Reliability Analysis for Finger Movement Recognition With Raw Electromyographic Signal by Evidential Convolutional Networks

Yuzhou Lin, Ramaswamy Palaniappan, *Senior Member, IEEE*, Philippe De Wilde, *Senior Member, IEEE*, and Ling Li

***Abstract*— Hand gesture recognition with surface electromyography (sEMG) is indispensable for Muscle-Gesture-Computer Interface. The usual focus of it is upon performance evaluation involving the accuracy and robustness of hand gesture recognition. However, addressing the reliability of such classifiers has been absent, to our best knowledge. This may be due to the lack of consensus on the definition of model reliability in this field. An uncertainty-aware model has the potential to self-evaluate the quality of its inference, thereby making it more reliable. Moreover, uncertainty-based rejection has been shown to improve the performance of sEMG-based hand gesture recognition. Therefore, we first define model reliability here as the quality of its uncertainty estimation and propose an offline framework to quantify it. To promote reliability analysis, we propose a novel end-to-end uncertainty-aware finger movement classifier, i.e., evidential convolutional neural network (ECNN), and illustrate the advantages of its multi-dimensional uncertainties such as vacuity and dissonance. Extensive comparisons of accuracy and reliability are conducted on NinaPro Database 5, exercise A, across CNN and three variants of ECNN based on different training strategies. The results of classifying 12 finger movements over 10 subjects show that the best mean accuracy achieved by ECNN is 76.34%, which is slightly higher than the state-of-the-art performance. Furthermore, ECNN variants are more reliable than CNN in general, where the highest improvement of reliability of 19.33% is observed. This work demonstrates the potential of ECNN and recommends using the proposed reliability analysis as a supplementary measure for studying sEMG-based hand gesture recognition.**

***Index Terms*— Convolutional neural network, evidential deep learning, hand gesture recognition, model reliability, surface electromyography (sEMG), uncertainty-awareness.**

I. Introduction

SURFACE electromyography (sEMG) refers to the collective electrical signals from muscles that are collected by noninvasive electrodes. The sEMG-based hand gesture recognition is a practical application of sEMG that has found wide usage in advanced prostheses control [1], [2] and other rehabilitation applications [3]. It is crucial that the development of such a classification-based control scheme highly relies on the accurate and robust hand gesture predictions of users. As a result, the current research on sEMG-based hand gesture recognition has focused on improving its accuracy [4]–[6] and robustness [5], [7]–[9] with recent deep learning techniques. Note that model robustness can be summarised as the ability to remain accurate in practical scenarios under many factors that may affect the prediction performance, such as electrode shifts, sweating, limb posture and force changes, and day-to-day variation [7], [10]–[15]. A special case of robustness is to tackle subject variability when considering the user-independent sEMG-based hand gesture recognition [5], [9].

Recently, the rejection of hand movements based on uncertainty measures has shown good potential as a general practical solution for improving the usability of sEMG-based myoelectric control by boosting both the accuracy and robustness of hand gesture recognition [16]–[18]. Ideally, most of the inaccurate ambiguous predictions could be rejected by introducing additional information, such as entropy or the normalized maximum probability of the predictive distribution, for the indication of confidence level. The intuition behind this is to address the concern where the gesture recognition process is being considered as a 'black box' for myoelectric control [16]. In this paper, we first defined the *reliability* R of an sEMG-based hand gesture classifier as the quality of its uncertainty measures that produce confidence scores on the predictions of test samples. Its reliability analysis then refers to the evaluation of R. This is supported by the commonly held opinion that accurate and robust hand gesture recognition is considered reliable [7], [8], and the statement that accurate uncertainty estimation is one of the essential factors for the reliable application of deep learning [19].

Although deep learning models, particularly those based on convolutional neural networks (CNNs), have achieved state-of-the-art (SoA) performance regarding both accuracy and robustness to sEMG-based hand gesture recognition, the reliability analysis of CNNs in this field has remained unexplored, which has become an increased necessity due to the vulnerability of

Manuscript received June 4, 2021; revised September 17, 2021, November 8, 2021, and December 16, 2021; accepted December 28, 2021. Date of publication January 7, 2022; date of current version January 28, 2022. This work was supported by the Vice Chancellor's Research Scholarship for International Students at the University of Kent. *(Corresponding author: Yuzhou Lin.)*

Yuzhou Lin, Ramaswamy Palaniappan, and Ling Li are with the School of Computing, University of Kent, Canterbury, Kent CT2 7NZ, U.K. (e-mail: yl339@kent.ac.uk; r.palani@kent.ac.uk; c.li@kent.ac.uk).

Philippe De Wilde is with the Division of Natural Sciences, University of Kent, Canterbury, Kent CT2 7NZ, U.K. (e-mail: p.dewilde@kent.ac.uk).

Digital Object Identifier 10.1109/TNSRE.2022.3141593

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

deep learning models reported recently [20]–[22]. The reliability analysis has direct benefits to current studies, which include latent concerns about model reliability in rejection-based hand gesture recognition. For example, Wu *et al*. [18] recently proposed a metric-learning guided CNN to enhance the robustness of myoelectric control systems by effectively rejecting novel patterns, i.e., new classes were not included in the training. It is evident that there is a positive correlation between the defined reliability R and the performance of rejection-capable sEMG-based hand gesture recognition. This implies that quantifying R could provide a useful indication of model performance without suffering from the limitations of evaluating its rejection-capable recognition performance, such as introducing extra evaluation measures (e.g., accuracy-rejection curve [23], false activation error [24]) and highly relying on determining the optimal rejection threshold [17].

Additionally, current uncertainty measures used in sEMG-based hand gesture recognition fail to provide meaningful insight into predictions. Recent studies in the field of predictive uncertainty estimation have shown that evidential neural networks [25], [26] modeled with Dirichlet-based uncertainty [19] are more efficient in explicitly measuring uncertainties such as vacuity and dissonance [27] with almost no extra computational cost, unlike other approaches such as Bayesian neural networks [28] or ensemble models [29]. The potential of applying evidential deep learning to the sEMG-based hand gesture recognition will be further explored in this paper.

This study aims to propose a framework to directly quantify R, with a specific focus on the reliability analysis of individuated finger movement recognition with raw sEMG. Such movements are highly complex and versatile [30], which naturally raises the real necessity of reliability analysis. We first employ an existing end-to-end CNN model [5] and propose an uncertainty-aware model, i.e., the evidential CNN (ECNN) by integrating it with evidential deep learning. As a pilot study towards the reliability analysis of sEMG-based finger movement classifiers, the discussion starts with an illustration of how the generated multidimensional uncertainties such as vacuity and dissonance of ECNN could be precisely quantified and leveraged for a 'difficult to classify' finger movement recognition compared with CNN. Furthermore, a brief comparison of the performance of rejection-capable finger movement recognition between them is provided as empirical evidence to support the intuition behind this research. Finally, and most importantly, we first recommend using a threshold-free evaluation metric called normalised Area Under Precision-Recall (nAUPRC) [31] to evaluate the misclassification detection, which is introduced to quantify R, to avoid the pitfall that current related evaluation metrics such as the Area Under Receiver Operating Characteristic (AUROC) [32] and Precision-Recall (AUPRC) [33] can only be used to assess the misclassification detection performance of a single model rather than directly compare across different models [34]. To further reduce the bias of results and ensure fair comparison, extensive empirical evaluations are provided by employing the stratified nested cross-validation with the Tree-Structured Parzen Estimator, which is one of the SoA hyperparameter optimisation algorithms.

II. Problem Statement

Reliability analysis for finger movement recognition relies on a framework that can explicitly measure the model reliability R, i.e., the quality of its uncertainty estimates. The challenges are manifold: it must be quantifiable and ideally located in a fixed interval $[0, 1]$; it must be consistent for any classifier and uncertainty measure; the results must be comparable in a fair way regardless of the model accuracy. Inspired by studies on evaluating uncertainty quantification, the reliability of the sEMG-based finger movement recognition could be evaluated by measuring the performance of the misclassification detection, which aims to detect wrong predictions with quantified uncertainty estimates as scores. An ideal reliable classifier enables the assignment of higher uncertainty measures when incorrect predictions are being made compared to correct predictions. In other words, the reliability assesses the discrimination level of uncertainty quantification assigned to wrong and correct predictions.

The misclassification detection can be considered as a binary classification problem where wrong predictions are positive samples and correct predictions refer to negative samples. The quantified uncertainty is taken as the score and any samples with scores higher than a threshold will be assigned to positive samples, and negative ones otherwise. To avoid providing arbitrary results with a user-defined score threshold, the AUROC and AUPRC are commonly used as threshold-free evaluation summary metrics, which can overcome most challenges addressed above. However, these are incomparable since each model has its own accuracy on each test set, which yields different positive and negative samples regarding misclassification detection. More details of our proposed framework with a solution to address this challenge are presented in Sec. V.

III. Evidential Convolutional Neural Network

In *Dempster-Shafer Theory of Evidence* [35] (DST), a *frame of discernment* Θ is defined as a finite set of mutually exclusive elements in a domain, where a subset of Θ is referred to as a hypothesis or proposition and a *singleton* is used to represent it if the cardinality of this subset equals to 1. The belief of a proposition could be quantified by *belief functions* based on the available evidence, which allows us to not follow the additivity principle of probability theory strictly, thus providing an additional "dimension of uncertainty" to make ignorance explicit [36]. Based upon the DST's notion of belief assignment over Θ, *Subjective Logic* (SL) [37] provides a structured approach to connect beliefs to Dirichlet distributions so that we can approximate second-order Bayesian reasoning in a computationally efficient way. The second-order uncertainty of a multiclass classifier is represented by a Dirichlet probability density function (PDF) over a multinomial distribution, which refers to the first-order uncertainty representing the predicted class probabilities. It enriches the uncertainty representation with extra information from beliefs. Let $Y = (Y_1, Y_2, \ldots, Y_K)$

be a discrete variable in a domain $\mathbb{Y}$, and represents the class label. For a multiclass classification problem, the number of class $K = |\mathbb{Y}| > 2$. A multinomial opinion over Y in SL is then defined as an ordered triplet $w_Y = (\boldsymbol{b}_Y, u_Y, \boldsymbol{a}_Y)$ where

- $\boldsymbol{b}_Y$ refers to a *belief mass distribution* over $\mathbb{Y}$;
- u_Y is the *uncertainty mass* expressing the vacuity of evidence, which decreases as more observations in terms of statistical events are found;
- $\boldsymbol{a}_Y$ represents a *base rate distribution* over $\mathbb{Y}$, which is known as *prior probability* in classic Bayesian theory.

The projected probability distribution of a multinomial opinion in SL is defined as follows [37]:

$$\boldsymbol{P}_Y = \boldsymbol{b}_Y + \boldsymbol{a}_Y u_Y. \tag{1}$$

SL demonstrates clearly that there is a specific bijective mapping between a multinomial opinion and a Dirichlet PDF over the same domain $\mathbb{Y}$. Before proceeding further, let us recall the definition of a Dirichlet PDF over the same discrete variable Y on domain $\mathbb{Y}$ [38]:

$$Dir(\boldsymbol{p}_Y) = \frac{\Gamma\left(\sum_{j=1}^{K} \alpha_j\right)}{\prod_{j=1}^{K} \Gamma(\alpha_j)} \prod_{j=1}^{K} p_{Y_j}^{(\alpha_j - 1)}, \tag{2}$$

where $\boldsymbol{p}_Y$ represents the probability distribution for discrete variable Y, such that each $p_{Y_j} \in (0, 1)$ and $\sum_{j=1}^{K} p_{Y_j} = 1$; $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ is a strength vector of positive-valued Dirichlet parameters; $\Gamma(\cdot)$ is the standard Gamma function. Since the Dirichlet distribution belongs to the exponential family, its conjugation property allows us to consider the Dirichlet parameter $\boldsymbol{\alpha}$ as the prior and observation evidence. From the perspective of SL, each singleton can have an arbitrary additive base rate distribution $\boldsymbol{a}_Y$ over the domain $\mathbb{Y}$ rather than default value $1/K$ and $\boldsymbol{\alpha}$ can be redefined as [37]:

$$\boldsymbol{\alpha} = \boldsymbol{r} + \boldsymbol{a}_Y W, \tag{3}$$

where $\boldsymbol{r}(\geq 0)$ is a vector of evidence over variable Y and W is a constant expressing the non-informative prior weight. The evidence representation of the Dirichlet PDF can then be obtained by substituting the above equation into (2) and the expected probability distribution over $\mathbb{Y}$ is [37]:

$$\mathbb{E}_Y = \frac{\boldsymbol{\alpha}}{\sum \boldsymbol{\alpha}} = \frac{\boldsymbol{r}}{W + \sum \boldsymbol{r}} + \boldsymbol{a}_Y \frac{W}{W + \sum \boldsymbol{r}}. \tag{4}$$

Intuitively, to build such a bijective mapping, the projected probability distribution defined in (1) is supposed to equal the expected probability distribution defined in (4). More specifically, the observed evidence in the Dirichlet PDF could be simply mapped to the belief mass distribution $\boldsymbol{b}_Y$ (i.e. $= \frac{\boldsymbol{r}}{W + \sum \boldsymbol{r}}$) and uncertainty mass u_Y (i.e. $= \frac{W}{W + \sum \boldsymbol{r}}$). Note that the total belief mass $\sum \boldsymbol{b}_Y$ approaches to 1 (or 0) while the u_Y reaches 0 (or 1), as the total evidence goes to infinity (or 0). These properties match the additivity requirement of a multinomial opinion over Y, i.e., $\sum \boldsymbol{b}_Y + u_Y = 1$.

Based on the framework of SL, evidential deep learning (EDL) was proposed to help explicitly train an uncertainty-aware model [25]. In EDL, the term *evidence* $\boldsymbol{e}$ has been defined as a measure of the amount of support collected from extracted features in favour of an input sample to be classified into a certain class. Recall that a discrete variable $Y = (Y_1, \ldots, Y_K)$ represents the class label for a K-classification problem. The non-informative prior weight W equals to K since a uniform prior PDF is required when there is no observation. Naturally, each element of the base rate vector $\boldsymbol{a}_Y$ equals to $1/K$ without any extra information. Therefore, one can compute the belief mass vector $\boldsymbol{b}$ by $\boldsymbol{e}/(K + \sum \boldsymbol{e})$. It is noted that the denominator is referred to as the *total evidence* S, which could be re-written as $\sum(\boldsymbol{e} + 1)$ because the number of elements in $\boldsymbol{e}$ is K. Furthermore, the Dirichlet distribution with parameter vector $\boldsymbol{\alpha}$ could be mapped to the evidence vector $\boldsymbol{e}$ by $\boldsymbol{\alpha} = \boldsymbol{e} + 1$.

In this paper, we propose an Evidential Convolutional Neural Network (ECNN) which is designed by integrating an existing end-to-end convolutional neural network [5] with EDL (the details are presented in Sec. V-B). Unlike using the *softmax* to obtain class probabilities directly, ECNN replaces it with an activation layer such as *ReLU* to output a nonnegative evidence vector for the predicted Dirichlet distribution of finger movement. With the aid of the loss function presented in (5), this allows ECNN to learn to collect the evidence leading to a subjective opinion used for predicting finger movement with the support of explicit uncertainty estimates. Note that other possible activation functions will be investigated later in this paper as part of the process of hyperparameter optimisation.

Given a sample i and let $\mathbf{y}_i$ be a one-hot encoding of the ground-truth class of it with $y_{ij} = 1$ and $y_{im} = 0$ for all $j \neq m$ where j and m are class labels. The predicted probability of sample i for j^{th} finger movement p_j in ECNN is computed as α_{ij}/S_i based on (4). Moreover, the sum-of-squares loss function can be used to train ECNN with the joint goal of minimising the prediction error and the variance of the Dirichlet distribution [25], presented as:

$$\mathcal{L}_1(f(\mathbf{x}_i|\Theta), \mathbf{y}_i) = \sum_{j=1}^{K} (y_{ij}^2 - 2y_{ij}\mathbb{E}[p_{ij}] + \mathbb{E}[p_{ij}^2]), \tag{5}$$

where $f(\cdot)$ is the evidence vector predicted given the observed feature $\mathbf{x}_i$ from sample i by the classifier with parameters Θ.

The *vacuity* (u_{vac}) and *dissonance* (u_{diss}), which are referred to as the *evidential uncertainty* of ECNN. Vacuity denotes uncertainty due to lacking evidence or knowledge, i.e., u_Y, which can be either calculated as K/S or $1 - \sum \boldsymbol{b}$. Dissonance represents the uncertainty due to conflicting evidence, derived from a sufficient number of conflicting evidence by comparing each two singleton belief masses [26]:

$$u_{diss} = \sum_{j=1}^{K} \left(\frac{b_j \sum_{m=1, m\neq j}^{K} \mathbf{Bal}(b_j, b_m)}{\sum_{m=1, m\neq j}^{K} b_m} \right), \tag{6}$$

where $\mathbf{Bal}(b_j, b_m)$ represents the relative mass balance between a pair of belief masses b_j and b_m for the sample i, equals to 0 when $b_j + b_m = 0$, and $1 - \frac{|b_j - b_m|}{b_j + b_m}$ otherwise.

We also introduce two uncertainty measures [16] which can be used for all models: entropy and negative maximum probability. The entropy is simply defined as $H = -\sum p(j) \ln p(j)$

and $p(j)$ is the predicted probability for class j. Since the maximum probability across classes can be interpreted as the confidence level, it could then be used as an uncertainty score by taking its negative value. However, the range of entropy and negative maximum probability is $[0, \ln(1/K)]$, and $[-1, 0]$ respectively. For consistency, they will be normalised to a range from 0 to 1 and noted as $u_{nEntropy}$ and u_{nnmp}.

IV. Illustration

This section aims to briefly illustrate the power of ECNN with its meaningful evidential uncertainty in classifying finger movements with raw sEMG. This was done by comparing apples to apples, i.e., ECNN and its conventional version (CNN). All details of the models and data used here can be found in Sec. V. Briefly, models were trained and tested only for the first subject from NinaPro Database 5 to classify 12 finger movements with 16-channels raw sEMG signals, which was segmented using a 250 ms window with a 90% overlap. Therein, models were trained by the 1st, 3rd, 4th and 6th cycles, whereas the 2nd cycle was used as validation set for early stopping and the 5th cycle was used to test the performance. For ease of comparison, we set the batch size, learning rate, and optimization method to 256, 0.002, and ADAM [39] during the training. Moreover, the cross-entropy loss was used for training the CNN, whereas the sum-of-squared loss as shown in (5) was used for training the ECNN.

We first illustrate the power of the evidential uncertainty of ECNN by taking an example of classifying 'thumb adduction', which is easily confused during classification as 'thumb flexion' due to the similarity of movements. The top and bottom panels of Fig. 1 show that CNN starts making wrong predictions during transient movements. This is consistent with the finding that the offline transient-state sEMG-based hand gesture recognition accuracy is usually less than the steady-state one as the transient-state sEMG has more variance than the steady-state one over time [40], [41]. The evidential uncertainty of ECNN reveals this clearly by presenting either high u_{vac} or u_{diss} during the transient phase, seen in the middle panel of Fig. 1. More importantly, it shows a clear understanding of the uncertainty sources in this example. What CNN attempts to show is that the uncertainty at the beginning comes from conflicting evidence since its predicted probabilities for the 12th finger movement 'thumb flexion' are high at this stage. This is exactly what ECNN has revealed by giving high values of u_{diss}. Similarly, CNN shows ignorance at the end since it assigns high predicted probabilities for 'middle flexion', which seems unrelated to the ground truth 'thumb adduction'. Again, this has been disclosed by ECNN via presenting high values of u_{vac}. Fig. 1 also shows that ECNN does not make overconfident predictions compared to CNN, especially when predictions may go wrong. Note that for ease of viewing, the focus is only on those classes with likely incorrect predictions, the sequential predictions of a wrong class are presented in Fig. 1 only if one of them has been assigned over 0.5.

In summary, Fig. 1 illustrates that ECNN has the potential to precisely quantify predictive uncertainties with an understanding of the uncertainty sources. A natural question that arises is:

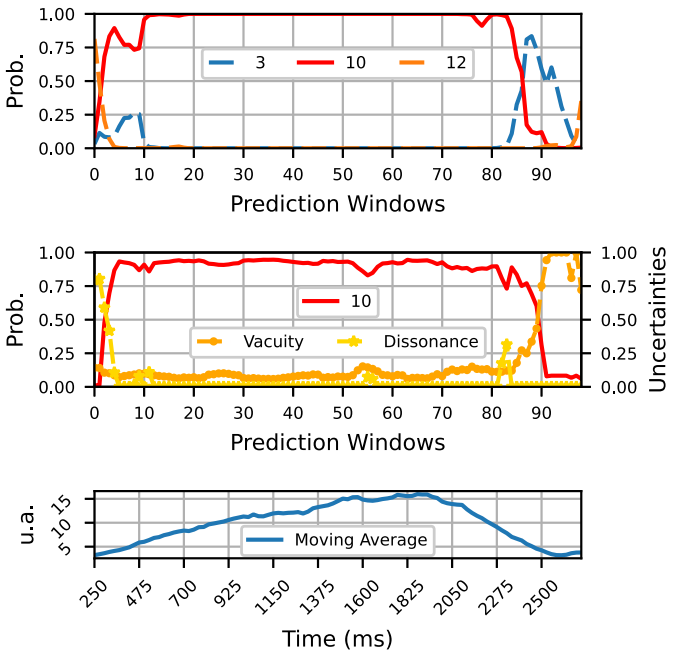

Fig. 1. Sequential predictions of the 'thumb adduction' (class 10) on offline testing. Note that the sequential predictions of a wrong class are presented only if one of them has been assigned over 0.5. **Top**: The predicted probabilities of the CNN; Class 3 and 12 refer to 'middle flexion' and 'thumb flexion'. **Middle**: The predicted probabilities of the ECNN with its evidential uncertainty. **Bottom**: The sum of moving averages of 16-channel raw rectified sEMG signals with absolute values regarding the dynamic finger movement of 'thumb adduction'. The u.a. means 'unitless' activation since sEMG recorded by Thalmic Myo armbands is claimed to be 'unitless' with an unknown conversion from mV.

how could we better leverage this for improving sEMG-based hand gesture recognition performance? One straightforward solution is to allow a classifier to reject making a prediction when whichever dimension of uncertainty is considered as high. Assuming that the high uncertainties are only generated when wrong predictions are being made, making rejections under such conditions is then definitely a benefit to boost the hand gesture recognition accuracy and make the accepted predictions more reliable. This is the intuition behind the rejection-capable sEMG-based finger movement recognition. To briefly compare the classification performance of CNN and ECNN when allowing a model to reject making predictions by leveraging the uncertainty estimate, we first calculated $u_{nEntropy}$ for CNN and $max(u_{vac}, u_{diss})$ for ECNN regarding uncertainty estimates. By setting a confidence threshold δ, where its range is set to be $[0, 0.5]$, for discrimination between certain and uncertain predictions, the model is allowed to not make a prediction whenever its quantified uncertainty is larger than $(1 - \delta)$. When $\delta = 0$, it simply refers to the standard recognition where no rejections will be made. The upper limit of δ was set to be 0.5 since a value of more than 0.5 is perceived as too strict, which might lead to a situation where no predictions are made. Inspired by studies of rejection-capable sEMG-based hand gesture recognition, the three evaluation metrics used here are defined as follows: *Rejection Rate* (RR) is the percentage of predictions that are rejected [16], [23]; *True Acceptance/Rejection Rate* (TAR/TRR) refers to the rate

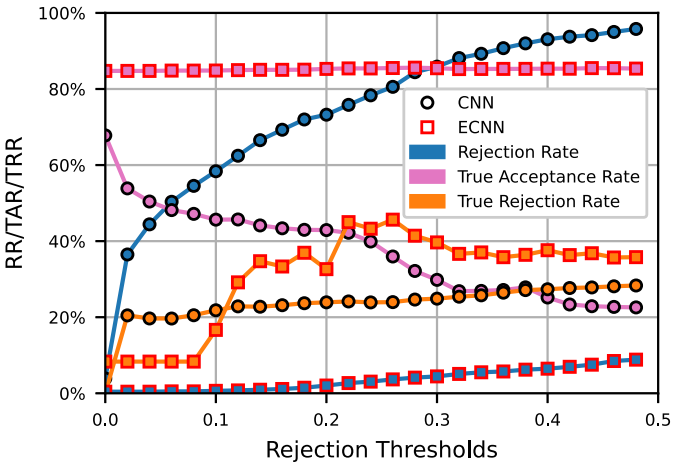

Fig. 2. Three comparison results of classifying 12 finger movements between CNN and ECNN with a condition that rejects making predictions when the quantified uncertainties are over a predefined threshold.

at which a classifier correctly makes active/inactive predictions. Note that the false acceptance/rejection rate (FAR/FRR) was defined in [16] and TAR/TRR $= 1-$ FAR/FRR.

Fig. 2 shows how ECNN outperforms CNN on rejection-capable sEMG-based finger movement recognition in this example. Firstly, even though more predictions will be rejected as the confidence threshold δ increases, the lines in blue show that the gradient of RR for ECNN is much smaller than CNN. When the threshold reaches 0.5, CNN almost stops making any predictions, but the RR of ECNN remains at about 10% only. This gives additional backing to the proposed statement that CNN is being overconfident. Secondly, the TAR of ECNN remains high constantly, whereas it drops for CNN as the δ goes up. Recall that the TAR can be considered as finger movement recognition accuracy but under the condition of allowing the model to not make an unsure prediction. The standard recognition accuracy of ECNN is also higher than CNN, as shown in pink points when $\delta = 0$. Finally, it shows that ECNN is making more valid rejections generally than CNN, supported by the TRR shown in orange. One may observe that ECNN has a lower TRR when the δ varies from 0 to 0.1, which may be caused by the extremely low RR of ECNN, i.e., very few predictions are rejected when the δ is small. Although ECNN has shown its superiority in this example, we have to claim that one example can not prove ECNN is more reliable than CNN. Therefore, the illustration here can only be considered as supplementary for readers to better understand the special properties of ECNN with evidential uncertainty. This small example also indicates how to investigate the rejection-capable sEMG-based finger movement recognition performance with uncertainty measures conventionally. The proposed proper reliability analysis for both models will be explained in detail later.

V. Experiments

A. Database

Our evaluations were carried out on the NinaPro Database 5 (*NinaPro DB5*), which was recorded with a double Myo setup in one session consisting of 6 repetitions of 52 hand movements (plus rest), which were divided into exercise sets A (finger movements), B (hand and wrist movements), and C (other functional movements), performed by 10 healthy subjects [2]. It is noted that each repetition of all complete movements is sometimes referred to as a *trial* [6] or a *cycle* [5]. Here the term 'cycle' is employed to avoid confusion from the term 'trial' used in the hyperparameter optimisation process. Since we are particularly interested in sEMG-based finger movement recognition, only exercise A is used, which covers 12 finger movements involving both flexion and extension of five fingers plus thumb adduction and abduction. To meet the real-time demands of controlling devices such as prostheses, i.e., the 300 ms constraint [42], the raw sEMG data was segmented by applying a sliding window of 250 ms with a non-overlap length of 25 ms. Such high overlap was used for data augmentation [5]. Hence, each frame has a dimension of 16 electrode channels $\times$ 50 sEMG sample points since the sampling frequency of *NinaPro DB5* is 200 Hz. Note that no extra signal preprocessing was required.

B. Models

To reduce any bias, in our work, the *enhanced raw ConvNet* architecture, which was first proposed by [5], was employed here to evaluate finger movement recognition performance in terms of both accuracy and reliability as a baseline method. It was modified to adapt for this task, which is to classify 12 finger movements by taking a frame of raw sEMG signals with a dimension of 16×50. In essence, the CNN architecture is composed of two convolutional layers and two fully connected layers which have 2304 and 500 hidden units, respectively. The 3×5 kernels with a stride of 1 and no zero padding were used on the convolutional layers. Furthermore, recent techniques such as Batch Normalisation (BN) [43], Parametric Rectified Linear Unit (PReLU) activation function [44], and dropout were applied to each layer. For a fair comparison, ECNN has the same network architecture as CNN except in the way of interpreting the model outputs and the loss functions used for training the network. More details are shown in Fig. 3.

C. Experimental Setup

All experiments were implemented in PyTorch v.1.1.0 and Python 3.7.3. The experimental sequences were constructed by data loading, data segmentation, model training, and model testing. A standard cross-validation (CV) procedure may cause biased results when assessing classification models [45], [46]. To reduce the bias and to better compare the finger movement recognition performance between CNN and ECNN, a stratified nested CV procedure [46], [47] was employed in this work, where an inner CV loop was used to determine the best hyperparameters for the training of a model, whereas an outer CV was then applied to test and compare the results. Stratification allows each fold divided from the data to have similar proportions of samples with the same label. This could be done by simply splitting the data via the repetition number here. Since each subject performed 6 repetitions of all gestures in the *NinaPro DB5*, the splitting ratio of training, validation, and testing datasets was set to $4:1:1$ regarding cycle number

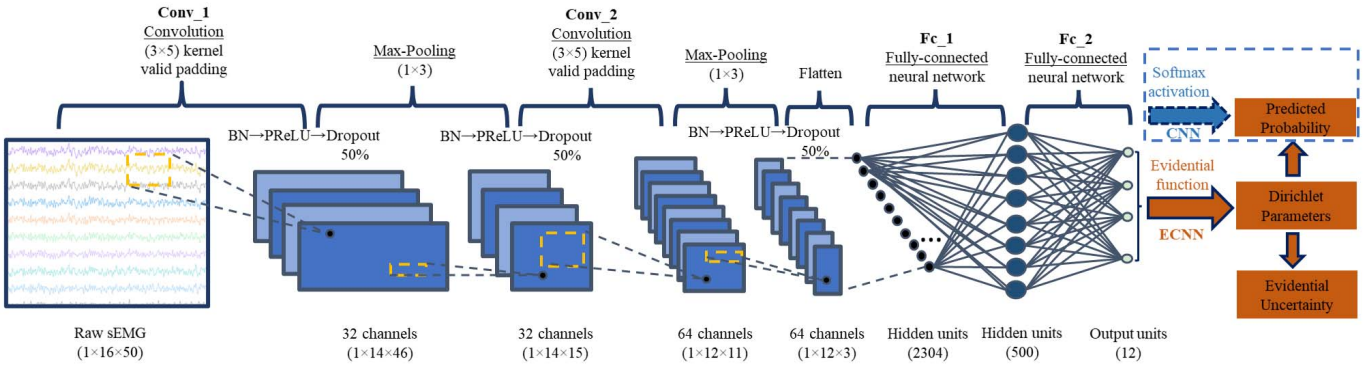

Fig. 3. The detailed illustrations of the proposed Evidential Convolutional Networks (ECNN) and its conventional version (i.e., CNN).

to maximise the data used for training. Such data splitting could also avoid data leakage between training and testing. Recall that the raw sEMG signal was segmented by a sliding window and the overlap between every two consecutive frames was as high as 90%. Hence, randomly splitting the sample set may cause such a leakage scenario where a sample falls into the training set while its adjacent segments could be found in the testing set. Furthermore, early stopping was employed to avoid overfitting by setting the patience term to 10. The training would then be stopped when no improvement was found in the validation set after waiting for 10 epochs or the training epoch up to 1000.

Unlike conventional hyperparameter optimisation (HPO) algorithms such as Grid or Random Search, we applied one of the SoA HPO algorithms, the Tree-structured Parzen Estimator (TPE) [48], [49], to reduce the computation burden. Being an approach based on sequential model-based global optimization algorithms [48], [50], the TPE organises hyperparameters into a tree-like space so that the available values of a specific hyperparameter will be determined based on the previous search results. With the aid of Optuna [51], which is a powerful hyperparameter optimisation framework, the unpromising *trials* will be terminated at an early stage where each trial refers to each evaluation of an objective function. Such a strategy is also referred to as *pruning*, and the 'MedianPruner' constructed by the Median Stopping Rule [52] was used here. Specifically, the objective value is then the mean of the validation losses collected from the inner CV loops. Moreover, the number of study trials was set to 25 and the pruning was enabled after 5 trials were completed in each process of HPO. The source code for this study is available on GitHub (https://github.com/YuzhouLin/ECNN-RAnal), and the determined optimal hyperparameters of each model on each test trial of CV for each individual can be found here as well.

The hyperparameter search space is listed in Table I. The common hyperparameters used for training both CNN and ECNN include batch size, learning rate, and optimizer method. To better explore the potential of ECNN, we investigated different functions to generate the evidence vector (called 'evidence fun' in Table I) and train the model. Instead of employing ReLU as the last activation function for ECNN

TABLE I
HYPERPARAMETER SEARCH SPACE

Hyperparameters	CNN	ECNN-A	ECNN-B	ECNN-C
batch size	{128, 256}			
learning rate	[1e-3, 1e-2]			
optimizer	{"ADAM", "RMSprop", "SGD"}			
evidence fun	×	{"*ReLU*", "*SoftPlus*", "*Exp*"}		
annealing step	×	×	[10, 60]	×
tau	×	×	×	[0.1, 1.0]

to turn the model outputs into the nonnegative evidence vector for the predicted Dirichlet distribution, other functions such as *SoftPlus* and the exponential function (*Exp*) can be investigated. Note that any value larger than 3 would be limited to 3 when using the exponential function for training convergence. More importantly, ECNN can be trained by incorporating a Kullback-Leibler (KL) divergence term into the sum-of-squares loss function [25], as shown in (7):

$$\mathcal{L}(\Theta) = \mathbb{E}_{(\mathbf{x}_i,\mathbf{y}_i)\sim\mathcal{D}}\Big[\mathcal{L}_1\left(f\left(\mathbf{x}_i \mid \Theta, \mathbf{y}_i\right)\right. \\ \left. + \lambda KL\left[\mathrm{Dir}\left(\mathbf{p}_{-k}; \boldsymbol{\alpha}_{-k}\right) \| \mathrm{Dir}\left(\mathbf{p}_{-k}; \mathbf{1}\right)\right]\right], \quad (7)$$

where λ is the trade-off coefficient and k is the ground truth class of sample i. This may avoid further generating misleading evidence for i by penalising those divergences from Dirichlet distribution over wrong classes and the uniform Dirichlet. For comparison's sake, three ECNN variants were explored regarding the loss function:

- ECNN-A was trained by (5).
- The loss function (7) was used to train ECNN-B and ECNN-C. For ECNN-B, λ is an annealing coefficient and its degree is controlled by a hyperparameter called 'annealing step' s shown in Table I, i.e., $\lambda = min(1.0, t/s)$ where t is the current training epoch number.
- For ECNN-C, λ is a constant coefficient, which is considered as a hyperparameter called 'tau' shown in Table I.

D. Performance Evaluation

1) Evaluation of Accuracy: First, we used the recall to evaluate the general efficacy of sEMG-based finger movement

recognition. As a multiclass classification problem, recall can be calculated by taking the macroaverage and microaverage. The macroaverage recall is calculated as:

$$r_M = \frac{1}{K}\sum_{j=1}^{K}\frac{tp_j}{tp_j + fn_j}, \tag{8}$$

where r_M is the macroaverage recall; tp and fn represent the number of true positives and false negatives; K is the number of finger movements and j refers to a specific one. It was employed here to measure the average per-class accuracy of such recognition because each finger movement is considered equally important, whereas the microaverage one favours bigger classes [53]. It would be further averaged over subjects for overall comparison. Second, to further investigate the accuracy of rejection-capable sEMG-based finger movement recognition, and for the sake of consistency with its related studies, the evaluation metric of the accuracy-rejection curve (ARC) [16], [23] was used here to compare the performance of CNN and ECNN variants in terms of their rejection rates. By varying the rejection threshold δ from 0 to 1, different pairs of RR and the corresponding accuracy (i.e., TAR) could be achieved when testing a trained classifier. For the overall comparison, we calculated the mean ARC for each model using 20 bins of RR under the CV scheme.

2) Evaluation of Reliability: As pointed in Sec. II, the reliability of the sEMG-based finger movement recognition could be evaluated by measuring the performance of the misclassification detection. The AUROC and AUPRC can then be used to calculate the model reliability and are noted as R_{AUROC} and R_{AUPRC}, which can be simply computed using the trapezoidal rule and Average Precision (AP) shown in (9), respectively. Consider a testing data set $\mathbf{D}^{(test)}$ with n samples and the number of positive (incorrect predictions) and negative samples (correct predictions) are represented by n_{pos} and n_{neg}, respectively,

$$AP = \frac{1}{n_{pos}}\sum_{i=1}^{n_{pos}} p(i), \tag{9}$$

where n samples will be sorted from high to low based on uncertainty estimates and i is the rank in the sequence of sorted positive samples; $p(i)$ is the precision at cut-off i. It has been proved that it is one of the most robust estimators to summarise the information in PRC [33].

Since each model has a specific class skew π on the misclassification detection, defined as n_{pos}/n, it is inappropriate to use R_{AUROC} and R_{AUPRC} for direct comparison between models. We recommend measuring the model reliability by R_{nAUPRC} for a robust and fair comparison, which is a normalised AUPRC. In this paper, we will present the results of R_{AUROC} and R_{AUPRC} for all models as a reference only and the ones of R_{nAUPRC} for the performance comparison. Boyd *et al.* [31] first proved that there is a region of PRC that is not achievable and the area of such an unachievable region depends on π. The nAUPRC was therefore proposed to account for this by using normalisation. As such,

$$R_{nAUPRC} = \frac{AP - AP_{min}}{AP_{max} - AP_{min}}, \tag{10}$$

where $AP_{max} = 1$, i.e., the theoretical maximum AUPRC; $AP_{min} = \frac{1}{n_{pos}}\sum_{i=1}^{n_{pos}}\frac{i}{n_{neg}+i}$, i.e., the theoretical minimum AUPRC proved by [31].

3) Evaluation Under Cross-Validation: There are two incompatible ways to compute the proposed evaluation metrics under nested CV. It can be calculated by either taking the mean of the results from each fold in the outer loop CV or aggregating the data from all folds into one first and then followed by the equations. Since merging assumes that the models are calibrated [54], which is not the case here, all evaluation metrics will be computed using the former approach here.

TABLE II
MACROAVERAGE RECALL OF THE CONVNETS WITH COMPARISONS

Models	ECNN-A	ECNN-C	CNN	ECNN-B
M±SD (%)	76.34±21.1	76.08±20.9	74.62±21.7	72.45±22.3
H0 (p) *	-	1 (7e-02)	0 (2e-06)	0 (6e-22)
H0 (p)	0 (2e-06)	0 (7e-05)	-	0 (2e-04)

* The Wilcoxon signed rank test was employed to compare the ECNN-A, which yields the highest mean macro average recall, with other models.
- The M and SD refer to the mean and standard deviation of macroaverage recall under nested cross-validation over 10 subjects.

VI. RESULTS

In all experiments, unless otherwise stated, the performance of CNN is taken as the baseline and compared with ECNN variants using statistical analysis with the Wilcoxon signed-rank test, where the null hypothesis assumes that there is no difference of evaluation results between the two models and will be rejected when p-value < 0.05. The difference in performance among ECNN variants will also be investigated.

A. Accuracy Analysis

Here we verified the accuracy of CNN and three ECNN variants. Table II shows that the ECNN-A and ECNN-C outperformed CNN overall in terms of classification accuracy on the *NinaPro DB5*. The average improvements, which were statistically significant, reached 1.72% and 1.46% respectively. It should be noted that the difference of accuracy between ECNN-A and ECNN-C was not statistically significant, and CNN significantly outperformed ECNN-B but with a difference of only 2.17% on accuracy. As such, one could notice that the rank of model accuracy was ECNN-A $\approx$ ECNN-C $>$ CNN $>$ ECNN-B. More comparisons of accuracy in terms of outer loop CV and each class are provided in Appendix II.

Fig. 4 shows the recognition accuracy comparison of rejection schemes in the form of ARC by revealing the trade-off relationship between the proportion of rejections and the resulting accuracy of the active predictions. One could observe clearly that ECNN-A was not substantially greater than ECNN-C and both of them outperformed CNN and ECNN-B in terms of recognition accuracy under the rejection condition, where the latter two also had approximately equal performance. With a specific focus on the regions where models had low RRs (i.e., $0 < RR \leq 15\%$), which may

TABLE III
RELIABILITY COMPARISON OF THE CONVNETS BY EVALUATING THE MISCLASSIFICATION DETECTION REGARDING UNCERTAINTY ESTIMATES

Scores	Models	R_{AUROC} M(%)±SD(%)	R_{AUROC} H0 (p)	R_{AUPRC} M(%)±SD(%)	R_{AUPRC} H0 (p)	R_{nAUPRC} M(%)±SD(%)	R_{nAUPRC} H0 (p)
$u_{nEntropy}$	CNN	82.06±3.72	-	54.71±8.57	-	47.64±8.05	-
	ECNN-A	83.17±4.72	0 (4.0e-03)	55.69±8.82	1 (5.9e-01)	49.32±8.82	1 (2.7e-01)
	ECNN-B	84.41±4.76	0 (1.9e-06)	63.37±8.65	0 (1.9e-08)	56.85±8.94	0 (2.6e-08)
	ECNN-C	83.65±4.59	0 (3.2e-04)	57.64±7.98	0 (1.1e-02)	51.39±7.98	0 (3.0e-03)
u_{nnmp}	CNN	83.18±3.68	-	57.09±7.05	-	50.33±6.37	-
	ECNN-A	83.60±4.46	1 (2.0e-01)	56.62±7.23	1 (2.3e-01)	50.35±6.91	1 (5.6e-01)
	ECNN-B	84.70±4.61	0 (2.4e-04)	63.80±8.28	0 (1.4e-07)	57.35±8.53	0 (1.5e-07)
	ECNN-C	83.92±4.62	0 (1.1e-02)	58.24±6.73	1 (3.1e-01)	52.02±6.65	1 (1.2e-01)
u_{vac} *	ECNN-A	71.88±7.51	-	44.96±11.48	-	37.03±11.70	-
	ECNN-B	83.02±5.20	0 (1.6e-11)	62.04±9.36	0 (3.5e-11)	55.32±9.68	0 (3.8e-11)
	ECNN-C	76.64±8.42	0 (6.5e-07)	51.22±9.77	0 (3.0e-08)	44.09±9.80	0 (2.4e-08)
u_{diss} *	ECNN-A	77.32±5.13	-	44.96±7.61	-	36.98±6.82	-
	ECNN-B	70.50±8.01	0 (3.9e-10)	42.06±8.85	0 (2.7e-03)	31.61±8.41	0 (1.0e-05)
	ECNN-C	74.75±5.41	0 (8.7e-06)	42.58±6.27	0 (5.3e-04)	33.99±5.32	0 (1.5e-04)
overall	CNN	82.76±3.74	-	55.96±8.03	-	49.06±7.50	-
	ECNN-A	83.47±4.60	0 (2.9e-02)	55.07±8.18	1 (1.7e-01)	48.56±8.11	1 (3.5e-01)
	ECNN-B	84.42±4.77	0 (1.8e-04)	63.37±8.65	0 (1.2e-07)	56.86±8.94	0 (2.9e-07)
	ECNN-C	83.85±4.46	0 (2.6e-03)	57.82±7.75	1 (8.2e-02)	51.60±7.70	0 (3.4e-02)

* The Wilcoxon signed rank test was employed to compare the ECNN-A with other ECNN variants.
- The M and SD refer to the mean and standard deviation of reliability measures under nested cross-validation over 10 subjects.
- Note that if the statistical results are conflicting, the results of R_{nAUPRC} shall be the standard ones.

be a reasonable target range in practical scenarios, all ECNN variants obtained higher accuracy than CNN.

B. Reliability Analysis

Here, we investigated the reliability analysis of CNN and three ECNN variants regarding different uncertainty estimates. Common uncertainty estimates such as $u_{nEntropy}$ and u_{nnmp} were considered for all models, whereas evidential uncertainty such as u_{vac} and u_{diss} only for ECNN variants. Furthermore, from the perspective of practical use, the overall uncertainty was noted as 'overall' in Table III and calculated by $max(u_{nEntropy}, u_{nnmp})$ for CNN and $max(u_{nEntropy}, u_{nnmp}, u_{vac}, u_{diss})$ for ECNN variants. Recall that the reliability analysis directly measures the quality of uncertainty estimates and only R_{nAUPRC} can be used for performance comparison between models.

From Table III, our first findings regarding the quality of uncertainty estimates were that all models with the uncertainty estimate u_{nnmp} achieved an overall highest R measured by either R_{AUROC}, R_{AUPRC}, or R_{nAUPRC} compared to other types of uncertainty estimate. Moreover, ECNN variants with the uncertainty estimate of either u_{vac} or u_{diss} alone obtained generally poor results of R. Our second findings regarding the R comparison between CNN and ECNN variants were that ECNN-B significantly outperformed CNN in any condition, where the highest improvement of reliability R_{nAUPRC} of 19.33% was achieved with the uncertainty estimate $u_{nEntropy}$ and 15.90% for the 'overall' uncertainty estimate. However, the difference in R_{nAUPRC} between CNN and ECNN-A was not significant in any condition, while that between CNN and ECNN-C was not either with u_{nnmp} only. Regarding the comparison of ECNN variants, ECNN-B achieved the highest R when using vacuity as the uncertainty estimate. Despite

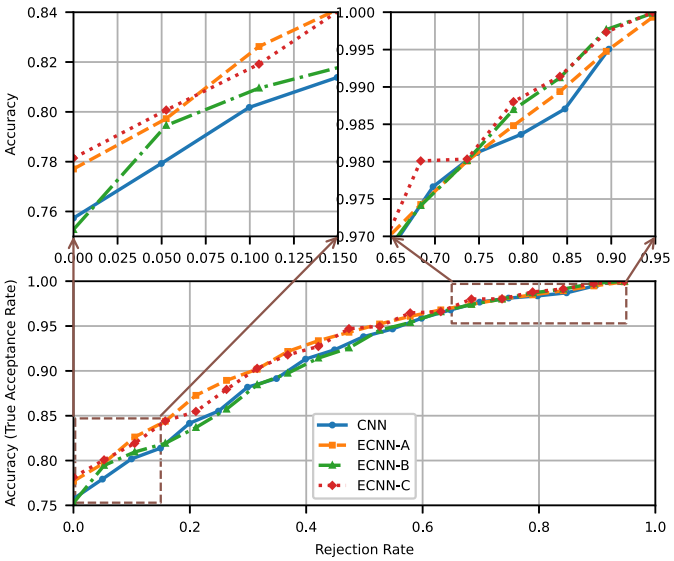

Fig. 4. The mean ARC plots of all models under CV scheme when considering the 'overall' uncertainty estimate.

ECNN-A performed best when using dissonance as the score of misclassification detection, the results of R_{nAUPRC} for all ECNN variants were generally quite low (no more than 36.98%). Eventually, the observed order of R_{nAUPRC} obtained with the uncertainty estimate of 'overall' was ECNN-B > ECNN-C > ECNN-A ≈ CNN.

VII. DISCUSSION

The current study had a particular focus on improving model efficiency and robustness, but not directly investigating model reliability. To fill this gap, we defined the model reliability

R as the quality of its uncertainty estimate and proposed an offline framework to quantify it. We focused our examination on the model reliability, and one implication of the results is that ECNN has great potential for complex and versatile finger movement recognition. Specifically, ECNN-C outperformed CNN with $p < 0.05$ in both accuracy and reliability with a difference of 1.46% in r_M (Table II), and 2.54% in R_{nAUPRC} with the 'overall' uncertainty (Table III), respectively. This suggests that the training of ECNN with a constant effect of KL should be applied when both model efficiency and reliability are weighted equally. Additionally, the loss function excluding the KL term is suggested for training the ECNN if model efficiency matters more than reliability. This is supported by the finding that ECNN-A achieved the best r_M of 76.34%, which was 1.72% higher than CNN with $p < 0.001$ (Table II) - but no significant difference of R_{nAUPRC} was found between them (Table III). Note that ECNN-A has shown its efficiency by presenting the SoA performance on NinaPro DB5 (Exercise A) since the best accuracy reported in the literature was 76.02%, achieved by taking an input of 300 ms sEMG signals to an ensemble classifier of three CNNs [4]. Conversely, ECNN is recommended to be trained by taking the annealing effect of KL term when there is a serious concern about model reliability, e.g., controlling a prosthetic limb for daily tasks to meet the needs of transradial amputee users. Our findings indicate that ECNN-B was determined as the most reliable one by showing improvements ranging from 14.25% to 19.33% in R_{nAUPRC} with different uncertainty measures (Table III), compared to CNN. Even though it was found less accurate than CNN where the difference in r_M was about 2% (Table II), its accuracy under the rejection scheme was approximately equal to CNN in general, and even better than CNN when RR is in a low range of 0% to 15% (Fig. 4).

Defining the comparable model reliability has implications for understanding how much an sEMG-based hand gesture classifier knows about its predictions, thereby providing us with general guidelines for designing such a reliable model which has the potential to improve its efficiency by rejecting making wrong predictions with the aid of its uncertainty estimate. The proposed framework of reliability analysis measures R by evaluating the performance of misclassification detection using the score of uncertainty estimate. Therefore, a model with a higher R could generate more discriminate uncertainty estimates, i.e., lower uncertainty estimates are assigned to correct predictions and vice versa. This implies that the value of R indicates how easily an optimal rejection threshold used for rejection-capable sEMG-based hand gesture recognition can be found. By measuring it, one can easily check the reliability of a model without the need to test its performance when allowing rejection by measuring several evaluation metrics such as RR, TAR, and TRR across a range of rejection thresholds. Additionally, we highly recommend using nAUPRC to measure R even though AUROC and AUPRC are commonly used for testing the performance of a misclassification detection task. One may observe the following order in each reliability analysis of a model with an uncertainty estimate: $R_{AUROC} > R_{AUPRC} > R_{nAUPRC}$. This finding is consistent with other research that reported ROC

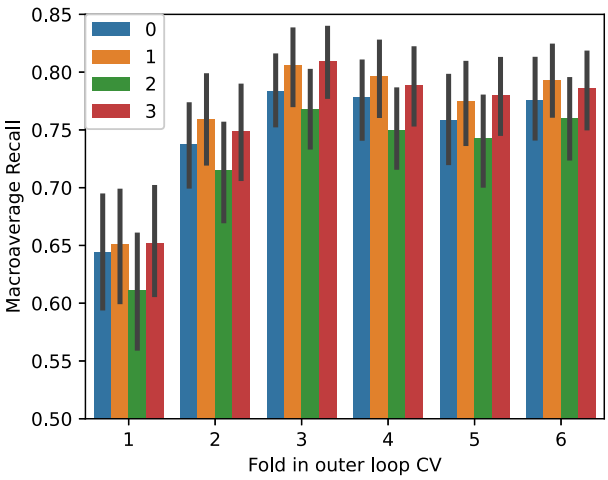

Fig. 5. Accuracy comparisons of the CNN and ECNN variants with nested cross validation. 0 is CNN and $1-3$ refer to ECNN-A, B and C.

plots usually make innocent impressions, whereas PR curves reveal the bitter truth, especially on imbalanced datasets [55]. We argue that the overall low value of R_{nAUPRC} may just exactly represent the situation in reality since averaging the nAUPRC under the CV can further reduce the effect of skew [31].

There are a few limitations that are important to note. First, one can not investigate the R of a model when it is tested with a classification accuracy of 100% or 0% because there are no positive or negative samples for misclassification detection in this case. We suggest setting R to 0 since such unusual results imply the model needs to be further investigated and can not be easily trusted. Second, even though we have demonstrated the potential of ECNN, the implications of its meaningful evidential uncertainty remain to be explored. Hypothetically, understanding the source of uncertainty is helpful to improve model robustness by making valid rejections. A potential research direction would then be to investigate the relationship between the proposed reliability analysis and the current studies on model robustness. Third, measuring the performance of misclassification detection with nAUPRC may not be the only way to investigate R. For example, it could be investigated by computing the area under the ARC or measuring the performance of out-of-domain data (e.g., unseen gestures or adversarial samples) detection. We encourage researchers to address the problem of sEMG-based hand gesture recognition from the perspective of model reliability together with model efficacy and robustness.

VIII. Conclusion

This paper has raised a concern about model reliability in sEMG-based hand gesture recognition. By defining the model reliability R as the quality of its uncertainty measures and providing an offline framework to investigate it, we have demonstrated that ECNN has great potential for classifying 12 individuated finger movements. Results on *NinaPro DB5* (Exercise A) with extensive comparisons across CNN and ECNN variants show that ECNN-A significantly outperformed

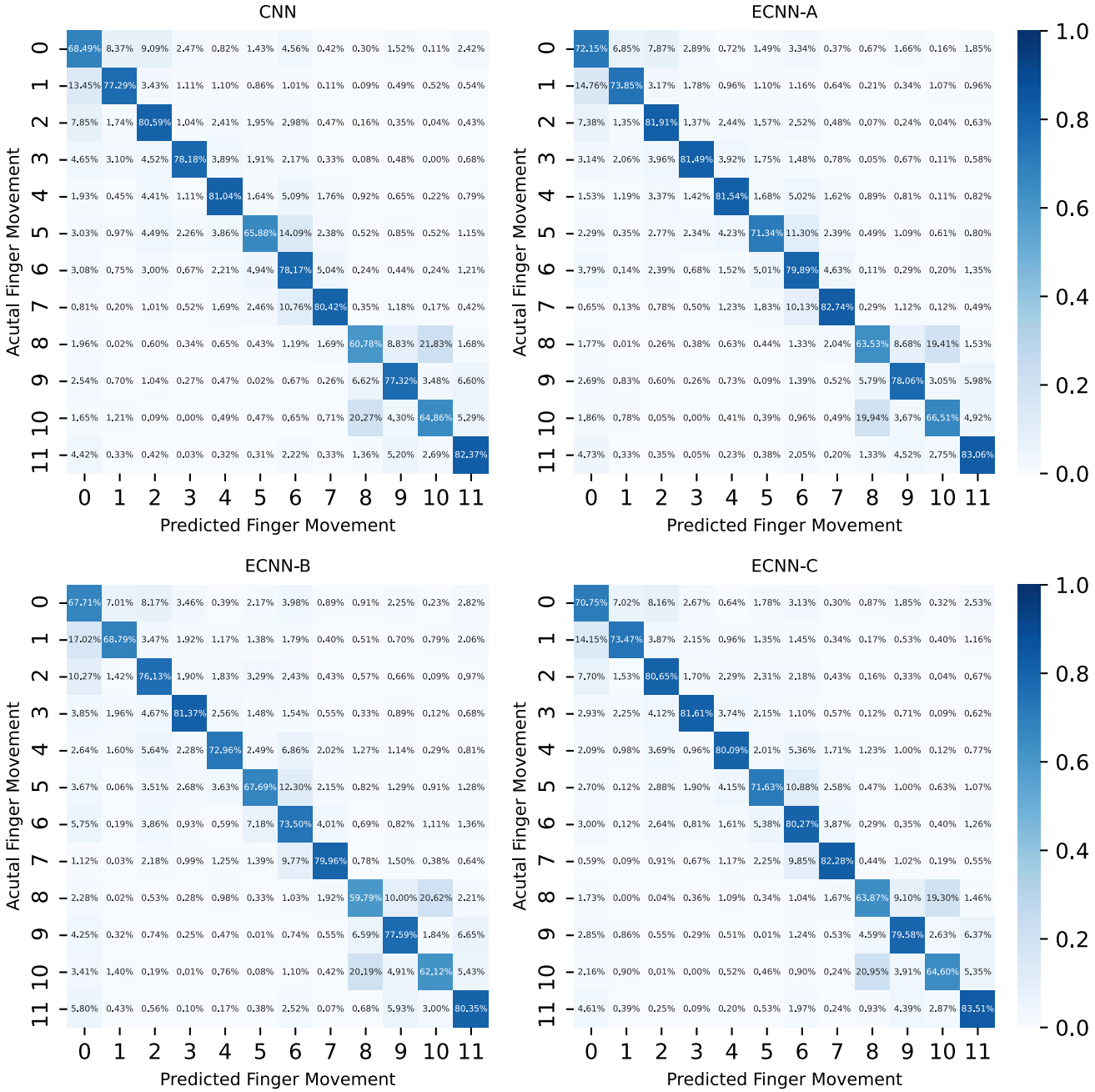

Fig. 6. Confusion matrices averaged over all subjects with nested cross validation of the CNN and ECNN variants.

CNN in model efficacy and achieved 0.32% higher accuracy than the SoA; ECNN-B has shown great reliability by presenting the highest improvement of 19.33% in R than CNN; ECNN-C has achieved the best trade-off between model efficacy and reliability by presenting 0.06% higher accuracy than the SoA and the best improvement of 7.87% in R than CNN. We encourage researchers to investigate model reliability and use the proposed reliability analysis as a supplementary tool for pursuing an accurate, robust, and reliable classifier, which is the overarching goal for sEMG-based hand gesture recognition. Our future work will focus on extending the reliability analysis of sEMG-based hand gesture recognition for amputee subjects and investigating if meaningful uncertainty estimates can be used to improve model robustness.

Appendix I
Algorithm for Model Training With Stratified Nested Cross-Validation

See Algorithm 1.

Appendix II
Supplementary Results on Accuracy Analysis

It can be seen that the rank of model performance regarding recognition accuracy averaged over all subjects is ECNN-A $\approx$ ECNN-C $>$ CNN $>$ ECNN-B on each fold in outer loop CV in Fig. 5. This is consistent with our main finding presented in Sec. VI-A. It is interesting to note that all models achieved the lowest accuracy on the 1st fold, indicating that there is significant variability between the first trial of sEMG and others.

Algorithm 1 Model Training With Stratified Nested CV

Input: $\mathbf{D} = \{\mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^{N}$, dataset includes segmented raw sEMG signals with labels, which has been divided by the repetition number from 1 to N. Define loss function J.
Output: Model parameters $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_N\}$ after training

for Each Repetition i **do** {**Outer Loop CV**}
  Load testing set $\mathbf{D}^{(test)} = \{\mathbf{X}_i^{(test)}, \mathbf{Y}_i^{(test)}\}$
  (**Hyperparameter Optimisation**)
  for Each trial of hyperparameter study k **do**
    Define the objective function of hyperparameter study with proposed hyperparameter search space
    Initialise a list $\mathbf{O}$ for collecting the objective values
    Initialise a list $\mathbf{L}$ for collecting the validation losses from the inner loop CV {**Inner Loop CV**}
    for Each repetition j (j **not** i) **do**
      Load validation set $\mathbf{D}^{(val)} = \{\mathbf{X}_j^{(val)}, \mathbf{Y}_j^{(val)}\}$
      Let the remaining dataset be the training set $D^{(train)}$
      Initialise θ_{ij} with random values
      $best\_val = inf$
      for Each epoch **do**
        Update the θ_{ij}
        if $J(\mathbf{X}^{(val)}, \mathbf{y}^{(val)}) < best\_val$ **then**
          $best\_val = J(\mathbf{X}^{(val)}, \mathbf{y}^{(val)})$
          $counter = 0$
        else
          $counter \mathrel{+}= 1$
        end if
        Stop training when $counter$ reaches to 10
      end for
      if the pruning is activated **then**
        Break the inner loop and move to the next hyperparameter study trial
      else
        Add $best\_val$ **to** the list $\mathbf{L}$
      end if
    end for
    Add the objective value $mean(\mathbf{L})$ **to** the list $\mathbf{O}$
  end for
  Load retraining dataset by combining both $\mathbf{D}^{(val)}$ and $\mathbf{D}^{(train)}$, i.e., $\mathbf{D}^{(retrain)} = \{\mathbf{X}^{(retrain)}, \mathbf{y}^{(retrain)}\}$
  Initialise the model parameters θ_i with random values
  Apply the optimal hyperparameter set which yields $min(\mathbf{O})$
  for Each epoch **do**
    Update the θ_i
    Stop training when $J(\mathbf{X}^{(retrain)}, \mathbf{y}^{(retrain)})$ reaches to $min(\mathbf{O})$
  end for
  Save model parameters θ_i **to** $\boldsymbol{\theta}$
end for
return $\boldsymbol{\theta}$

This may be because subjects need time to accommodate the Myo band to perform hand gestures.

Fig. 6 shows the average confusion matrices for CNN and three ECNN variants, where each annotated score represents the per-class normalised accuracy averaged over 6 outer CV trials across 10 subjects. It can be observed that all models have similar performance. For example, they all performed well in the classes '2 (Middle flexion)', '3 (Middle extension)', '7 (Little finger extension)', '9 (Thumb adduction)' and '11 (Thumb flexion)', while the pair (8, 10) is found more closely related than the other classes. Note that class 8 ('Thumb abduction') and class 10 ('Thumb extension') are commonly confused with each other. Regarding the per-class performance comparison of models for finger movement recognition, it can be observed that ECNN-A and ECNN-C performed better than CNN and ECNN-B on all classes except 'Ring flexion' (class 4) and 'Thumb extension', where ECNN-C achieved a slightly lower accuracy than CNN on these two classes, with the differences of 0.05% and 0.26% only. Furthermore, CNN outperformed ECNN-B on most classes except for 'Middle extension', 'Ring extension' (class 5), and 'Thumb adduction'.

REFERENCES

[1] I. M. Rezazadeh, M. Firoozabadi, H. Hu, and S. M. R. H. Golpayegani, "Co-adaptive and affective human-machine interface for improving training performances of virtual myoelectric forearm prosthesis," *IEEE Trans. Affect. Comput.*, vol. 3, no. 3, pp. 285–297, Jul. 2012.

[2] M. Atzori and H. Müller, "Control capabilities of myoelectric robotic prostheses by hand amputees: A scientific research and market overview," *Frontiers Syst. Neurosci.*, vol. 9, p. 162, Nov. 2015.

[3] M. Ghassemi *et al.*, "Development of an EMG-controlled serious game for rehabilitation," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 2, pp. 283–292, Feb. 2019.

[4] S. Shen, K. Gu, X.-R. Chen, M. Yang, and R.-C. Wang, "Movements classification of multi-channel sEMG based on CNN and stacking ensemble learning," *IEEE Access*, vol. 7, pp. 137489–137500, 2019.

[5] U. Côté-Allard *et al.*, "Deep learning for electromyographic hand gesture signal classification using transfer learning," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 4, pp. 760–771, Jan. 2019.

[6] W. Wei, Q. Dai, Y. Wong, Y. Hu, M. Kankanhalli, and W. Geng, "Surface-electromyography-based gesture recognition by multi-view deep learning," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 10, pp. 2964–2973, Oct. 2019.

[7] Y. Gu, D. Yang, Q. Huang, W. Yang, and H. Liu, "Robust EMG pattern recognition in the presence of confounding factors: Features, classifiers and adaptive learning," *Expert Syst. Appl.*, vol. 96, pp. 208–217, Apr. 2018.

[8] M. Z. U. Rehman *et al.*, "Multiday EMG-based classification of hand motions with deep learning techniques," *Sensors*, vol. 18, no. 8, p. 2497, 2018.

[9] Y. Lin, R. Palaniappan, P. De Wilde, and L. Li, "A normalisation approach improves the performance of inter-subject sEMG-based hand gesture recognition with a ConvNet," in *Proc. 42nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2020, pp. 649–652.

[10] L. Hargrove, K. Englehart, and B. Hudgins, "A training strategy to reduce classification degradation due to electrode displacements in pattern recognition based myoelectric control," *Biomed. Signal Process. Control*, vol. 3, no. 2, pp. 175–180, 2008.

[11] A. J. Young, L. J. Hargrove, and T. A. Kuiken, "The effects of electrode size and orientation on the sensitivity of myoelectric pattern recognition systems to electrode shift," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 9, pp. 2537–2544, Sep. 2011.

[12] D. Farina *et al.*, "The extraction of neural information from the surface EMG for the control of upper-limb prostheses: Emerging avenues and challenges," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 797–809, Jul. 2014.

[13] R. N. Khushaba, M. Takruri, J. V. Miro, and S. Kodagoda, "Towards limb position invariant myoelectric pattern recognition using time-dependent spectral features," *Neural Netw.*, vol. 55, pp. 42–58, Jul. 2014.

[14] H.-J. Hwang, J. M. Hahne, and K.-R. Müller, "Real-time robustness evaluation of regression based myoelectric control against arm position change and donning/doffing," *PLoS ONE*, vol. 12, no. 11, Nov. 2017, Art. no. e0186318.

[15] C. Prahm *et al.*, "Counteracting electrode shifts in upper-limb prosthesis control via transfer learning," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 5, pp. 956–962, May 2019.

[16] E. J. Scheme and K. B. Englehart, "A comparison of classification based confidence metrics for use in the design of myoelectric control systems," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2015, pp. 7278–7283.

[17] J. W. Robertson, E. J. Scheme, and K. B. Englehart, "Effects of confidence-based rejection on usability and error in pattern recognition-based myoelectric control," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 5, pp. 2002–2008, Sep. 2019.

[18] L. Wu, X. Zhang, X. Zhang, X. Chen, and X. Chen, "Metric learning for novel motion rejection in high-density myoelectric pattern recognition," *Knowl.-Based Syst.*, vol. 227, Sep. 2021, Art. no. 107165.

[19] A. Kopetzki, B. Charpentier, D. Zügner, S. Giri, and S. Günnemann, "Evaluating robustness of predictive uncertainty estimation: Are Dirichlet-based models reliable?" in *Proc. ICML*, vol. 139, 2021, pp. 5707–5718.

[20] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proc. ICLR*, 2014, pp. 1–10.

[21] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proc. ICLR*, 2017, pp. 1–17.

[22] J. Su, D. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.

[23] M. S. A. Nadeem, J. Zucker, and B. Hanczar, "Accuracy-rejection curves (ARCs) for comparing classification methods with a reject option," in *Proc. 3rd Int. Workshop Mach. Learn. Syst. Biol.*, Ljubljana, Slovenia, Sep. 2009.

[24] L. J. Hargrove, E. J. Scheme, K. B. Englehart, and B. S. Hudgins, "Multiple binary classifications via linear discriminant analysis for improved controllability of a powered prosthesis," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 1, pp. 49–57, Feb. 2010.

[25] M. Sensoy, L. M. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Proc. NIPS*, 2018, pp. 3183–3193.

[26] X. Zhao, F. Chen, S. Hu, and J. Cho, "Uncertainty aware semi-supervised learning on graph data," in *Proc. NIPS*, vol. 33, 2020, pp. 12827–12836.

[27] A. Josang, J.-H. Cho, and F. Chen, "Uncertainty characteristics of subjective opinions," in *Proc. 21st Int. Conf. Inf. Fusion (FUSION)*, Jul. 2018, pp. 1998–2005.

[28] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proc. ICML*, vol. 37, 2015, pp. 1613–1622.

[29] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. NIPS*, 2017, pp. 6402–6413.

[30] H. Scherberger, "Neural control of motor prostheses," *Current Opinion Neurobiol.*, vol. 19, no. 6, pp. 629–633, Dec. 2009.

[31] K. Boyd, J. Davis, D. Page, and V. S. Costa, "Unachievable region in precision-recall space and its effect on empirical evaluation," in *Proc. ICML*, 2012, pp. 639–646.

[32] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Dec. 2005.

[33] K. Boyd, K. H. Eng, and C. D. Page, "Area under the precision-recall curve: Point estimates and confidence intervals," in *Machine Learning and Knowledge Discovery in Databases* (Lecture Notes in Computer Science), vol. 8190. Berlin, Germany: Springer, 2013, pp. 451–466.

[34] A. Ashukha, A. Lyzhov, D. Molchanov, and D. P. Vetrov, "Pitfalls of in-domain uncertainty estimation and ensembling in deep learning," in *Proc. ICLR*, 2020, pp. 1–30.

[35] A. P. Dempster, "A generalization of Bayesian inference," in *Classic Works Dempster-Shafer Theory Belief Functions* (Studies in Fuzziness and Soft Computing), vol. 219. Berlin, Germany: Springer, 2008, pp. 73–104.

[36] T. Reineking, "Belief functions: Theory and algorithms," Ph.D. dissertation, Dept. Fac. Eng., Univ. Bremen, Bremen, Germany, 2014.

[37] A. Jøsang, *Subjective Logic—A Formalism for Reasoning Under Uncertainty* (Artificial Intelligence: Foundations, Theory, and Algorithms). Cham, Switzerland: Springer, 2016.

[38] K. W. Ng, G.-L. Tian, and M.-L. Tang, *Dirichlet and Related Distributions: Theory, Methods and Applications*. Hoboken, NJ, USA: Wiley, 2011.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.

[40] K. Englehart, B. Hudgins, and P. A. Parker, "A wavelet-based continuous classification scheme for multifunction myoelectric control," *IEEE Trans. Biomed. Eng.*, vol. 48, no. 3, pp. 302–311, Mar. 2001.

[41] A. Jaramillo-Yanez, M. E. Benalcazar, and E. Mena-Maldonado, "Real-time hand gesture recognition using surface electromyography and machine learning: A systematic literature review," *Sensors*, vol. 20, no. 9, p. 2467, 2020.

[42] B. Hudgins, P. Parker, and R. N. Scott, "A new strategy for multifunction myoelectric control," *IEEE Trans. Biomed. Eng.*, vol. 40, no. 1, pp. 82–94, Jan. 1993.

[43] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, Jul. 2015, pp. 448–456.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[45] S. Varma and R. Simon, "Bias in error estimation when using cross-validation for model selection," *BioMed Central*, vol. 7, no. 1, p. 91, 2006.

[46] D. Krstajic, L. J. Buturovic, D. E. Leahy, and S. Thomas, "Cross-validation pitfalls when selecting and assessing regression and classification models," *J. Cheminformatics*, vol. 6, no. 1, p. 10, 2014.

[47] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Roy. Stat. Soc. B, Methodol.*, vol. 36, no. 2, pp. 111–133, Jan. 1974.

[48] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. NIPS*, 2011, pp. 2546–2554.

[49] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," *J. Mach. Learn. Res.*, vol. 28, pp. 115–123, Jun. 2013.

[50] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Learning and Intelligent Optimization* (Lecture Notes in Computer Science), vol. 6683. Berlin, Germany: Springer, 2011, pp. 507–523.

[51] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2623–2631.

[52] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro, and D. Sculley, "Google vizier: A service for black-box optimization," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1487–1495.

[53] 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.

[54] G. Forman and M. Scholz, "Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement," *ACM SIGKDD Explor. Newslett.*, vol. 12, no. 1, pp. 49–57, 2010.

[55] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, Mar. 2015, Art. no. e0118432.