A Neural Network Based on the Johnson $S_U$ Translation System and Related Application to Electromyogram Classification

HIDEAKI HAYASHI (Member, IEEE), TARO SHIBANOKI (Member, IEEE), AND TOSHIO TSUJI (Member, IEEE)

1Department of Advanced Information Technology, Kyushu University, Fukuoka 819-0395, Japan
2Graduate School of Natural Science and Technology, Okayama University, Okayama 700-8530, Japan
3Department of System Cybernetics, Graduate School of Engineering, Hiroshima University, Higashihiroshima 739-8527, Japan

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

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP17K12752 and Grant JP21H03511, and in part by the Japan Science and Technology Agency (JST) ACT-I under Grant JPMJPR18U0.

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

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 high 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 1 summarizes the characteristics of the classification algorithms. The SVM is a distinguished classifier that 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 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 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. Many feature extraction methods have been proposed [32]–[34], and we focus on the method by 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 \( \sigma \). 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

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

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

\[
M_y = \int_{-\infty}^{\infty} y p(y) dy = \sqrt{\frac{2}{\pi}} \sigma, \\
V_y = \int_{-\infty}^{\infty} (y - M_y)^2 p(y) dy = \left(1 - \frac{2}{\pi}\right)\sigma^2.
\]
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]:

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

(4)

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

(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.

### B. Multivariate Johnson translation system

The Johnson 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 $x \in \mathbb{R}^d$ with skewness and kurtosis. The multivariate Johnson translation system [25] involves the normalizing translation:

$$z = \gamma + \delta g(\lambda^{-1}(x - \xi)) \sim N(0, \Sigma),$$

(6)

where $z$ is a random vector obeying a normal distribution with mean 0 and variance $\Sigma$, $\gamma \equiv [\gamma_1, \ldots, \gamma_d]^T$ and $\delta \equiv \text{diag}[\delta_1, \ldots, \delta_d]$ are shape parameters, $\lambda \equiv \text{diag}[\lambda_1, \ldots, \lambda_d]$ is a scale parameter, $\xi \equiv [\xi_1, \ldots, \xi_d]^T$ is a location parameter, and $g(\cdot) \equiv [g_1(\cdot), \ldots, g_d(\cdot)]^T$ 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:

$$g_i(y) = \begin{cases} 
\ln(y) & \text{for } S_L \text{ (lognormal)} \\
\ln\left[\frac{y + \sqrt{y^2 + 1}}{2}\right] & \text{for } S_U \text{ (unbounded)} \\
\ln\left[\frac{y}{(1 - y)}\right] & \text{for } S_B \text{ (bounded)} \\
y & \text{for } S_N \text{ (normal)} 
\end{cases},$$

(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 $\lambda$ and $\xi$ affect the location and scale of the distribution of $x$, respectively. The combination of $\gamma$ and $\delta$ are associated with skewness and kurtosis, and $g(\cdot)$ 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 $g_i(\cdot)$. 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 $x \in \mathbb{R}^d$ into one of the given $C$ classes, we must examine the posterior probability $P(c|x)$ ($c = 1, \ldots, C$). First, $x$ is translated into a vector $z^{(c)}$ using (6), which has the four parameters $\gamma^{(c)}$, $\delta^{(c)}$, $\lambda^{(c)}$, and $\xi^{(c)}$.

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

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

(8)

$$P(x|c) = \frac{|J^{(c)}|}{(2\pi)^{\frac{d}{2}}|\Sigma^{(c)}|^\frac{1}{2}} \exp\left(-\frac{1}{2}(x - \mu^{(c)})^T (\Sigma^{(c)})^{-1} (x - \mu^{(c)})\right).$$

(9)

$J^{(c)}$ is the Jacobian of the transformation $g(x)$. $\sum_{c=1}^{C} P(c)$ is the prior probability of each class.
where $P(c)$ is the prior probability of $c$, $\Sigma(c)$ is the variance matrix of $z(c)$, and $J^{(c)}$ is the $d \times d$ Jacobian matrix, whose $(i,j)$th element is given by
\[
\frac{\partial z_i^{(c)}}{\partial x_j} = \begin{cases} 
\frac{\delta_i^{(c)}}{\lambda_i} (\lambda_i^{-1} g'[(\lambda_i^{-1} (x_i - \xi_i^{(c)})] & (i = j), \\
0 & (i \neq j),
\end{cases}
\]
where
\[
g'_c(y) = \begin{cases} 
1/y & \text{for } S_L \text{ (lognormal)} \\
1/\sqrt{y^2 + 1} & \text{for } S_U \text{ (unbounded)} \\
1/|y(1-y)| & \text{for } S_B \text{ (bounded)} \\
1 & \text{for } S_N \text{ (normal)}.
\end{cases}
\]
The determinant of $J^{(c)}$ is therefore calculated as
\[
|J^{(c)}| = \prod_{i=1}^{d} \frac{\partial z_i^{(c)}}{\partial x_i} = \prod_{i=1}^{d} \frac{\delta_i^{(c)}}{\lambda_i} g'[(x_i - \xi_i^{(c)})].
\]

**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 $y^{(c)}$ be a calculation in the function $g^{(c)}$ of (6). $y^{(c)}$ is then transformed as follows:
\[
y^{(c)} = \lambda^{(c)-1} (x - \xi^{(c)})
= \lambda^{(c)-1} x - \lambda^{(c)-1} \xi^{(c)}
= \begin{bmatrix} -\lambda^{(c)-1} \xi_1^{(c)} & \lambda^{(c)-1} & 0 \\
-\lambda^{(c)-1} \xi_2^{(c)} & 0 & \lambda^{(c)-1} \\
\vdots & \ddots & \ddots \\
-\lambda^{(c)-1} \xi_d^{(c)} & 0 & \lambda^{(c)-1}
\end{bmatrix} \begin{bmatrix} x_1 \\
x_2 \\
\vdots \\
x_d \\
1
\end{bmatrix}
= (1^{(c)} \mathbf{W}^{(c)})^T \mathbf{X}.
\]

Hence, $y^{(c)}$ 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 $z^{(c)}$ is also transformed and expressed as the product of a coefficient matrix and an augmented vector as follows:
\[
z^{(c)} = y^{(c)} + \delta^{(c)} g(y^{(c)})
= \begin{bmatrix} y_1^{(c)} \\
\vdots \\
y_d^{(c)} \\
1
\end{bmatrix}
+ \begin{bmatrix} \delta_1^{(c)} \\
\vdots \\
\delta_d^{(c)} \\
1
\end{bmatrix} \begin{bmatrix} g(1) \\
\vdots \\
1
\end{bmatrix}
= (2)\mathbf{W}^{(c)} \mathbf{Y}^{(c)},
\]
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 $y^{(c)}$.

Finally, setting
\[
\zeta^{(c)} = P(c)P(x|c)
\]
and taking the log-linearization of $\zeta^{(c)}$ gives
\[
\log \zeta^{(c)} = [\log P(c) + \sum_{i=1}^{d} \log \frac{\delta_i^{(c)}}{\lambda_i^{(c)}}] - \frac{1}{2} \log \|\Sigma(c)\| - \frac{1}{2} \delta_1^{(c)} s_1^{(c)} - \cdots - \frac{1}{2} \delta_d^{(c)} s_d^{(c)}|Z^{(c)}
+ \log \prod_{i=1}^{d} g'(y_i^{(c)})
= (3)\mathbf{W}^{(c)}^T \mathbf{Z}^{(c)} + \sum_{i=1}^{d} \log g'(y_i^{(c)}),
\]
where $s_1^{(c)}, \ldots, s_d^{(c)}$ are elements of the inverse matrix $\Sigma(c)^{-1}$, and $\delta_{ij}$ is the Kronecker delta (1 if $i = j$, 0 otherwise).

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 bias unit, nonlinear unit, linear unit, and multiplication unit are denoted by $\bullet$, $\circ$, $\otimes$, and $\odot$, respectively. Because of this structure, the output $(5)O_c$ of this network estimates the posterior probability of each class $c$ given $x$, $P(c|x)$.

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:
\[
(1)I_i = \begin{cases} 
1 & (i = 0) \\
x_i & (i = 1, \ldots, d),
\end{cases}
(19)
\]
\[
(1)O_i = (1)I_i,
(20)
\]
The third layer is comprised of $Cd$ units. The relationship between the input $(3)I_{c,k}$ and the output $(5)O_{c,k}$ is defined as:

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

$$(3)O_{c,k} = (3)I_{c,k},$$

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

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

This layer corresponds to the multiplication of $(2)W^{(c)}$ and $Y^{(c)}$ 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

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

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

$$(5)I_{c} = \sum_{h=1}^{H}(3)w_{h}^{(c)}(4)O_{c,h} + (2)O_{c,d+1},$$

$$(5)O_{c} = \exp((5)I_{c}) \sum_{c'=1}^{C} \exp((5)I_{c'}).$$

The output $(5)O_{c}$ corresponds to the posterior probability for class $c, P(c|x)$. Here, the posterior probability $P(c|x)$ can be calculated if the NN coefficients $(1)W^{(c)}$, $(2)W^{(c)}$, and $(3)W^{(c)}$ 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 $(1)W^{(c)}$ and $(2)W^{(c)}$, 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

![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 bias unit, nonlinear unit, linear unit, and multiplication unit are denoted by $\cdot$, $\odot$, $\circ$, and $\otimes$, 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 $x, P(c|x)$.](image-url)
parameters for the Johnson translation system are determined as follows:

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

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

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

\[
\xi_i^{(c)} = \frac{x_{i,-}^{(c)} + x_{i,3}^{(c)}}{2} + \frac{p_i^{(c)} \left( \frac{n_i^{(c)}}{p_i^{(c)}} - \frac{m_i^{(c)}}{p_i^{(c)}} \right)}{2 \left( \frac{m_i^{(c)}}{p_i^{(c)}} + \frac{n_i^{(c)}}{p_i^{(c)}} - 2 \right)} \tag{34}
\]

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

\[
m_i^{(c)} = x_{3i,c}^{(c)} - x_{i,c}^{(c)}, \tag{35}
\]

\[
n_i^{(c)} = x_{-3i,c}^{(c)} - x_{-i,c}^{(c)}, \tag{36}
\]

\[
p_i^{(c)} = x_{c,c}^{(c)} - x_{-c,c}^{(c)} \tag{37}
\]

For more detail, refer to [37]. Next, \( W_1^{(c)} \) and \( W_2^{(c)} \) can then be determined by substituting \( \delta_i^{(c)}, y_i^{(c)}, \lambda_i^{(c)}, \) and \( \xi_i^{(c)} \) in (13) and (14).

The second step concerns the discriminative learning of the remaining weight \( W_{(3)}^{(c)} \), which includes probabilistic parameters such as the prior probability \( P(c) \) and covariance matrix \( \Sigma^{(c)} \). A set of vectors \( x^{(n)} (n = 1, \ldots, N) \) is given for training, with the teacher vector \( T^{(n)} = [T^{(n)}_1, \ldots, T^{(n)}_C] \) for the \( n \)th input. The training process of \( W_{(3)}^{(c)} \) involves minimizing the energy function \( E \), which is defined as

\[
E = \sum_{n=1}^{N} E_n = -\sum_{n=1}^{N} \sum_{i=1}^{C} T_{i,n}^{(n)} \log (5) O_{i,n}^{(n)}, \tag{38}
\]

to maximize the log-likelihood. Here, \( (5) O_{i,n}^{(n)} \) is the output for the input vector \( x^{(n)} \). The weight modification for \( W_{(3)}^{(c)} \) based on Newton’s method is defined as

\[
(3) W_{\text{new}} = (3) W_{\text{old}} - H^{-1} \nabla E, \tag{39}
\]

where \( (3) W_{\text{old}} \) and \( (3) W_{\text{new}} \) are the weight coefficients before and after weight modification, respectively, which have

| Table 2. Parameters for data generation in the simulation experiment. |
|-------------|-------------|-------------|-------------|-------------|
| \( \xi \)  | \( \lambda \) | \( \delta \) | \( \gamma \) | \( \Sigma^{-1} \) |
| Class 1    | 0.15        | 0.04        | 0.9         | -0.9        | 0.6         |
| Class 2    | 0.7         | 0.05        | 0.8         | 0.5         | 0.5         |
|            | 0.5         | 0.05        | 0.8         | 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.

(3) \( W_{(3)}^{(c)} \) in the \( c \)th block. \( \nabla E_n \) is the gradient vector whose \( i \)th element in the \( c \)th block can be calculated as:

\[
\frac{\partial E}{\partial (3) W_{h}^{(c)}} = \sum_{n=1}^{N} \frac{\partial E_n}{\partial (3) W_{h}^{(c)}} = \sum_{n=1}^{N} \sum_{i=1}^{C} \frac{\partial E_n}{\partial (5) O_{i,c}^{(n)}} \frac{\partial (5) O_{i,c}^{(n)}}{\partial (3) W_{h}^{(c)}} = \sum_{n=1}^{N} (5) O_{i,c}^{(n)} (3) W_{h}^{(c)} \tag{40}
\]

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

\[
\sum_{n=1}^{N} \frac{\partial^2 E_n}{\partial (3) W_{h}^{(c)} \partial (3) W_{l}^{(k)}} = \sum_{n=1}^{N} (5) O_{i,c}^{(n)} (3) W_{h}^{(c)} (3) W_{l}^{(k)} \tag{41}
\]

Note that \( H \) is positive semi-definite (see Appendix). It follows that \( E \) is a convex function of \( (3) W_{(3)}^{(c)} \) 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 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 ($P(c = 1|x)$) given by the proposed NN, LLR, and LLGMN. The probability of Class 2 ($P(c = 2|x)$) is clear from this graph because it can be calculated as $P(c = 2|x) = 1 - P(c = 1|x)$.

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 hyperparameters for ν-SVM (γ and ν) were optimized using 10-fold cross-validation (CV) and a 10 × 10 grid search (γ ranged from $\log_{10} 0.5$ to $\log_{10} 10^{-5}$, and ν ranged from $\nu_{max}$ to $\log_{10} 10^{-5}$ 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 $100 \times \frac{N_{correct}}{N_{total}}$, 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.

### 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 [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:

$$x_i^{(n)} = \frac{EMG_i(n) - EMG_i^{\text{m}}}{\sum_{i=1}^{d} (EMG_i(n) - EMG_i^{\text{m}})},$$

where $EMG_i^{\text{m}}$ 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. 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

### 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.

### FIGURE 7. Locations of electrodes for Dataset I.
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 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.

Fig. 9 shows the accuracy for each number of electrodes (Dataset IV) 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.

Fig. 9 shows the classification accuracy for each number of electrodes while decreasing the electrodes for Dataset IV. For comparison, the accuracies of \(\nu\)-SVM are also plotted. Significant differences between the proposed NN and \(\nu\)-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.

²http://ninapro.hevs.ch/node/131
TABLE 4. Results of EMG classification.

| Dataset | Algorithm | Accuracy [%] | P-measure | Kappa | CV time [s] | Training time [s] | Prediction time [s] |
|---------|-----------|--------------|-----------|--------|------------|-----------------|-------------------|
| I       | Proposed NN | 100          | 1         | 1      | 0          | 23.390 ± 0.548  | 0.170 ± 0.015     |
|         | SVM       | 100          |           | 1      | 1          | 197.407 ± 1.187 | 0.120 ± 0.001     |
|         | LLGMN     | 100          |           | 1      | 0          | 207.310 ± 1.759 | 0.120 ± 0.001     |
|         | MLP       | 100          |           | 1      | 0          | 99.504 ± 1.489  | 0.120 ± 0.001     |
|         | LLR       | 100          |           | 1      | 1          | 59.340 ± 0.584  | 0.120 ± 0.001     |
|         | k-NN      | 100          |           | 1      | 1          | 4.084 ± 0.079   | 0.120 ± 0.001     |
|         | Random Forest | 100        |           | 1      | 1          | 29.025 ± 4.891  | 6.536 ± 0.104     |

| II      | Proposed NN | 100          | 1         | 1      | 0          | 207.310 ± 1.759 | 0.120 ± 0.001     |
|         | SVM       | 100          |           | 1      | 0          | 99.504 ± 1.489  | 0.120 ± 0.001     |
|         | LLGMN     | 100          |           | 1      | 0          | 207.310 ± 1.759 | 0.120 ± 0.001     |
|         | MLP       | 100          |           | 1      | 0          | 99.504 ± 1.489  | 0.120 ± 0.001     |
|         | LLR       | 100          |           | 1      | 1          | 59.340 ± 0.584  | 0.120 ± 0.001     |
|         | k-NN      | 100          |           | 1      | 1          | 4.084 ± 0.079   | 0.120 ± 0.001     |
|         | Random Forest | 100        |           | 1      | 1          | 29.025 ± 4.891  | 6.536 ± 0.104     |

| III     | Proposed NN | 100          | 1         | 1      | 0          | 207.310 ± 1.759 | 0.120 ± 0.001     |
|         | SVM       | 100          |           | 1      | 0          | 99.504 ± 1.489  | 0.120 ± 0.001     |
|         | LLGMN     | 100          |           | 1      | 0          | 207.310 ± 1.759 | 0.120 ± 0.001     |
|         | MLP       | 100          |           | 1      | 0          | 99.504 ± 1.489  | 0.120 ± 0.001     |
|         | LLR       | 100          |           | 1      | 1          | 59.340 ± 0.584  | 0.120 ± 0.001     |
|         | k-NN      | 100          |           | 1      | 1          | 4.084 ± 0.079   | 0.120 ± 0.001     |
|         | Random Forest | 100        |           | 1      | 1          | 29.025 ± 4.891  | 6.536 ± 0.104     |

| IV      | Proposed NN | 100          | 1         | 1      | 0          | 207.310 ± 1.759 | 0.120 ± 0.001     |
|         | SVM       | 100          |           | 1      | 0          | 99.504 ± 1.489  | 0.120 ± 0.001     |
|         | LLGMN     | 100          |           | 1      | 0          | 207.310 ± 1.759 | 0.120 ± 0.001     |
|         | MLP       | 100          |           | 1      | 0          | 99.504 ± 1.489  | 0.120 ± 0.001     |
|         | LLR       | 100          |           | 1      | 1          | 59.340 ± 0.584  | 0.120 ± 0.001     |
|         | k-NN      | 100          |           | 1      | 1          | 4.084 ± 0.079   | 0.120 ± 0.001     |
|         | Random Forest | 100        |           | 1      | 1          | 29.025 ± 4.891  | 6.536 ± 0.104     |

| V       | Proposed NN | 100          | 1         | 1      | 0          | 207.310 ± 1.759 | 0.120 ± 0.001     |
|         | SVM       | 100          |           | 1      | 0          | 99.504 ± 1.489  | 0.120 ± 0.001     |
|         | LLGMN     | 100          |           | 1      | 0          | 207.310 ± 1.759 | 0.120 ± 0.001     |
|         | MLP       | 100          |           | 1      | 0          | 99.504 ± 1.489  | 0.120 ± 0.001     |
|         | LLR       | 100          |           | 1      | 1          | 59.340 ± 0.584  | 0.120 ± 0.001     |
|         | k-NN      | 100          |           | 1      | 1          | 4.084 ± 0.079   | 0.120 ± 0.001     |
|         | Random Forest | 100        |           | 1      | 1          | 29.025 ± 4.891  | 6.536 ± 0.104     |

| VI      | Proposed NN | 100          | 1         | 1      | 0          | 207.310 ± 1.759 | 0.120 ± 0.001     |
|         | SVM       | 100          |           | 1      | 0          | 99.504 ± 1.489  | 0.120 ± 0.001     |
|         | LLGMN     | 100          |           | 1      | 0          | 207.310 ± 1.759 | 0.120 ± 0.001     |
|         | MLP       | 100          |           | 1      | 0          | 99.504 ± 1.489  | 0.120 ± 0.001     |
|         | LLR       | 100          |           | 1      | 1          | 59.340 ± 0.584  | 0.120 ± 0.001     |
|         | k-NN      | 100          |           | 1      | 1          | 4.084 ± 0.079   | 0.120 ± 0.001     |
|         | Random Forest | 100        |           | 1      | 1          | 29.025 ± 4.891  | 6.536 ± 0.104     |

**: 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.

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 generalization 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 \( \nu \)-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 \( \nu \)-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, \( \nu \)-SVM and \( k \)-NN showed the longest times. Since \( \nu \)-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 in 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 \( \nu \)-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 such a 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**

**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

\[
\sum_{n=1}^{N} \frac{\partial^2 E_n}{\partial(3)w_h^{(c)} \partial(3)w_l^{(k)}} = \sum_{n=1}^{N} (5)c_{k}^{(n)} (5)O_{c, k}^{(n)}(4)c_{h}^{(n)}(4)O_{k, l}^{(n)}- (5)c_{k}^{(n)} (5)O_{c, k}^{(n)}(4)c_{h}^{(n)}(4)O_{k, l}^{(n)} + (5)c_{k}^{(n)} (5)O_{c, k}^{(n)}(4)c_{h}^{(n)}(4)O_{k, l}^{(n)}.
\]

For simplification, we 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 \( \mathbf{u} \in \mathbb{R}^{CH} \) with elements \( u_{c,h} \), where \( u_{c,h} \) denotes \( H(c-1)+h \)th element of \( \mathbf{u} \). Then,

\[
\mathbf{u}^T \mathbf{H} \mathbf{u} = \sum_{c,k} \sum_{h,l} u_{c,h}^{(5)} O_k(\delta_{c,k} - (5) O_k^{(4)} O_{c,h}^{(4)} O_{k,l} u_{k,l})
\]

\[
= \sum_{c,k} b_c^{(5)} O_k(\delta_{c,k} - (5) O_k b_k)
\]

\[
= \sum_{c,k} b_c b_k^{(5)} O_k - \sum_{c,k} b_c^{(5)} O_k b_k^{(5)} O_k
\]

\[
= \sum_{k} b_k^{(5)} O_k - \sum_{k} b_k^{(5)} O_k^2,
\]

\[
(44)
\]

where

\[
b_c = H \sum_{h} u_{c,h}^{(4)} O_{c,h},
\]

\[
b_k = l \sum_{l} u_{k,l}^{(4)} O_{k,l}.
\]

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

\[
\sum_{k} b_k^{(5)} O_k = \sum_{k} f(b_k) O_k^{(5)} O_k
\]

\[
\geq f(\sum_{k} b_k O_k) = \sum_{k} b_k^{(5)} O_k^2.
\]

\[
(47)
\]

Therefore,

\[
\mathbf{u}^T \mathbf{H} \mathbf{u} \geq 0.
\]

\[
(48)
\]

From the definition of definiteness, the Hessian matrix \( \mathbf{H} \) is positive semi-definite.

REFERENCES

[1] N. Mamonne, 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., vol. 29, no. 10, pp. 5122–5135, Oct. 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., vol. 29, no. 11, pp. 5619–5629, Nov. 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, Nov. 2016.

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

[5] Y. Geng, Y. Quyang, 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. 38326–38335, 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 Annus. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC), Jul. 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 Signal Process. (ICASSP), May 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. 13121–13134, 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., early access, Aug. 10, 2020, doi: 10.1109/TCYB.2020.3007173.

[10] T. 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,” J. Ambient Intell. Hum. Comput., vol. 11, no. 12, pp. 6021–6031, Dec. 2020.

[11] A. Subasi and S. M. Qaisar, “Surface EMG signal classification using TQTW, bagging and boosting for hand movement recognition,” J. Ambient Intell. Hum. Comput., vol. 2020, pp. 1–16, Apr. 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,” Biomed. Signal Process. Control, vol. 58, Apr. 2020, Art. no. 101872.

[13] W. Tigrer, B. Navarro, A. Cherubini, X. Gorron, A. Gelis, C. Fattal, D. Guiraud, and C. A. Coste, “A novel EMG interface for individuals with tetraplegia to pilot robot hand grasping,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 26, no. 2, pp. 291–298, Feb. 2018.

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

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

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

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” Nature, vol. 323, pp. 533–536, Oct. 1986.

[18] T. Cover and P. Hart, “Nearest neighbor pattern classification,” IEEE Trans. Inf. Theory, vol. IT-13, no. 1, pp. 21–27, Jan. 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. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2730–2739.

[20] H. Hayashi, T. Shibanozki, 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, Feb. 2015.

[21] T. Tsuji, N. Bu, O. Fukuda, and M. Kaneko, “A recurrent log-linearized Gaussian mixture network,” IEEE Trans. Neural Netw. Learn. Syst., vol. 14, no. 2, pp. 304–316, Mar. 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, Feb. 1999.

[23] A. Furui, T. Iguae, and T. Tsuji, “EMG pattern recognition via Bayesian inference with scale mixture-based stochastic generative models,” Expert Syst. Appl., vol. 185, Dec. 2021, Art. no. 115644.

[24] H. Johnson, “Systems of frequency curves generated by methods of transcription,” Biometrika, vol. 36, nos. 1–2, pp. 149–176, 1949.

[25] P. M. Stanfield, J. R. Wilson, G. A. Mirka, N. F. Glasscock, J. P. Psichogios, and J. R. Davis, “Multivariate input modeling with Johnson distributions,” in Proc. 28th Conf. Winter Simul., Dec. 1996, pp. 1457–1464.

[26] N. L. Johnson, “Bivariate distributions based on simple translation systems,” Biometrika, vol. 36, nos. 3–4, pp. 297–304, 1949.

[27] N. L. Johnson and S. Kotz, Distributions in Statistics. Continuous Multivariate Distributions. Hoboken, NJ, USA: 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 5th Int. Workshop Comput. Intell. Appl. (IWCA), Nov. 2015, pp. 115–120.

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

[30] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, 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. BME-27, no. 7, pp. 382–395, Jul. 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, Oct. 2017.

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

[34] A. Altmu and O. Er, “Comparison of different time and frequency domain feature extraction methods on elbow gesture’s EMG,” Eur. J. Interdiscipl. Stud., vol. 2, no. 3, p. 35, Aug. 2016.

[35] E. J. Rechy-Ramirez and H. Hu, “Stages for developing control systems using EMG and EEG signals: A survey,” School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., Tech. Rep. CES-513, 2011.

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

[37] J. F. Slifker and S. S. Shapiro, “The Johnson system: Selection and parameter estimation,” Technometrics, vol. 22, no. 2, pp. 239–246, May 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 r-support vector classifiers: Theory and algorithms,” Neural Comput., vol. 13, no. 9, pp. 2119–2147, 2001.

[40] D. E. King, “Dlib-ml: A machine learning toolkit,” J. Mach. Learn. Res., vol. 10, pp. 1755–1758, Jan. 2009.

[41] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” Inf. Process. Manag., 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 Programs Biomed., 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. 12nd Int. Conf. Control Automat. Robot. Vis. (ICARCV), Dec. 2012, pp. 1534–1539.

[44] T. Shibano, 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, Mar. 2013.

[45] M. Atzori, A. Gijsberts, B. 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,” Sci. Data, vol. 1, no. 1, pp. 1–13, Dec. 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. Biol. Soc. (EMBC), Jul. 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. Soc. Nat. Comput. Conf. (NCC), Apr. 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. Adv. Sci. Eng. Technol. Int. Conf. (ASET), Feb. 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.