the randomly selected subset of input variables should be determined empirically.

The proposed NN is designed to optimize the above characteristics as much as possible. In particular, there is no need to optimize the hyperparameters, and the uniqueness of solutions and computability of posterior probabilities can be explained theoretically.

III. MODEL STRUCTURE

A. SKEWNESS AND KURTOSIS IN PROCESSED EMG SIGNALS

Although the raw EMG signals can be considered to obey a Gaussian distribution with a zero mean [31], the distribution of extracted features may exhibit some skewness and kurtosis. We describe how the features obtain skewness and kurtosis during the feature extraction process. Although many feature extraction methods have been proposed [32]–[34], we focus on the method Fukuda *et al.* [14] described because of its simplicity and universality.

Fukuda's method consists of two main parts: rectification and smoothing based on a Butterworth low-pass filter. Rectification takes the absolute value of the raw EMG signals, converting negative EMG values to positive values. Rectification is also used in methods such as integrated EMG (IEMG), mean absolute value (MAV), and modified mean absolute value (MMAV) [35], and is strongly related to the occurrence of skewness and kurtosis. Let x be a raw EMG signal that obeys a Gaussian distribution with a mean of 0 and a standard deviation of σ. The skewness and kurtosis of x are both 0. Fig. 1 (a) shows an example of a raw EMG signal x and its histogram.

The probability density function of the rectified EMG signal $y = |x|$ is represented as

$$p(y) = \begin{cases} \frac{2}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{y^2}{2\sigma^2}\right] & (0 \leq y < \infty) \\ 0 & (-\infty < y < 0) \end{cases}. \quad (1)$$

The mean M_y and the variance V_y of y are then calculated as

$$M_y = \int_{-\infty}^{\infty} y p(y) dy = \sqrt{\frac{2}{\pi}}\sigma, \quad (2)$$

$$V_y = \int_{-\infty}^{\infty} (y - M_y)^2 p(y) dy = (1 - \frac{2}{\pi})\sigma^2. \quad (3)$$

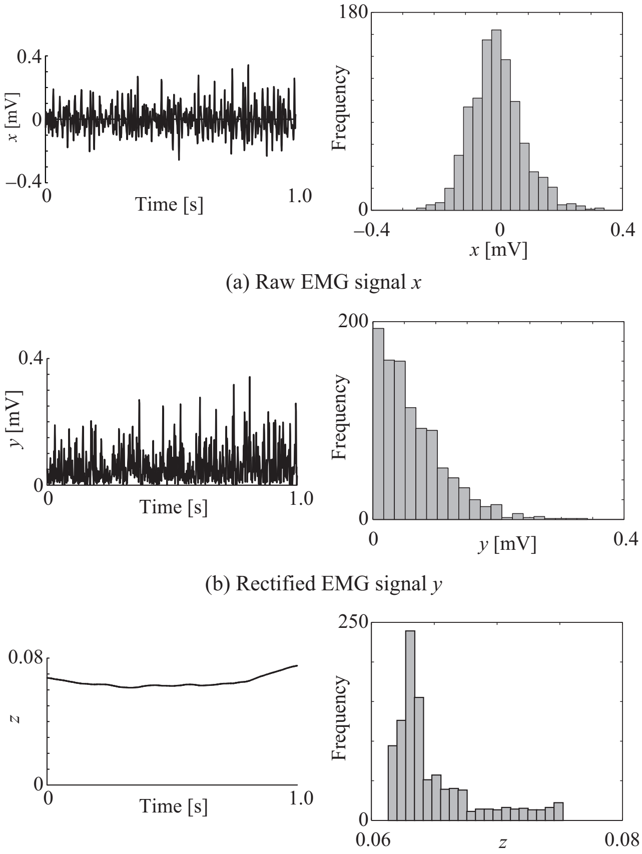
For the rectified signal y, the skewness S_y and the kurtosis K_y are no longer 0. They can be calculated as follows [36]:

$$S_y = \frac{\int_{-\infty}^{\infty}(y - M_y)^3 p(y) dy}{V_y^{\frac{3}{2}}} = \frac{(\frac{4}{\pi} - 1)\sqrt{\frac{2}{\pi}}}{(1 - \frac{2}{\pi})^{\frac{3}{2}}} \neq 0, \quad (4)$$

$$K_y = \frac{\int_{-\infty}^{\infty}(y - M_y)^4 p(y) dy}{V_y^2} - 3$$
$$= \frac{(3 - \frac{4}{\pi} - \frac{12}{\pi^2})}{(1 - \frac{2}{\pi})^2} - 3 \neq 0. \quad (5)$$

The influence of rectification is also visible in the smoothed signal z (see Fig. 1 (c)).

Since the extracted features include skewness and kurtosis, conventional Gaussian-based models cannot readily model the data. In the next subsection, we discuss the multivariate Johnson translation system [25] that we adopted, which is suitable for data with skewness and kurtosis.



(a) Raw EMG signal x

(b) Rectified EMG signal y

(c) Smoothed EMG signal z

FIGURE 1. Examples of the time-series signal and histogram of (a) raw EMG x, (b) rectified EMG y, and (c) smoothed EMG z. (a) The raw EMG x obeys a Gaussian distribution with zero mean (this is also discussed in Hogan and Mann [31]). The histograms of (b) rectified EMG y and (c) smoothed EMG z, however, become asymmetric and include skewness and kurtosis.

B. MULTIVARIATE JOHNSON TRANSLATION SYSTEM

The Jonson translation system is a family of transformations of a non-normal variate to a standard normal variate proposed by N. L. Johnson in 1949 [24]. Based on this translation, Johnson derived a system of distributions that is suitable for representing distributions with skewness and kurtosis. Later, a multivariate extension was also proposed [25].

Consider a d-dimensional continuous random vector $\boldsymbol{x} \in \mathbb{R}^d$ with skewness and kurtosis. The multivariate Johnson translation system [25] involves the normalizing translation:

$$\boldsymbol{z} = \boldsymbol{\gamma} + \boldsymbol{\delta}\boldsymbol{g}[\boldsymbol{\lambda}^{-1}(\boldsymbol{x} - \boldsymbol{\xi})] \sim \mathcal{N}(\boldsymbol{0}, \Sigma), \tag{6}$$

where $\boldsymbol{z}$ is a random vector obeying a normal distribution with mean 0 and variance Σ, $\boldsymbol{\gamma} \equiv [\gamma_1, \ldots, \gamma_d]^{\mathrm{T}}$ and $\boldsymbol{\delta} \equiv \mathrm{diag}[\delta_1, \ldots, \delta_d]$ are shape parameters, $\boldsymbol{\lambda} \equiv \mathrm{diag}[\lambda_1, \ldots, \lambda_d]$ is a scale parameter, $\boldsymbol{\xi} \equiv [\xi_1, \ldots, \xi_d]^{\mathrm{T}}$ is a location parameter, and $\boldsymbol{g}(\cdot) \equiv [g_1(\cdot), \ldots, g_d(\cdot)]^{\mathrm{T}}$ denotes the transformation function that determines the family of a system. $g_i(\cdot)$ $(i = 1, \ldots, d)$ is defined by the following four functions:

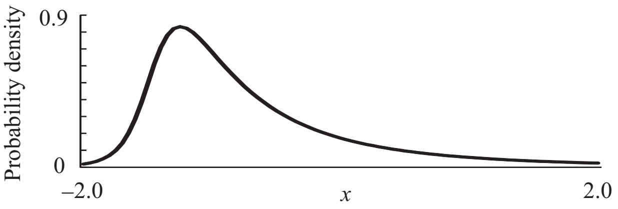


FIGURE 2. Probability density function of the Johnson SU distribution ($d = 1$, $\gamma_1 = -1.4$, $\delta_1 = 1.0$, $\lambda_1 = 0.3$, $\xi_1 = -1.5$). This distribution represents skewness and kurtosis, hence the shape becomes asymmetric.

$$g_i(y) = \begin{cases} \ln(y) & \text{for } S_{\mathrm{L}} \text{ (lognormal)} \\ \ln\left[y + \sqrt{y^2+1}\right] & \text{for } S_{\mathrm{U}} \text{ (unbounded)} \\ \ln\left[y/(1-y)\right] & \text{for } S_{\mathrm{B}} \text{ (bounded)} \\ y & \text{for } S_{\mathrm{N}} \text{ (normal)} \end{cases}. \tag{7}$$

The domains of x_i for S_{L}, S_{U}, S_{B}, and S_{N} are $(\xi, +\infty)$, $(-\infty, +\infty)$, $(\xi, \xi + \lambda)$, and $(-\infty, +\infty)$, respectively. In (6), the parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\xi}$ affect the location and scale of the distribution of $\boldsymbol{x}$, respectively. The combination of $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ are associated with skewness and kurtosis, and $\boldsymbol{g}[\cdot]$ determines the shape of distribution tails, i.e., whether the distribution tails are bounded or infinite.

Since EMG can be seen as a random process, systems with bounds are not suitable for EMG classification as probabilities cannot be calculated if an observation is out of bounds. In unbounded systems, S_{U} is expected to be an extension of the normal distribution in particular, and have enough flexibility to fit data from an arbitrary unimodal distribution. This paper, therefore, focuses on S_{U} as the form of the function $g_i(y)$. Fig. 2 shows an example of the Johnson S_{U} distribution, which can be calculated from the Johnson S_{U} translation system ($d = 1$). This asymmetric distribution represents skewness and kurtosis, and seems to be adaptable to the histogram of the smoothed EMG shown in Fig. 1(c).

C. POSTERIOR PROBABILITY ESTIMATION

To classify the vector $\boldsymbol{x} \in \mathbb{R}^d$ into one of the given C classes, we must examine the posterior probability $P(c|\boldsymbol{x})$ $(c = 1, \ldots, C)$. First, $\boldsymbol{x}$ is translated into a vector $\boldsymbol{z}^{(c)}$ using (6), which has the four parameters $\boldsymbol{\gamma}^{(c)}$, $\boldsymbol{\delta}^{(c)}$, $\boldsymbol{\lambda}^{(c)}$, and $\boldsymbol{\xi}^{(c)}$.

Assuming that the translated vector obeys a normal distribution, the posterior probability of $\boldsymbol{x}$ for class c is calculated as

$$P(c|\boldsymbol{x}) = \frac{P(c)P(\boldsymbol{x}|c)}{\sum_{c=1}^{C} P(c)P(\boldsymbol{x}|c)}, \tag{8}$$

$$P(\boldsymbol{x}|c) = \frac{|\mathbf{J}^{(c)}|}{(2\pi)^{\frac{d}{2}}|\Sigma^{(c)}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\boldsymbol{z}^{(c)\mathrm{T}}\Sigma^{(c)-1}\boldsymbol{z}^{(c)}\right), \tag{9}$$

where $P(c)$ is the prior probability of c, $\Sigma^{(c)}$ is the variance matrix of $\boldsymbol{z}^{(c)}$, and $\mathbf{J}^{(c)}$ is the $d \times d$ Jacobian matrix, whose

(i, j)th element is given by

$$\frac{\partial z_i^{(c)}}{\partial x_j} = \begin{cases} \delta_i^{(c)} {\lambda_i^{(c)}}^{-1} g'[{\lambda_i^{(c)}}^{-1}(x_i - \xi_i^{(c)})] & (i=j) \\ 0 & (i \neq j) \end{cases}, \quad (10)$$

where

$$g_i'(y) = \begin{cases} 1/y & \text{for } S_\mathrm{L} \text{ (lognormal)} \\ 1/\sqrt{y^2+1} & \text{for } S_\mathrm{U} \text{ (unbounded)} \\ 1/[y(1-y)] & \text{for } S_\mathrm{B} \text{ (bounded)} \\ 1 & \text{for } S_\mathrm{N} \text{ (normal)} \end{cases}. \quad (11)$$

The determinant of $\mathbf{J}^{(c)}$ is therefore calculated as

$$|\mathbf{J}^{(c)}| = \prod_{i=1}^{d} \frac{\partial z_i^{(c)}}{\partial x_i} = \prod_{i=1}^{d} \left\{ \frac{\delta_i^{(c)}}{\lambda_i^{(c)}} g' \left[\frac{(x_i - \xi_i^{(c)})}{\lambda_i^{(c)}} \right] \right\}. \quad (12)$$

D. LOG-LINEARIZATION

To incorporate the probabilistic model described above into a network structure, we transform the calculation of the Johnson translation and posterior probability estimation to linear combinations of coefficient matrices and input vectors.

First, let $\boldsymbol{y}^{(c)}$ be a calculation in the function $\boldsymbol{g}(\cdot)$ of (6). $\boldsymbol{y}^{(c)}$ is then transformed as follows:

$$\begin{aligned} \boldsymbol{y}^{(c)} &= {\boldsymbol{\lambda}^{(c)}}^{-1}(\boldsymbol{x} - \boldsymbol{\xi}^{(c)}) \\ &= {\boldsymbol{\lambda}^{(c)}}^{-1}\boldsymbol{x} - {\boldsymbol{\lambda}^{(c)}}^{-1}\boldsymbol{\xi}^{(c)} \\ &= \begin{bmatrix} -{\lambda_1^{(c)}}^{-1}\xi_1^{(c)} & {\lambda_1^{(c)}}^{-1} & & 0 \\ \vdots & & \ddots & \\ -{\lambda_d^{(c)}}^{-1}\xi_d^{(c)} & 0 & & {\lambda_d^{(c)}}^{-1} \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{x} \end{bmatrix} \\ &= {}^{(1)}{\mathbf{W}^{(c)}}^{\mathrm{T}}\mathbf{X}. \end{aligned} \quad (13)$$

Hence, $\boldsymbol{y}^{(c)}$ is expressed by multiplying the coefficient matrix ${}^{(1)}\mathbf{W}^{(c)} \in \mathbb{R}^{(d+1)\times d}$ and the augmented input vector $\mathbf{X} \in \mathbb{R}^{d+1}$.

The translated vector $\boldsymbol{z}^{(c)}$ is also transformed and expressed as the product of a coefficient matrix and an augmented vector as follows:

$$\begin{aligned} \boldsymbol{z}^{(c)} &= \boldsymbol{\gamma}^{(c)} + \boldsymbol{\delta}^{(c)}\boldsymbol{g}\left(\boldsymbol{y}^{(c)}\right) \\ &= \begin{bmatrix} \gamma_1^{(c)} & \delta_1^{(c)} & & 0 \\ \vdots & & \ddots & \\ \gamma_d^{(c)} & 0 & & \delta_d^{(c)} \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{g}\left(\boldsymbol{y}^{(c)}\right) \end{bmatrix} \\ &= {}^{(2)}{\mathbf{W}^{(c)}}^{\mathrm{T}}\mathbf{Y}^{(c)}, \end{aligned} \quad (14)$$

where ${}^{(2)}\mathbf{W}^{(c)} \in \mathbb{R}^{(d+1)\times d}$ is a coefficient matrix and $\mathbf{Y}^{(c)} \in \mathbb{R}^{d+1}$ is determined by the nonlinear transformation of $\boldsymbol{y}^{(c)}$.

Finally, setting

$$\zeta_c = P(c)P(\boldsymbol{x}|c) \quad (15)$$

and taking the log-linearization of ζ_c gives

$$\begin{aligned} \log \zeta_c = [&\log P(c) + \sum_{i=1}^{d} \log \frac{\delta_i^{(c)}}{\lambda_i^{(c)}} - \frac{1}{2}\log|\Sigma^{(c)}|, -\frac{1}{2}s_{1,1}^{(c)}, \\ &-s_{1,2}^{(c)}, \ldots, -\frac{1}{2}(2-\delta_{i,j})s_{i,j}^{(c)}, \ldots, -\frac{1}{2}s_{d,d}^{(c)}]\mathbf{Z}^{(c)} \\ &+ \log \prod_{i=1}^{d} g'(y_i^{(c)}) \\ = {}&{}^{(3)}{\mathbf{W}^{(c)}}^{\mathrm{T}}\mathbf{Z}^{(c)} + \sum_{i=1}^{d} \log g'(y_i^{(c)}), \end{aligned} \quad (16)$$

where $s_{1,1}^{(c)}, \ldots, s_{d,d}^{(c)}$ are elements of the inverse matrix ${\Sigma^{(c)}}^{-1}$, and $\delta_{i,j}$ is the Kronecker delta (1 if $i = j$, 0 otherwise). Note that $(2\pi)^{-\frac{d}{2}}$ in (9) is omitted because it is canceled out in (8). Additionally, $\mathbf{Z}^{(c)} \in \mathbb{R}^{H}$ ($H = 1 + \frac{d(d+1)}{2}$) is defined as

$$\begin{aligned} \mathbf{Z}^{(c)} = [&1, {z_1^{(c)}}^2, z_1^{(c)}z_2^{(c)}, \ldots, z_1^{(c)}z_d^{(c)}, \\ &{z_2^{(c)}}^2, z_2^{(c)}z_3^{(c)}, \ldots, {z_d^{(c)}}^2]. \end{aligned} \quad (17)$$

Taking the exponent of (16), ζ_c (i.e., the numerator of (8)) is ultimately expressed by

$$\zeta_c = \exp\left({}^{(3)}{\mathbf{W}^{(c)}}^{\mathrm{T}}\mathbf{Z}^{(c)} + \sum_{i=1}^{d} \log g'(y_i^{(c)})\right) \quad (18)$$

As outlined above, the Johnson translation and posterior probability estimation are calculated as linear combinations of coefficient matrices and nonlinearly transformed input vectors. If these coefficients are appropriately determined, the parameters and structure of the model can be defined, and therefore the posterior probability of the input vectors can be calculated for each class. The next section describes how the NN weight coefficients ${}^{(1)}\mathbf{W}^{(c)}$, ${}^{(2)}\mathbf{W}^{(c)}$, and ${}^{(3)}\mathbf{W}^{(c)}$ are determined via learning.

IV. PROPOSED NEURAL NETWORK

A. NETWORK STRUCTURE

Fig. 3 shows the structure of the proposed NN. This is a five-layer feedforward network with weight coefficients ${}^{(1)}\mathbf{W}^{(c)}$, ${}^{(2)}\mathbf{W}^{(c)}$, and ${}^{(3)}\mathbf{W}^{(c)}$ between the first/second, second/third, and fourth/fifth layers, respectively. A summation unit, identity unit, and multiplication unit are denoted by ◯, ⊘, and ⊗, respectively. Because of this structure, the output ${}^{(5)}O_c$ of this network estimates the posterior probability of each class c given $\boldsymbol{x}$, $P(c|\boldsymbol{x})$.

The first layer consists of $d+1$ units corresponding to the dimensions of the input data x. The relationship between the input and the output is defined as:

$$ {}^{(1)}I_i = \begin{cases} 1 & (i=0) \\ x_i & (i=1,\ldots,d) \end{cases}, \quad (19)$$

$$ {}^{(1)}O_i = {}^{(1)}I_i, \quad (20)$$

where ${}^{(1)}I_i$ and ${}^{(1)}O_i$ are the input and output of the ith unit, respectively. This layer corresponds to the construction of $\mathbf{X}$ in (13).

The second layer is composed of $C(d+1)$ units, each receiving the output of the first layer weighted by the coefficient ${}^{(1)}w_{i,j}^{(c)}$. The relationship between the input ${}^{(2)}I_{c,j}$

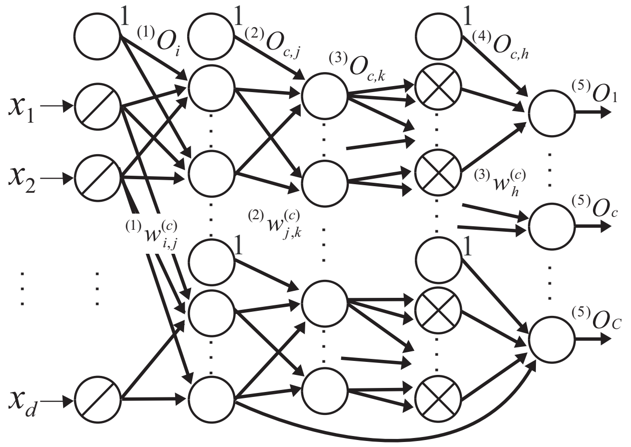


FIGURE 3. Structure of the proposed NN. This network consists of five layers and is constructed by incorporating the posterior probability calculation based on the Johnson S_{U} translation system into the network structure. A summation unit, identity unit, and multiplication unit are denoted by ◯, ⊘, and ⊗, respectively. The weight coefficients between the first/second layers and the second/third layers correspond to the parameters of the Johnson translation system. The weight coefficients between the fourth/fifth layers correspond to the probabilistic parameters, such as the prior probability and the variance matrix. Because of this structure, the output $^{(5)}O_c$ of this network estimates the posterior probability of each class c given $\boldsymbol{x}$, $P(c|\boldsymbol{x})$.

and the output $^{(2)}O_{c,j}$ of unit $\{c,j\}$ ($c = 1,\ldots,C$, $j = 1,\ldots,d+1$) is described as:

$$^{(2)}I_{c,j} = \sum_{i=0}^{d} {}^{(1)}w_{i,j}^{(c)}\,{}^{(1)}O_i, \tag{21}$$

$$^{(2)}O_{c,j} = \begin{cases} 1 & (j=0) \\ g(^{(2)}I_{c,j}) & (j=1,\ldots,d) \\ \sum_{j'=1}^{d} \log g'(^{(2)}I_{c,j'}) & (j=d+1) \end{cases}, \tag{22}$$

where the weight coefficient $^{(1)}w_{i,j}^{(c)}$ is an element of the matrix $^{(1)}\mathbf{W}^{(c)}$, which is given as:

$$^{(1)}\mathbf{W}^{(c)} = \begin{bmatrix} ^{(1)}w_{0,1}^{(c)} & \cdots & ^{(1)}w_{0,d}^{(c)} \\ ^{(1)}w_{1,1}^{(c)} & & 0 \\ & \ddots & \\ 0 & & ^{(1)}w_{d,d}^{(c)} \end{bmatrix}. \tag{23}$$

This layer is equal to the multiplication of $^{(1)}\mathbf{W}^{(c)}$ and $\mathbf{X}$ in (13), the construction of $\mathbf{Y}^{(c)}$ in (14), and the non-coefficient part of the Jacobian in (12).

The third layer is comprised of Cd units. The relationship between the input $^{(3)}I_{c,k}$ and the output $^{(3)}O_{c,k}$ is defined as:

$$^{(3)}I_{c,k} = \sum_{j=0}^{d} {}^{(2)}w_{j,k}^{(c)}\,{}^{(2)}O_{c,j}, \tag{24}$$

$$^{(3)}O_{c,k} = {}^{(3)}I_{c,k}, \tag{25}$$

where the weight coefficient $^{(2)}w_{j,k}^{(c)}$ is an element of the matrix $^{(2)}\mathbf{W}^{(c)}$, which can be written as:

$$^{(2)}\mathbf{W}^{(c)} = \begin{bmatrix} ^{(2)}w_{0,1}^{(c)} & \cdots & ^{(2)}w_{0,d}^{(c)} \\ ^{(2)}w_{1,1}^{(c)} & & 0 \\ & \ddots & \\ 0 & & ^{(2)}w_{d,d}^{(c)} \end{bmatrix}. \tag{26}$$

This layer corresponds to the multiplication of $^{(2)}\mathbf{W}^{(c)}$ and $\mathbf{Y}^{(c)}$ in (14).

The fourth layer has CH ($H = 1 + \frac{d(d+1)}{2}$) units. The relationship between the input $^{(4)}I_{c,h}$ and the output $^{(4)}O_{c,h}$ of the units $\{c,h\}$ ($h = 1,\ldots,H$) is defined as

$$^{(4)}I_{c,h} = \begin{cases} 1 \quad (h=1) \\ ^{(3)}O_{c,k}\,{}^{(3)}O_{c,k'} \\ \qquad (h = k' - \frac{1}{2}k^2 + (d+\frac{1}{2})k - d + 1) \end{cases}, \tag{27}$$

$$^{(4)}O_{c,h} = {}^{(4)}I_{c,h}, \tag{28}$$

where $k \leq k'$ ($k' = 1, \ldots, d$), and (27) corresponds to the nonlinear conversion shown in (17).

Finally, the fifth layer consists of C units, and its input $^{(5)}I_c$ and output $^{(5)}O_c$ are

$$^{(5)}I_c = \sum_{h=1}^{H} {}^{(3)}w_h^{(c)}\,{}^{(4)}O_{c,h} + {}^{(2)}O_{c,d+1}, \tag{29}$$

$$^{(5)}O_c = \frac{\exp\left(^{(5)}I_c\right)}{\sum_{c'=1}^{C} \exp\left(^{(5)}I_{c'}\right)}. \tag{30}$$

The output $^{(5)}O_c$ corresponds to the posterior probability for class c, $P(c|\boldsymbol{x})$. Here, the posterior probability $P(c|\boldsymbol{x})$ can be calculated if the NN coefficients $^{(1)}\mathbf{W}^{(c)}$, $^{(2)}\mathbf{W}^{(c)}$, and $^{(3)}\mathbf{W}^{(c)}$ have been appropriately established.

B. LEARNING ALGORITHM

This subsection describes a learning algorithm that can acquire a unique optimal solution without any hyperparameters. The learning algorithm consists of two steps.

In the first step, we estimate $^{(1)}\mathbf{W}^{(c)}$ and $^{(2)}\mathbf{W}^{(c)}$, which contain the parameters of the Johnson translation system. Although various parameter estimation algorithms have been proposed for the Johnson translation system, this paper adopts the percentile method [37] because it analytically estimates the parameters with a certain degree of accuracy. The percentile method calculates Johnson system parameters by comparing distances in the tails with distances in the central portion of the distribution. Using the percentile method, the parameters for the Johnson translation system are determined as follows:

$$\delta_i^{(c)} = \frac{2z}{\cosh^{-1}\left[\frac{1}{2}\left(\frac{m_i^{(c)}}{p_i^{(c)}} + \frac{n_i^{(c)}}{p_i^{(c)}}\right)\right]}, \tag{31}$$

$$\gamma_i^{(c)} = \delta_i^{(c)} \sinh^{-1}\left[\frac{\frac{n_i^{(c)}}{p_i^{(c)}} - \frac{m_i^{(c)}}{p_i^{(c)}}}{\left[2\left(\frac{m_i^{(c)} n_i^{(c)}}{(p_i^{(c)})^2} - 1\right)\right]^{\frac{1}{2}}}\right], \tag{32}$$

$$\lambda_i^{(c)} = \frac{2p_i^{(c)}\left(\frac{m_i^{(c)}n_i^{(c)}}{(p_i^{(c)})^2}-1\right)^{\frac{1}{2}}}{\left(\frac{m_i^{(c)}}{p_i^{(c)}}+\frac{n_i^{(c)}}{p_i^{(c)}}-2\right)\left(\frac{m_i^{(c)}}{p_i^{(c)}}+\frac{n_i^{(c)}}{p_i^{(c)}}+2\right)^{\frac{1}{2}}}, \quad (33)$$

$$\xi_i^{(c)} = \frac{x_{z,i}^{(c)}+x_{-z,i}^{(c)}}{2} + \frac{p_i^{(c)}\left(\frac{n_i^{(c)}}{p_i^{(c)}}-\frac{m_i^{(c)}}{p_i^{(c)}}\right)}{2\left(\frac{m_i^{(c)}}{p_i^{(c)}}+\frac{n_i^{(c)}}{p_i^{(c)}}-2\right)}, \quad (34)$$

where $z > 0$ is chosen depending on the number of data points, and $i = 1, \dots, d$, $c = 1, \dots, C$ are indices corresponding to the dimension and class, respectively. The variable $x_{\zeta,i}^{(c)}$ ($\zeta = -3z, -z, z, 3z$) is the P_ζth percentile of the ith dimension of training data for class c, where P_ζ is the percentage of the area in the normal distribution corresponding to ζ. Using percentiles, $m_i^{(c)}, n_i^{(c)}, p_i^{(c)}$ are calculated as:

$$m_i^{(c)} = x_{3z,i}^{(c)} - x_{z,i}^{(c)}, \quad (35)$$

$$n_i^{(c)} = x_{-z,i}^{(c)} - x_{-3z,i}^{(c)}, \quad (36)$$

$$p_i^{(c)} = x_{z,i}^{(c)} - x_{-z,i}^{(c)}. \quad (37)$$

For more detail, refer to [37]. Next, ${}^{(1)}\mathbf{W}^{(c)}$ and ${}^{(2)}\mathbf{W}^{(c)}$ can then be determined by substituting $\delta_i^{(c)}, \gamma_i^{(c)}, \lambda_i^{(c)}$, and $\xi_i^{(c)}$ in (13) and (14).

The second step concerns the discriminative learning of the remaining weight ${}^{(3)}\mathbf{W}^{(c)}$, which includes probabilistic parameters such as the prior probability $P(c)$ and covariance matrix $\Sigma^{(c)}$. A set of vectors $\boldsymbol{x}^{(n)}$ $(n = 1, \dots, N)$ is given for training, with the teacher vector $\mathbf{T}^{(n)} = [T_1^{(n)}, \dots, T_c^{(n)}, \dots, T_C^{(n)}]$ for the nth input. The training process of ${}^{(3)}\mathbf{W}^{(c)}$ involves minimizing the energy function E, which is defined as

$$E = \sum_{n=1}^{N} E_n = -\sum_{n=1}^{N}\sum_{c=1}^{C} T_c^{(n)} \log {}^{(5)}O_c^{(n)}, \quad (38)$$

to maximize the log-likelihood. Here, ${}^{(5)}O_c^{(n)}$ is the output for the input vector $\boldsymbol{x}^{(n)}$. The weight modification for ${}^{(3)}w_h^{(c)}$ based on Newton's method is defined as

$${}^{(3)}\mathbf{W}_{\mathrm{new}} = {}^{(3)}\mathbf{W}_{\mathrm{old}} - \mathbf{H}^{-1}\nabla E, \quad (39)$$

where ${}^{(3)}\mathbf{W}_{\mathrm{old}}$ and ${}^{(3)}\mathbf{W}_{\mathrm{new}}$ are the weight coefficients before and after weight modification, respectively, which have ${}^{(3)}\mathbf{W}^{(c)}$ in the cth block. ∇E_n is the gradient vector whose hth element in the cth block can be calculated as:

$$\begin{aligned}\frac{\partial E}{\partial {}^{(3)}w_h^{(c)}} &= \sum_{n=1}^{N} \frac{\partial E_n}{\partial {}^{(3)}w_h^{(c)}} \\ &= \sum_{n=1}^{N}\sum_{c'=1}^{C} \frac{\partial E_n}{\partial {}^{(5)}O_{c'}^{(n)}} \frac{\partial {}^{(5)}O_{c'}^{(n)}}{\partial {}^{(5)}I_c^{(n)}} \frac{\partial {}^{(5)}I_c^{(n)}}{\partial {}^{(3)}w_h^{(c)}} \\ &= \sum_{n=1}^{N} ({}^{(5)}O_c^{(n)} - T_c^{(n)}) {}^{(4)}O_{c,h}^{(n)}. \end{aligned} \quad (40)$$

TABLE 2. Parameters for data generation in the simulation experiment

	ξ	λ	δ	γ	Σ^{-1}
Class 1	0.15 0.7	0.04 0.05	0.9 0.8	-0.9 0.5	0.6
Class 2	0.5 0.55	0.05 0.01	0.8 0.5	0.5 -0.5	0.9

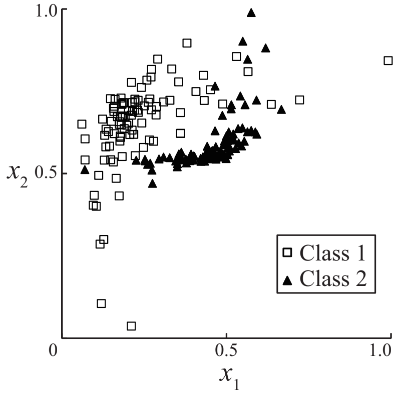


FIGURE 4. Scattergram of a dataset used in the simulation experiment. Each class had different skewness and kurtosis, which should be considered to accurately calculate the posterior probabilities.

H is the Hessian matrix comprised of $H \times H$ blocks, where the (h,l)-the element of the (c,k)-the is

$$\sum_{n=1}^{N} \frac{\partial^2 E_n}{\partial {}^{(3)}w_h^{(c)} \partial {}^{(3)}w_l^{(k)}} = \sum_{n=1}^{N} {}^{(5)}O_k^{(n)}(\delta_{c,k} - {}^{(5)}O_c^{(n)}) {}^{(4)}O_{c,h}^{(n)} {}^{(4)}O_{k,l}^{(n)}. \quad (41)$$

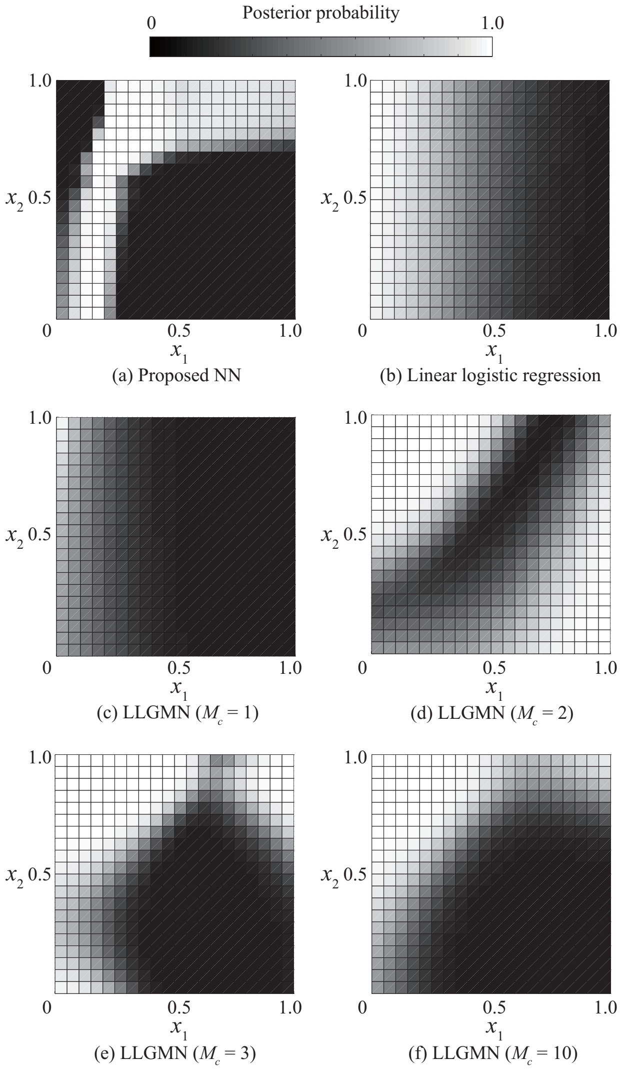
Note that H is positive semi-definite (see Appendix A). It follows that E is a convex function of ${}^{(3)}\mathbf{W}^{(c)}$ and hence has a unique minimum. Using this algorithm, the process of training the network converges to a unique solution without the need for any hyperparameters.

V. SIMULATION EXPERIMENT

A. METHOD

To verify that the proposed network can properly calculate the posterior probability for data with skewness and kurtosis, we performed a simulation experiment using two-dimensional ($d = 2$) two-class ($C = 2$) data. The data were artificially generated using the inverse of the multivariate Johnson S_{U} translation system [25]. An example of a dataset used in the experiments is shown in Fig. 4. The generated data simulated EMG signals, after preprocessing, with skewness and kurtosis. Table 2 lists the parameters used for each class in this generation. Each class had different skewness and kurtosis qualities, as well as different means and variances.

In the experiment, 100 samples of each class were assigned to the training dataset. The function $g_i(y)$ was of type S_{U} (unbounded). After training, the proposed NN was tested



(a) Proposed NN

(b) Linear logistic regression

(c) LLGMN (M_c = 1)

(d) LLGMN (M_c = 2)

(e) LLGMN (M_c = 3)

(f) LLGMN (M_c = 10)

FIGURE 5. Posterior probability of Class 1 ($P(c = 1|\boldsymbol{x})$) given by (a) the proposed NN, (b) LLR, and (c) LLGMN ($M_c = 1$), (d) LLGMN ($M_c = 2$), (e) LLGMN ($M_c = 3$), and (f) LLGMN ($M_c = 10$), where M_c is the number of components for a Gaussian mixture model used in the LLGMN. The probability of Class 2 ($P(c = 2|\boldsymbol{x})$) is the reverse of that of Class 1 with respect to black and white.

using inputs in the range $0 \leq x_1 \leq 1$ and $0 \leq x_2 \leq 1$. The corresponding posterior probabilities were compared with those given by LLR and LLGMN [22]. The LLR was trained using Newton's method [29], and LLGMN was trained by terminal learning [38] with an ideal convergence time of 1.0 and a learning sampling time of 0.001. The number of components M_c in the LLGMN was varied between 1 and 10.

B. RESULTS AND DISCUSSION

Fig. 5 shows the posterior probability of Class 1 ($P(c = 1|\boldsymbol{x})$) given by the proposed NN, LLR, and LLGMN. The probability of Class 2 ($P(c = 2|\boldsymbol{x})$) is clear from this graph, because it can be calculated as $P(c = 2|\boldsymbol{x}) = 1 - P(c = 1|\boldsymbol{x})$.

From Fig. 5 (a), it is clear that the posterior probability given by the proposed NN resembles the distribution shape of the experimental data of Class 1 (Fig. 4). The probability given by LLR (Fig. 5 (b)) differs completely from the experimental data distribution. Although LLGMN with $M_c = 1$ (Fig. 5 (c)) is also different from the experimental data distribution, this classifier produced a probability map that becomes closer to the experimental data as the number of components increased.

It can be inferred that the proposed NN is capable of appropriately dealing with data including skewness and kurtosis because it is based on the Johnson translation system. In contrast, LLR, a linear classifier, could not be adapted to data with skewness and kurtosis. LLGMN was capable of handling data with skewness and kurtosis when a sufficient number of components was used, but did not represent the data distribution well with too few components. The above results demonstrate that the proposed NN can handle data with skewness and kurtosis without hyperparameters, whereas conventional methods require hyperparameter optimization.

VI. EMG CLASSIFICATION EXPERIMENT

A. METHOD

To evaluate the suitability of the proposed network for real biological data, a classification experiment was conducted using EMG data. In this experiment, we evaluated the performance of the proposed NN from various aspects such as classification accuracy, computational time for hyperparameter tuning/training/prediction, and the effect of decreasing the number of electrodes EMG using datasets with various characteristics.

The procedure of the experiment is shown in Fig. 6. In the experiments, feature extraction was performed on raw EMG signals based on signal rectification and smoothing to extract the amplitude features of the EMGs. The extracted features were then classified using the proposed NN.

Table 3 shows the characteristics of six datasets prepared for this experiment in terms of the number of motions that the subjects performed, number of electrodes, number of subjects, number of trials for each subject, and number of samples for each trial. Details of the data acquisition will be described in the next subsection. The number of motions and the number of electrodes correspond to the number of classes and the number of input dimensions, respectively. The training samples were randomly chosen from the available samples for each trial, with the remaining samples used for testing. In real world applications, it is difficult to procure many training samples. Our aim was to evaluate the validity of the proposed NN for learning with limited training data; therefore, only 1% of the available samples were selected for training.

We compared the performance of the proposed NN with those of well-known classifiers and neural networks: ν-SVM [39] with a one-vs-one classifier, LLGMN [22], MLP, LLR, k-NN, and random forests [30]. The hyperparame-

TABLE 3. Summary of the datasets for the EMG classification experiment

Dataset	# Motions	# Electrodes	# Sub.	# Trials	# Samples
I	6	6	1	10	20000
II	14	8	8	4	36000
III	15	8	8	3	72000
IV	16	13	1	15	10000
V	17	12	9	6	12000
VI	6	2	5	30	2000

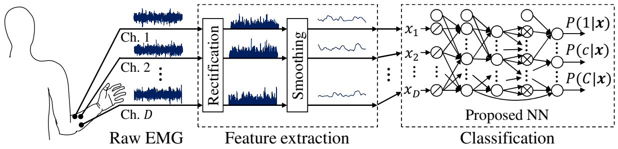


FIGURE 6. Procedure of the EMG classification experiment

ters for ν-SVM (γ and ν) were optimized using 10-fold cross-validation (CV) and a 10 × 10 grid search (γ ranged from $\log_{10} 5.0$ to $\log_{10} 1.0^{-5}$, and ν ranged from $\nu_{\max}$ to $\log_{10} 1.0^{-5}$ at even intervals in a logarithmic space, where $\nu_{\max}$ was dependent on the ratio of labels in the training data). LLGMN was trained by terminal learning [38] with an ideal convergence time of 1.0 and a learning sampling time of 0.001. The number of components (from 1 to 5) in the LLGMN was determined using 10-fold CV. The number of nodes (from d to $d + 10$) in the hidden layer of MLP was also determined using 10-fold CV, and MLP was trained using the backpropagation algorithm with a learning rate of 0.1. The LLR was trained using Newton's method, and the value of k in the k-NN algorithm was chosen from the range 1 to 10 using 10-fold CV. In the training of random forests, the size of the randomly selected subset of features at each tree node was determined based on the out-of-bag error. All algorithms were programmed using C++ and the dlib C++ Library [40]. The experiments were run on a computer with an Intel Core(TM) i7-7700K (4.2 GHz) processor and 16.0 GB RAM.

To evaluate the usefulness of a classifier in real-world applications, it is necessary to measure not only the classification accuracy, but also the training/preparation time and the prediction time. We therefore compared the performance of the above algorithms through six metrics: accuracy, F-measure, Cohen's kappa statistic, CV time, training time, and prediction time. The accuracy is defined by 100 × $N_{\text{correct}}/N_{\text{total}}$, where N_{correct} is the number of correctly classified test samples and N_{total} is the total number of test samples. The F-measure is the harmonic mean of precision and recall. To apply the F-measure to multi-class classification, we employed the micro F-measure [41]. The kappa statistic is defined by $(p_o - p_e)/(1 - p_e)$, where p_o is the relative observed agreement and p_e is the hypothetical probability of chance agreement. CV time is the total time taken for hyperparameter optimization based on CV, and training time is the time required for training. The sum of CV time and training time is the total time to prepare the classifier. Prediction time is the total time taken to classify all test samples. These metrics were measured for every trial and every subject, after which the average values were calculated.

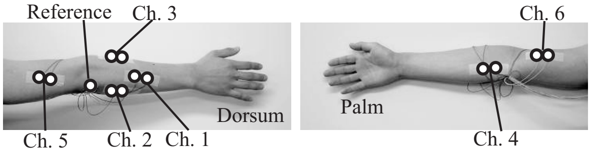


FIGURE 7. Locations of electrodes for Dataset I

B. DATA ACQUISITION

Dataset I contains six-channel ($d = 6$) EMG data recorded by the authors. The six pairs of electrodes were located as follows: Ch. 1: extensor carpi ulnaris; Ch. 2: flexor digitorum profundus; Ch. 3: extensor digitorum; Ch. 4: flexor carpi ulnaris; Ch. 5: triceps brachii; Ch. 6: biceps brachii (see Fig. 7). A healthy 22-year-old male subject performed six successive motions in a relaxed state ($C = 6$; M1: hand opening; M2: hand grasping; M3: wrist extension; M4: wrist flexion; M5: pronation; M6: supination). EMG signals were recorded at 1 kHz and quantized using a 16-bit A/D converter. Feature extraction was then conducted according to the method of [14]. The signals were rectified and smoothed using a second-order Butterworth low-pass filter with a cut-off frequency of 1 Hz. These features were defined as $EMG_i(n)$ ($i = 1, \ldots, d, n = 1, \ldots, N$: N is the number of data points) and normalized as follows:

$$x_i^{(n)} = \frac{EMG_i(n) - \overline{EMG_i^{\mathrm{st}}}}{\displaystyle\sum_{i'=1}^{d} \left(EMG_{i'}(n) - \overline{EMG_{i'}^{\mathrm{st}}} \right)}, \tag{42}$$

where $\overline{EMG_i^{\mathrm{st}}}$ is the mean of $EMG_i(n)$ in a state of muscular relaxation. Subsequently, $x_i^{(n)}$ was used as the input for the network.

Datasets II and III are those used in [42] and [43], respectively[1]. Dataset II contains measurements from eight subjects (aged from 20–35 years old) while seated on an armchair, putting their hands on a steering wheel attached to a desk and performing twelve classes of finger pressures and two classes of finger pointing (i.e., a total of 14 classes ($C = 14$)). It was reported that the method based on fuzzy neighborhood discriminant analysis proposed in [42] achieved less than a 7% error rate on average. For Dataset III, eight subjects (aged from 20–35 years old) performed fifteen classes of finger and hand movements while seated on an armchair with their arm supported and fixed in one position. In both these datasets, EMG signals were recorded using eight-channel electrodes at 4 kHz and quantized using a 12-bit A/D converter. Feature extraction for these datasets was conducted by rectification

[1] These datasets are available on Dr. Khushaba's webpage: http://www.rami-khushaba.com/electromyogram-emg-repository.html

and smoothing, as for Dataset I. The method proposed in the original paper [43] achieved over 95% accuracy on average.

Dataset IV contains measurements from a healthy 23-year-old male performing 16 forearm motions ($C = 16$). EMG signals were recorded using 13 pairs of electrodes ($d = 13$) at 1 kHz with a 60-Hz notch filter and a bandpass filter of 0.1–200 Hz. Details of the experimental conditions are described in [44]. Additionally, to evaluate the performance of the proposed NN for more difficult classification problems, we also confirmed the change of accuracy according to the decrease of the number of electrodes. The elimination of electrodes was conducted in order from the ones having the largest channel numbers, and then average classification accuracy and standard deviation were calculated for all the trials.

Dataset V is the Ninapro Database 3 exercise 1 [45], which is available on Ninaweb[2]. EMG signals were recorded from 11 trans-radial amputated subjects using 12 electrodes ($d = 12$) at 2 kHz while conducting 17 movements ($C = 17$). Each movement lasted five seconds and was repeated six times with a rest interval of three seconds. The signals of two subjects were recorded with less than 12 electrodes, hence only nine subjects' signals were used to ensure uniformity in the number of channels during the classification experiment. Feature extraction for these datasets was conducted by rectification and smoothing, as for other datasets. The original paper [45] reported that classification accuracy for this dataset using popular classifiers such as k-NN, SVM, random forests, and linear discriminant analysis was approximately 40%.

Dataset VI is the sEMG for basic hand movements dataset provided by Sapsanis *et al.* [46]. Five healthy subjects (two males and three females) were asked to perform six grasping movements ($C = 6$): holding a cylindrical tool, supporting a heavy load, holding a small tool, grasping with palm facing the object, holding a spherical tool, and holding a thin and flat object. Each movement lasted six seconds and was repeated 30 times. EMG signals were collected from two forearm surface EMG electrodes ($d = 2$) at a sampling rate of 500 Hz. Feature extraction was conducted in the same way as for other datasets, and the initial 1,000 samples for each movement were then discarded to remove transition states. For this dataset, classification accuracy was calculated based on 5×2 CV approach referring to the original paper that provided this dataset [46]. The average classification accuracies using the full training dataset reported in [46]–[48] were 89.21%, 87.78%, and 93.44%, respectively. The number of training data was also limited in this dataset by randomly sampling 1% of the training set in each fold.

C. RESULTS

Table 4 summarizes the results of EMG classification. The values show the average and standard deviation of scores measured for each trial of each subject, and are presented as "average value ± standard deviation" or "average value" if the standard deviation was 0. In the accuracy column, "$**$" denotes a significant difference, based on the Holm method, when compared to the proposed NN ($p < 0.01$). The absence of "$**$" in accuracy denotes no significant statistical difference. Fig. 8 shows the confusion matrix of the classification results for Dataset VI using the proposed NN.

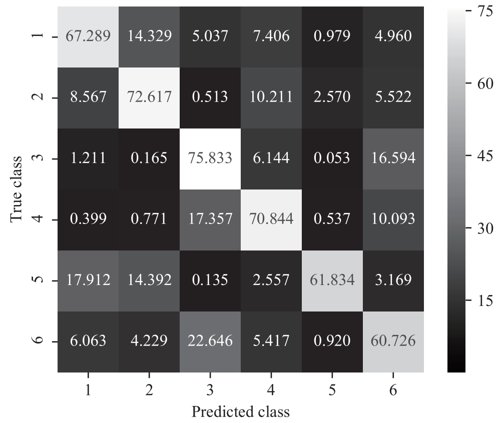


FIGURE 8. Confusion matrix of the classification results for Dataset VI. Values are normalized by the number of test samples for each class.

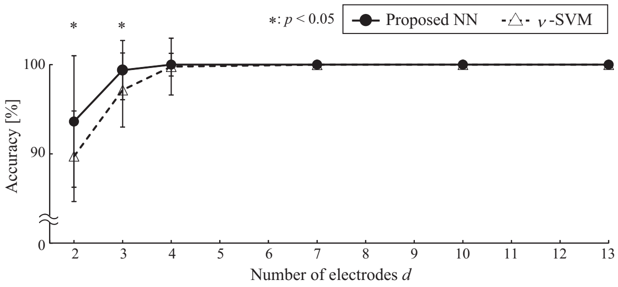


FIGURE 9. Classification accuracy for each number of electrodes (Dataset IV).

Fig. 9 shows the accuracy for each number of electrodes while decreasing the electrodes for Dataset IV. For comparison, the accuracies of ν-SVM are also plotted. Significant differences between the proposed NN and ν-SVM were confirmed when the number of electrodes was $d = 2$ and $d = 3$ ($p < 0.05$).

Fig. 10 shows the relationship between accuracy and preparation time (CV time + training time), representing the time until the classifier is ready, and between accuracy and prediction time for each classification method. Proximity to the upper-left corner indicates superior performance.

D. DISCUSSION

In terms of accuracy, the proposed NN, ν-SVM, and k-NN achieved the same level of performance, demonstrating suitability for EMG signal classification. The proposed NN involves Johnson distribution in its structure based on prior knowledge of EMG signals for appropriate modeling of EMG distribution in the network. ν-SVM showed strong gen-

[2] http://ninapro.hevs.ch/node/131

TABLE 4. Results of EMG classification

Dataset	Algorithm	Accuracy [%]	F-measure	Kappa	CV time [s]	Training time [s]	Prediction time [s]
I	Proposed NN	100	1	1	0	0.370 ± 0.015	0.525 ± 0.004
	SVM	100	1	1	23.390 ± 0.548	0.115 ± 0.017	29.221 ± 7.102
	LLGMN	100	1	1	197.497 ± 1.187	1.209 ± 0.011	0.318 ± 0.022
	MLP	100	1	1	207.315 ± 7.259	36.034 ± 0.580	0.070 ± 0.004
	LLR	99.504 ± 1.489	0.995 ± 0.015	0.994 ± 0.018	0	0.029 ± 0.007	0.303 ± 0.037
	k-NN	100	1	1	1.347 ± 0.037	0	15.231 ± 0.149
	Random Forest	100	1	1	4.084 ± 0.079	0.206 ± 0.006	2.562 ± 0.146
II	Proposed NN	100	1	1	0	29.025 ± 4.891	6.536 ± 0.104
	SVM	99.999 ± 0.005	1.000 ± 0.000	1.000 ± 0.000	522.975 ± 8.864	0.447 ± 0.349	131.309 ± 119.874
	LLGMN	99.198 ± 0.695 **	0.992 ± 0.007	0.991 ± 0.007	3074.678 ± 137.516	40.300 ± 24.863	4.392 ± 1.166
	MLP	96.689 ± 5.300 **	0.977 ± 0.035	0.964 ± 0.057	1609.333 ± 34.906	161.952 ± 3.923	0.418 ± 0.034
	LLR	95.908 ± 6.812 **	0.963 ± 0.063	0.956 ± 0.073	0	1.145 ± 0.619	2.792 ± 0.131
	k-NN	99.999 ± 0.002	1.000 ± 0.000	1.000 ± 0.000	26.992 ± 0.779	0	319.459 ± 11.172
	Random Forest	98.862 ± 1.643 **	0.989 ± 0.016	0.988 ± 0.018	43.713 ± 0.674	1.379 ± 0.024	19.971 ± 1.645
III	Proposed NN	99.973 ± 0.107	1.000 ± 0.001	1.000 ± 0.001	0	99.295 ± 39.217	15.235 ± 0.324
	SVM	99.726 ± 0.633	0.997 ± 0.006	0.997 ± 0.007	2454.330 ± 49.024	0.922 ± 0.614	254.447 ± 183.636
	LLGMN	96.131 ± 3.286 **	0.961 ± 0.033	0.959 ± 0.035	6844.277 ± 168.153	105.441 ± 55.835	10.436 ± 2.686
	MLP	83.631 ± 8.659 **	0.893 ± 0.069	0.825 ± 0.093	3483.782 ± 47.591	356.209 ± 7.832	1.021 ± 0.116
	LLR	90.590 ± 8.986 **	0.914 ± 0.087	0.899 ± 0.096	0	2.329 ± 1.951	6.904 ± 0.382
	k-NN	99.997 ± 0.005	1.000 ± 0.000	1.000 ± 0.000	144.351 ± 6.757	0	1740.607 ± 62.525
	Random Forest	97.252 ± 3.783	0.973 ± 0.038	0.971 ± 0.041	105.956 ± 2.710	3.357 ± 0.074	48.156 ± 3.105
IV	Proposed NN	100	1	1	0	66.226 ± 6.118	4.149 ± 0.037
	SVM	100	1	1	70.399 ± 1.417	0.144 ± 0.071	66.870 ± 66.167
	LLGMN	98.386 ± 2.022 **	0.984 ± 0.020	0.983 ± 0.022	2638.634 ± 48.720	26.279 ± 18.556	2.080 ± 0.850
	MLP	85.038 ± 11.362 **	0.943 ± 0.048	0.840 ± 0.121	786.566 ± 9.134	55.727 ± 1.386	0.181 ± 0.014
	LLR	97.906 ± 1.863 **	0.979 ± 0.019	0.978 ± 0.020	0	0.542 ± 0.527	1.042 ± 0.096
	k-NN	100	1	1	3.214 ± 0.087	0	36.481 ± 0.357
	Random Forest	100	1	1	31.526 ± 0.394	0.389 ± 0.017	5.136 ± 0.168
V	Proposed NN	39.122 ± 7.568	0.391 ± 0.076	0.353 ± 0.080	0	218.288 ± 19.043	6.511 ± 0.056
	SVM	39.169 ± 7.699	0.392 ± 0.077	0.354 ± 0.082	124.734 ± 1.403	0.065 ± 0.014	23.799 ± 5.279
	LLGMN	34.782 ± 7.297	0.348 ± 0.073	0.307 ± 0.078	2965.718 ± 76.741	40.293 ± 19.812	3.025 ± 0.937
	MLP	36.622 ± 6.492	0.369 ± 0.064	0.327 ± 0.069	960.015 ± 6.262	73.066 ± 1.389	0.274 ± 0.030
	LLR	33.848 ± 8.373	0.338 ± 0.084	0.297 ± 0.089	0	0.369 ± 0.098	1.291 ± 0.044
	k-NN	37.798 ± 7.455	0.378 ± 0.075	0.339 ± 0.079	4.974 ± 0.295	0	58.331 ± 1.936
	Random Forest	33.109 ± 8.264	0.331 ± 0.083	0.289 ± 0.088	39.456 ± 0.677	0.569 ± 0.032	8.446 ± 0.767
VI	Proposed NN	68.190 ± 10.014	0.682 ± 0.100	0.618 ± 0.120	0	0.049 ± 0.032	0.501 ± 0.017
	SVM	67.544 ± 9.073	0.675 ± 0.091	0.611 ± 0.109	31.037 ± 0.603	0.064 ± 0.041	7.821 ± 5.088
	LLGMN	65.852 ± 6.457	0.670 ± 0.082	0.590 ± 0.077	92.514 ± 2.071	1.609 ± 0.637	0.505 ± 0.066
	MLP	53.117 ± 7.198 **	0.627 ± 0.140	0.437 ± 0.086	171.200 ± 2.574	56.152 ± 0.988	0.115 ± 0.013
	LLR	70.992 ± 8.646	0.710 ± 0.086	0.652 ± 0.104	0	0.020 ± 0.006	0.454 ± 0.025
	k-NN	68.039 ± 8.658	0.680 ± 0.087	0.616 ± 0.104	2.766 ± 0.115	0	29.957 ± 0.669
	Random Forest	69.388 ± 9.645	0.694 ± 0.096	0.633 ± 0.116	1.321 ± 0.073	0.445 ± 0.042	5.349 ± 0.568

**: significant difference with the proposed NN ($p < 0.01$)

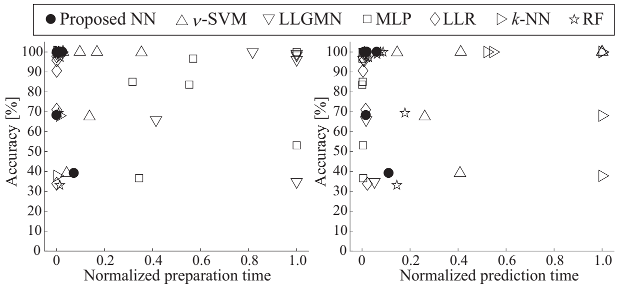


FIGURE 10. Relationships between (left) accuracy and preparation time (CV time + Training time) and between (right) accuracy and prediction time by using each classification method for six different datasets. The preparation time and the prediction time were normalized by the maximum value for each dataset. Proximity to the upper-left corner indicates superior performance. RF represents random forests.

eralization ability derived from margin maximization. k-NN can be used to express arbitrary complex decision boundaries based on the determination of the parameter k, ensuring a fit to the skewness and kurtosis of EMG data. On the other hand, the accuracies of LLGMN, MLP, LLR, and random forests were notably low in some cases. This is because LLGMN and MLP require many parameters to fit data with skewness and kurtosis, which resulted in over-fitting, and LLR could not solve nonlinear classification problems because it is a linear classifier. Random forests could not precisely represent the classification boundary of data with kurtosis and skewness because they make a classification using an ensemble of simple trees. For Dataset V, the accuracies were relatively low in all the algorithms. This is because this dataset is recorded from amputated subjects and therefore EMG signals were unstable, thereby lowering the reproducibility of motions compared with the data recorded from intact subjects. The accuracies for Dataset VI were also low because the number of electrodes is small for the number of motions. In addition, the classification accuracies for Dataset VI were lower than those of previous investigators' studies, such as [47], [48], because the number of training data was limited to 1% of the whole training set in this experiment.

It can be seen in Fig. 8 that there was frequent confusion between Class 3 and Class 6 when compared with other classes. This is reasonable since the motions of these classes are similar (class 3: holding a small tool, class 6: holding a thin and flat object). However, there was no extreme bias toward a certain class; therefore, the proposed NN worked properly for multi-class classification.

In Fig. 9, the accuracies of the proposed NN and ν-SVM both decreased according to the decrease in the number of electrodes. Since the decrease in electrodes yielded less information to classify the motions, the classification problem became increasingly difficult. In particular, the accuracies were sharply reduced when the number of channels was reduced $d = 3$ to $d = 2$, although the proposed NN exceeded ν-SVM. One possible explanation is that the substantial reduction of the input dimensions yielded the overlap of distribution for each class, and thus the proposed NN could not model the data distribution precisely.

With respect to CV time, LLGMN and MLP took particularly long. In contrast, the proposed NN and LLR had CV times of 0 because they have a unique solution of learning and therefore do not require hyperparameters such as a learning rate.

The training time for the proposed NN was relatively short for Dataset I. For Datasets II, III, IV, and V, however, significant training time was required. This can be attributed to the cost of calculating the Hessian matrix and finding its inverse (see (39) and (41)) that increases with the number of classes and input dimensions. Although the total time for the classifier to be ready was still relatively short (because the CV time is 0), there is room for improvement by making the numerical calculations more efficient.

Regarding the prediction time, ν-SVM and k-NN showed the longest times. Since ν-SVM was originally a binary classifier, it solves multi-class classification problems by calculating a two-class classification for all combinations. k-NN had to calculate the distance between the input sample and every training sample, resulting n a long computation time. The prediction time of the proposed NN was relatively short because it realized a compact model for EMG classification by incorporating prior knowledge of processed EMG characteristics.

Overall performance is summarized in Fig. 10. The plots of the proposed NN are concentrated toward the upper-left corner, demonstrating a well-balanced performance for accuracy and computational cost.

Finally, the performance of the proposed NN can be summarized as follows:

- High accuracy for classification of EMG signals.
- Relatively short time until the classifier becomes available.
- Shorter prediction time than ν-SVM and k-NN.

VII. CONCLUSION

In this paper, we proposed a NN based on the Johnson S_{U} translation system. The NN includes a discriminative model based on the multivariate Johnson S_{U} translation system, with the model transformed into linear combinations of weight coefficients and nonlinearly transformed input vectors. This enables the representation of more flexible distributions for data with skewness and kurtosis. Parameters describing the shape of the distribution can be determined as network coefficients via network learning. The proposed NN can be trained without hyperparameter optimization, and the training converges to a unique solution. In addition, the posterior probability of input vectors for each class can be calculated as the output of the NN.

In a simulation experiment, the proposed network was shown to be more suitable for data with skewness and kurtosis than a conventional GMM-based network and linear logistic regression. The applicability of the proposed NN to biosignal classification was also demonstrated by the results of an EMG classification experiment.

In future research, we will apply the proposed NN to other signals such as EEGs. We also plan to construct an expanded model of the proposed NN. In this study, the function $g_i(y)$, which determines the shape of the distribution, was examined only in relation to S_{U}, whereas future work will investigate other functions. Despite the assumption of S_{U} distribution, EMG data are occasionally distributed like a different type of distribution such as S_{B}. In such a situation, S_{U} distribution is used as an approximation. Although S_{U} distribution showed satisfactory classification accuracy even in a such situation, more detailed comparisons with other types of function and development of selection criteria for the distribution type are needed. Using a different type of function for each dimension will also enable the classification of multivariate biosignals, such as the combination of EMG and EEG. Furthermore, attempts will be made to improve the learning algorithm, and the training time will be shortened by contriving numerical calculations for the Hessian matrix. Complete discriminative learning for ${}^{(1)}W^{(c)}$ and ${}^{(2)}W^{(c)}$ will also be developed using backpropagation-based learning.

.

APPENDIX A POSITIVE DEFINITENESS OF THE HESSIAN MATRIX

This appendix shows that the Hessian matrix described in (39) is positive semi-definite. As described in (41), the (h, l)th element of the (c, k)th block of the matrix is given as

$$\sum_{n=1}^{N} \frac{\partial^2 E_n}{\partial^{(3)} w_h^{(c)} \partial^{(3)} w_l^{(k)}} = \sum_{n=1}^{N} {}^{(5)}O_k^{(n)} (\delta_{c,k} - {}^{(5)}O_c^{(n)}) {}^{(4)}O_{c,h}^{(n)} {}^{(4)}O_{k,l}^{(n)}. \quad (43)$$

For simplification, we can consider just one term in the summation over n, because the sum of positive semi-definite matrices is also positive semi-definite.

Consider an arbitrary vector $\boldsymbol{u} \in C \times H$ with elements $u_{c,h}$. Then,

$$\boldsymbol{u}^T H \boldsymbol{u}$$

$$
\begin{aligned}
&= \sum_{c,k}^{C}\sum_{h,l}^{H} u_{c,h}{}^{(5)}O_k(\delta_{c,k} - {}^{(5)}O_c){}^{(4)}O_{c,h}{}^{(4)}O_{k,l}u_{k,l} \\
&= \sum_{c,k}^{C} b_c{}^{(5)}O_k(\delta_{c,k} - {}^{(5)}O_c)b_k \\
&= \sum_{c,k}^{C} b_c b_k{}^{(5)}O_k\delta_{c,k} - \sum_{c,k}^{C} b_c{}^{(5)}O_c b_k{}^{(5)}O_k \\
&= \sum_{k}^{C} b_k^2{}^{(5)}O_k - (\sum_{k}^{C} b_k{}^{(5)}O_k)^2, \qquad (44)
\end{aligned}
$$

where

$$b_c = \sum_{h}^{H} u_{c,h}{}^{(4)}O_{c,h}, \qquad (45)$$

$$b_k = \sum_{l}^{H} u_{k,l}{}^{(4)}O_{k,l}. \qquad (46)$$

Here, ${}^{(5)}O_k$ is the posterior probability satisfying $0 \leq {}^{(5)}O_k \leq 1$ and $\sum_k {}^{(5)}O_k = 1$. Furthermore, the function $f(b_c) = b_c^2$ is a convex function. Hence, we can apply Jensen's inequality [49] to give

$$
\begin{aligned}
\sum_{k}^{C} b_k^2{}^{(5)}O_k &= \sum_{k}^{C} f(b_k){}^{(5)}O_k \\
&\geq f(\sum_{k}^{C} b_k{}^{(5)}O_k) = (\sum_{k}^{C} b_k{}^{(5)}O_k)^2 \qquad (47)
\end{aligned}
$$

Therefore,

$$\boldsymbol{u}^T H \boldsymbol{u} \geq 0. \qquad (48)$$

From the definition of definiteness, the Hessian matrix H is positive semi-definite.

REFERENCES

[1] N. Mammone, C. Ieracitano, H. Adeli, A. Bramanti, and F. C. Morabito, "Permutation jaccard distance-based hierarchical clustering to estimate EEG network density modifications in MCI subjects," IEEE Trans. Neural Netw. Learn. Syst., 2018.

[2] S. Sakhavi, C. Guan, and S. Yan, "Learning temporal information for brain-computer interface using convolutional neural networks," IEEE Trans. Neural Netw. Learn. Syst., 2018.

[3] H. Zeng and A. Song, "Optimizing single-trial EEG classification by stationary matrix logistic regression in brain–computer interface," IEEE Trans. Neural Netw. Learn. Syst., vol. 27, no. 11, pp. 2301–2313, 2016.

[4] Y. Zhang, G. Zhou, J. Jin, Q. Zhao, X. Wang, and A. Cichocki, "Sparse bayesian classification of EEG for brain–computer interface," IEEE Trans. Neural Netw. Learn. Syst., vol. 27, no. 11, pp. 2256–2267, 2016.

[5] Y. Geng, Y. Ouyang, O. W. Samuel, S. Chen, X. Lu, C. Lin, and G. Li, "A robust sparse representation based pattern recognition approach for myoelectric control," IEEE Access, vol. 6, pp. 38 326–38 335, 2018.

[6] A. Furui, H. Hayashi, and T. Tsuji, "An EMG pattern classification method based on a mixture of variance distribution models," in Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Bio. Soc. (EMBC), 2018, pp. 5216–5219.

[7] S. Raurale, J. McAllister, and J. M. del Rincon, "EMG wrist-hand motion recognition system for real-time embedded platform," in IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP), 2019, pp. 1523–1527.

[8] F. Xiao, Y. Wang, L. He, H. Wang, W. Li, and Z. Liu, "Motion estimation from surface electromyogram using adaboost regression and average feature values," IEEE Access, vol. 7, pp. 13 121–13 134, 2019.

[9] T.-Y. Pan, W.-L. Tsai, C.-Y. Chang, C.-W. Yeh, and M.-C. Hu, "A hierarchical hand gesture recognition framework for sports referee training-based EMG and accelerometer sensors," IEEE Trans. Cybern., pp. 1–12, 2020.

[10] J. Kobylarz, J. J. Bird, D. R. Faria, E. P. Ribeiro, and A. Ekárt, "Thumbs up, thumbs down: non-verbal human-robot interaction through real-time EMG classification via inductive and supervised transductive transfer learning," Journal of Ambient Intelligence and Humanized Computing, vol. 11, pp. 6021–6031, 2020.

[11] A. Subasi and S. M. Qaisar, "Surface EMG signal classification using TQWT, bagging and boosting for hand movement recognition," Journal of Ambient Intelligence and Humanized Computing, pp. 1–16, 2020.

[12] T. Tuncer, S. Dogan, and A. Subasi, "Surface EMG signal classification using ternary pattern and discrete wavelet transform based feature extraction for hand movement recognition," Biomedical Signal Processing and Control, vol. 58, p. 101872, 2020.

[13] W. Tigra, B. Navarro, A. Cherubini, X. Gorron, A. Gélis, C. Fattal, D. Guiraud, and C. Azevedo-Coste, "A novel EMG interface for individuals with tetraplegia to pilot robot hand grasping," IEEE Trans. Neural Syst. Rehabili. Eng., vol. 26, no. 2, pp. 291–298, 2016.

[14] O. Fukuda, T. Tsuji, M. Kaneko, and A. Otsuka, "A human-assisting manipulator teleoperated by EMG signals and arm motions," IEEE Trans. Robot. Autom., vol. 19, no. 2, pp. 210–222, 2003.

[15] M. A. Oskoei and H. Hu, "Myoelectric control systems–a survey," Biomed. Signal Process. and Control, vol. 2, no. 4, pp. 275–294, 2007.

[16] C. Cortes and V. Vapnik, "Support-vector networks," Mach. learning, vol. 20, no. 3, pp. 273–297, 1995.

[17] D. E. Rumelhart, G. E. Hintont, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, no. 9, pp. 533–536, 1986.

[18] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Trans. Inf. Theory, vol. 13, no. 1, pp. 21–27, 1967.

[19] Q. Wang, P. Li, and L. Zhang, "G2denet: Global gaussian distribution embedding network and its application to visual recognition," in Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR), 2017, pp. 2730–2739.

[20] H. Hayashi, T. Shibanoki, K. Shima, Y. Kurita, and T. Tsuji, "A recurrent probabilistic neural network with dimensionality reduction based on time-series discriminant component analysis," IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 12, pp. 3021–3033, 2015.

[21] T. Tsuji, N. Bu, O. Fukuda, and M. Kaneko, "A recurrent log-linearized gaussian mixture network," IEEE Trans. Neural Netw., vol. 14, no. 2, pp. 304–316, 2003.

[22] T. Tsuji, O. Fukuda, H. Ichinobe, and M. Kaneko, "A log-linearized gaussian mixture network and its application to EEG pattern classification," IEEE Trans. Syst. Man Cybern. C, Appl. Rev., vol. 29, no. 1, pp. 60–72, 1999.

[23] A. Furui, T. Igaue, and T. Tsuji, "Emg pattern recognition via bayesian inference with scale mixture-based stochastic generative models," Expert Systems with Applications, vol. 185, p. 115644, 2021.

[24] N. L. Johnson, "Systems of frequency curves generated by methods of translation," Biometrika, vol. 36, no. 1/2, pp. 149–176, 1949.

[25] P. M. Stanfield, J. R. Wilson, G. A. Mirka, N. F. Glasscock, J. P. Psihogios, and J. R. Davis, "Multivariate input modeling with johnson distributions," in Proc. 28th Conf. Winter Simulation. IEEE Computer Society, 1996, pp. 1457–1464.

[26] N. L. Johnson, "Bivariate distributions based on simple translation systems," Biometrika, vol. 36, no. 3/4, pp. 297–304, 1949.

[27] N. L. Johnson and S. Kotz, Distributions in statistics, continuous multivariate distributions. Wiley, 1972.

[28] H. Hayashi, Y. Kurita, and T. Tsuji, "A non-gaussian approach for biosignal classification based on the johnson su translation system," in Proc. IEEE 8th Int. Workshop on Comput. Intell. and Appl. (IWCIA). IEEE, 2015, pp. 115–120.

[29] C. M. Bishop and N. M. Nasrabadi, Pattern Recognition and Machine Learning. Springer New York, 2006, vol. 1.

[30] L. Breiman, "Random forests," Machine learning, vol. 45, pp. 5–32, 2001.

[31] N. Hogan and R. W. Mann, "Myoelectric signal processing: Optimal estimation applied to electromyography-part i: Derivation of the optimal myoprocessor," IEEE Trans. Biomed. Eng., vol. 27, no. 7, pp. 382–395, 1980.

[32] R. N. Khushaba, A. H. Al-Timemy, A. Al-Ani, and A. Al-Jumaily, "A framework of temporal-spatial descriptors-based feature extraction for

improved myoelectric pattern recognition," IEEE Trans. Neural Syst. Rehabil. Eng., vol. 25, no. 10, pp. 1821–1831, 2017.

[33] N. Nazmi, M. A. Abdul Rahman, S.-I. Yamamoto, S. A. Ahmad, H. Zamzuri, and S. A. Mazlan, "A review of classification techniques of EMG signals during isotonic and isometric contractions," Sensors, vol. 16, no. 8, p. 1304, 2016.

[34] C. Altın and O. Er, "Comparison of different time and frequency domain feature extraction methods on elbow gesture's EMG," EJIS European J. Interdisciplinary Stud. Articles, vol. 5, 2016.

[35] E. J. Rechy-Ramirez and H. Hu, "Stages for developing control systems using EMG and EEG signals: A survey," Technical Report: CES-513 in School of Comput. Sci. and Electron. Eng., University of Essex, United Kingdom, Tech. Rep., 2011.

[36] T.-H. Kim and H. White, "On more robust estimation of skewness and kurtosis," Finance Research Lett., vol. 1, no. 1, pp. 56–73, 2004.

[37] J. F. Slifker and S. S. Shapiro, "The Johnson system: selection and parameter estimation," Technometrics, vol. 22, no. 2, pp. 239–246, 1980.

[38] M. Zak, "Terminal attractors in neural networks," Neural Netw., vol. 2, no. 4, pp. 259–274, 1989.

[39] C. C. Chang and C. J. Lin, "Training ν-support vector classifiers: theory and algorithms," Neural computation, vol. 13, no. 9, pp. 2119–2147, 2001.

[40] D. E. King, "Dlib-ml: A machine learning toolkit," The J. Mach. Learning Research, vol. 10, pp. 1755–1758, 2009.

[41] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Information processing & management, vol. 45, no. 4, pp. 427–437, 2009.

[42] R. N. Khushaba, S. Kodagoda, D. Liu, and G. Dissanayake, "Muscle computer interfaces for driver distraction reduction," Comput. Methods and Programs in Biomedicine, vol. 110, no. 2, pp. 137–149, 2013.

[43] R. N. Khushaba and S. Kodagoda, "Electromyogram (EMG) feature reduction using mutual components analysis for multifunction prosthetic fingers control," in Proc. 12th Int. Conf. Control Automat. Robot & Vision (ICARCV), 2012, pp. 1534–1539.

[44] T. Shibanoki, K. Shima, T. Tsuji, A. Otsuka, and T. Chin, "A quasi-optimal channel selection method for bioelectric signal classification using a partial kullback–leibler information measure," IEEE Trans. Biomed. Eng., vol. 60, no. 3, pp. 853–861, 2013.

[45] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," Scientific Data, vol. 1, 2014.

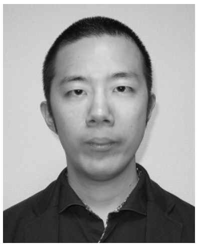
[46] C. Sapsanis, G. Georgoulas, A. Tzes, and D. Lymberopoulos, "Improving EMG based classification of basic hand movements using EMD," in Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Bio. Soc. (EMBC),. IEEE, 2013, pp. 5754–5757.

[47] A. Subasi, A. Alaskandarani, A. A. Abubakir, and S. M. Qaisar, "sEMG signal classification using DWT and bagging for basic hand movements," in Proc. 21st Saudi Comput. Society National Comput. Conf. IEEE, 2018, pp. 1–6.

[48] A. Subasi, L. Alharbi, R. Madani, and S. M. Qaisar, "Surface EMG based classification of basic hand movements using rotation forest," in Proc. Advances in Science and Eng. Technology Int. Conf. IEEE, 2018, pp. 1–5.

[49] J. L. W. V. Jensen, "Sur les fonctions convexes et les inégalités entre les valeurs moyennes," Acta Mathematica, vol. 30, no. 1, pp. 175–193, 1906.

HIDEAKI HAYASHI (S' 13–M' 16) received B.E., M.Eng, and D.Eng. degrees from Hiroshima University, Hiroshima, Japan, in 2012, 2014, and 2016 respectively. He was a Research Fellow of the Japan Society for the Promotion of Science from 2015 to 2017. He is currently an assistant professor in Department of Advanced Information Technology, Kyushu University. His current research interests focus on biosignal analysis, neural networks, and machine learning.



TARO SHIBANOKI (S' 11–M' 13) received the B.E. degree from the University of Tokushima, Tokushima, Japan, in 2008, and the M.E. and D.Eng. degrees from Hiroshima University, Hiroshima, Japan, in 2010 and 2012, respectively. He was a Research Fellow of the Japan Society for the Promotion of Science since 2011 and an Assistant Professor (Special Appointment) in Hiroshima University since 2013. He was a lecturer in College of Engineering, Ibaraki University from 2014 to 2020. He is currently an associate professor in Graduate School of Natural Science and Technology, Okayama University. His current research interests focus on dimension elimination, biological signal processing, and human–machine interface.

TOSHIO TSUJI (A' 88–M' 99) received the B.E. degree in industrial engineering, and the M.E. and D.Eng. degrees in systems engineering from Hiroshima University, Hiroshima, Japan, in 1982, 1985, and 1989, respectively. He was a Research Associate from 1985 to 1994, and an Associate Professor from 1994 to 2002, in the Faculty of Engineering, Hiroshima University. He was a Visiting Professor at the University of Genova, Genova, Italy, from 1992 to 1993. He is currently a Professor in the Department of System Cybernetics, Hiroshima University. His current research interests focus on human-machine interface and computational neural sciences, in particular, biological motor control. Dr. Tsuji won the Best Paper Award from the Society of Instrumentation and Control Engineers in 2002, and the K. S. Fu Memorial Best Transactions Paper Award of the IEEE Robotics and Automation Society in 2003. He is a member of the Japan Society of Mechanical Engineers, the Robotics Society of Japan, and the Society of Instrument and Control Engineers in Japan.