IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 June 2023, accepted 29 July 2023, date of publication 4 August 2023, date of current version 15 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3301872

RESEARCH ARTICLE

Cascade Windows-Based Multi-Stream Convolutional Neural Networks Framework for Early Detecting In-Sleep Stroke Using Wristbands

SANGHOON JEON[1], (Member, IEEE), YANG-SOO LEE[2], AND SANG HYUK SON[3], (Life Fellow, IEEE)

[1]Department of Emergency Medicine, College of Medicine, Hanyang University, Seoul 04763, South Korea
[2]Department of Physical Medicine and Rehabilitation, School of Medicine, Kyungpook National University, Daegu 41944, South Korea
[3]Department of Electrical Engineering and Computer Science, DGIST, Daegu 42988, South Korea

Corresponding author: Sang Hyuk Son (son@dgist.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2021R1I1A1A01040943, and in part by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) under Grant 2018R1A5A1060031.

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 A stroke, particularly when it occurs during sleep, is likely to have a negative prognosis due to delayed detection. Timely and early detection plays a vital role in ensuring prompt administration of reperfusion therapy and preventing permanent disabilities. To address this, we propose a wearable system comprising two wristbands that monitor asymmetric motion patterns (hemiparesis) during sleep. A novel deep learning framework called Early Detection of In-sleep Stroke (EDIS) serves as the core engine for stroke detection during sleep. The framework employs cascading windows of various sizes for convolutional neural networks (CNNs) to enhance both the detection performance and the detection time. We utilize 1D accelerometer sensor data from both hands to generate 2D matrix images, which serve as input for multiple CNN models. Predictions from these models are combined using blending ensemble learning to make a final decision. Although the EDIS framework requires a larger parameter size and longer inference time due to its network architecture with multiple CNNs, it outperforms five single-CNN models by improving detection performance and reducing detection time. Extensive evaluation results demonstrate that EDIS framework accurately and quickly detects in-sleep stroke within the deadline (3 hours). EDIS-Resnet50 has the best classification performance out of the ten DL model candidates, with an F1-score of 0.955 (0.950, 0.960). We believe that our framework will be a fundamental component of real-time stroke monitoring systems, contributing to a reduction in mortality rates among patients suspected of having a stroke.

INDEX TERMS Deep learning, ensemble learning, stroke detection, sleep, wearable computing.

I. INTRODUCTION

Stroke is a medical emergency. It is the fifth leading cause of death in the United States [1]. Every year, more than 795,000 people in the United States have a stroke (Of those, 610,000 are first stroke and 185,000 are recurrent stroke) [2].

Stroke can be classified into two categories: ischemic stroke caused by blockage of a blood vessel in the brain and hemorrhagic stroke caused by bleeding [3]. The majority (87%) of the strokes are the ischemic stroke in nature [4]. During a stroke, every minute counts because the stroke cuts off oxygen and nutrients from the blood and causes brain cells to die within minutes. Depending on what region of the brain is affected, several symptoms occur, such

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

as weakness, numbness, speech abnormalities, confusion, headache, visual abnormalities, and dizziness [5]. Therefore, it is crucial to recognize these symptoms early and treat the stroke as quickly as possible so that the damage is not permanent. Time from stroke onset to hospital arrival (onset-to-door time) is important for determining clinical outcomes, i.e., early arrival ($\leq$4.5 hours) is likely to get better outcomes [6]. Thrombolytic therapy with intravenous (IV) tissue plasminogen activator (t-PA) within 4.5 hours after stroke onset is used for the standard treatment of stroke and heart attack [7], [8], [9]. Within the eligible time, thrombolytics help limit stroke damage and disability. Otherwise, the risk of bleeding may increase.

Additionally, we need to consider the emergency medical service (EMS) response time from stroke detection to hospital transfer. In fact, the EMS response time takes 83 minutes before COVID-19 and 103.7 minutes in the early stages of COVID-19, respectively [10]. Due to the delay before hospitalization, this work aims at detecting a stroke during sleep within 3 hours (180 minutes), i.e., the deadline of stroke detection.

Stroke can occur during sleep. Approximately 25% of all strokes occur during sleep without knowing the exact time when stroke symptoms began [11]. The stroke during sleep is difficult to recognize in time and may not be detected until the next morning. This phenomenon is called Wake-Up Stroke (WUS), and it refers to a patient who wakes up with stroke symptoms after falling asleep normally [12]. WUS poses a therapeutic dilemma for thrombolytic therapy because it is difficult to know when a stroke occurs. In order to distinguish a stroke during sleep from WUS, we introduce the term in-sleep stroke.

Clinical imaging approaches [13] using non-contrast computed tomography (CT) [14], and Magnetic Resonance Imaging (MRI) [15] have been proposed to identify eligible patients within 4.5 hours for thrombolytic therapy. According to the American Heart Association/American Stroke Association (AHA/ASA) guidelines for healthcare professionals, MRI findings (DWI+/FLAIR-) can be useful for selecting eligible patients who may benefit from IV t-PA treatment [16]. However, MRI-based approaches have a good spatial resolution but lack temporal resolution, making them unsuitable for real-time stroke detection.

For the real-time detection and assessment of stroke, non-invasive sensor technologies have been developed mainly using two sensors: Electroencephalography (EEG) and Accelerometer. The basic principle behind these sensor technologies is the use of stroke-induced lateralization properties. That is, hemispheric dominance and lateralization of brain activity are commonly observed after stroke [17]. Motor lateralization, which shows paralysis on one side of the body (also called hemiparesis or hemiplegia), is also common in stroke survivors [18].

EEG is used to detect abnormal frequencies between sides of the brain in lateralization detection of brain activity. Abnormal frequencies, such as increased delta-to-alpha ratio (DAR) or (delta+theta)/alpha+beta ratio (DTABR), are observed during ischemic stroke due to lowering of cerebral blood flow (CBF) in EEG signals [19]. Symmetrical metrics such as Brain Symmetry Index (BSI) [20] and modified BSI [21] have also been proposed for the assessment and prognosis of stroke by quantifying the difference in spectral characteristics between the left and right hemispheres.

Generally, the 10-20 system is an internationally accepted and a standardized EEG placement method using 16 or 21 electrodes [22]. Portable EEG devices, e.g., the Muse headband by InteraXon Inc. [23] and AlphaStroke by Forest Device Inc. [24], have been developed to detect stroke by providing wireless functionality and reducing the number of electrodes. Nevertheless, EEG is still cumbersome to use in a sleeping environment, and electrodes may not be properly positioned during sleep.

Accelerometer-based measurements, a.k.a., actigraphy, are helpful for identifying movement impairments in the upper extremity (UE), classifying the level of activity limitations, and performing rehabilitation based on recovery goals [25]. Feasibility studies on upper extremity asymmetry showed that, for stroke patients, the asymmetry ratio index for the 24 hours (AR2_24h) positively correlated with the National Institutes of Health Stroke Scale (NIHSS), a clinical standard for measuring hemiparetic severity in stroke [26], [27]. In addition, AR2_24h parameter can be used as an additional prognostic predictor to the modified Rankine Scale (mRS), which measures the severity of disability after 90 days [28].

However, AR_24 requires a long actigraphic recordings (more than 24 hours) to evaluate the asymmetry. An automated hamiparesis scoring measurements [29] or novel asymmetric parameters [30] have been proposed to address the long recording problem by formulating well-defined upper limb movements for a short time.

Nevertheless, classifying the motor asymmetry during sleep is difficult because sleep motions occur unconsciously and asynchronously over long periods of time. We propose a novel approach for early detection of stroke during sleep within the deadline ($\leq$3 hours) in real-time. To the best of our knowledge, it is one of the first approaches to detect strokes (asymmetric motion patterns) during sleep in real-time using long sliding windows. In particular, we create two-dimensional matrix images from two accelerometer sensor data. We then apply the images to a new developed deep learning (DL) model tailored to in-sleep stroke detection based on sliding windows.

In our previous study, RISK-Sleep [31] showed insight into stroke detection during sleep by classifying asymmetric motion patterns based on sliding windows. Fig. 1(a) shows our motivation of developing an in-sleep stroke detection system using wearable devices. By using wearable devices containing Inertial Measurement Unit (IMU) sensors, as shown in Fig. 1(b), preliminary study showed the system could

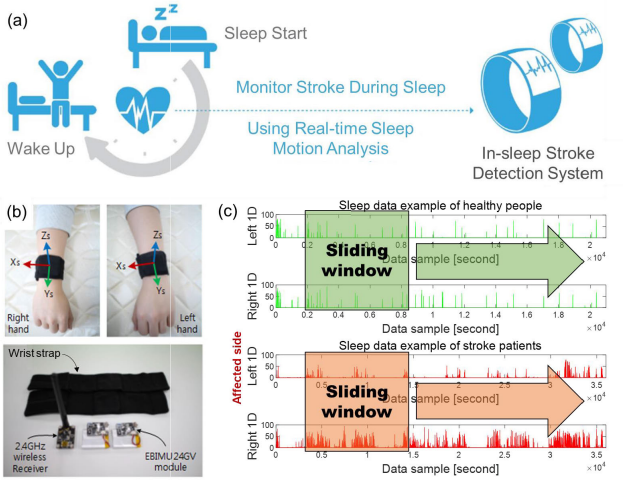

FIGURE 1. Motivation of developing a system for early detecting in-sleep stroke.

detect in-sleep stroke with simple Machine Learning (ML) models. Clearly, a stroke patient with hemiparesis, could not move the affected arm freely on the affected side, resulting in reduced sleep motion patterns in both intensity and frequency, as shown in Fig. 1(c).

Developing a system to detect strokes during sleep presents several key challenges, including the selection of optimal window size, the generation of distinctive asymmetric features, and the building of effective classification models. In this study, we enhance the RISK-Sleep by gathering additional sleep data and developing a new deep-learning framework, called EDIS-framework.

To achieve early detection of strokes during sleep, we propose a novel framework called EDIS framework, which utilizes multi-stream Convolutional Neural Networks (CNNs) based on cascade windows. The primary objective of the proposed EDIS framework is to identify asymmetric motion patterns (hemiparesis) during sleep. The uniqueness of the EDIS framework lies in its network design, which leverages aligned cascading sliding windows. By employing the same deep learning model on windows of varying sizes and aligning the present time, the EDIS framework improves classification performance and reduces the detection time.

The major contributions of this paper are as follows:

- We propose a system using two wristbands for the timely detection of strokes during sleep within the deadline. In particular, we develop a DL model framework, the core engine of the system, to accurately and quickly detect an in-sleep stroke.
- To classify the asymmetric sleep motion pattern associated with stroke patients, specifically hemiparesis, we introduce a novel feature called a two-dimensional matrix image. The matrix image is generated by accelerometer sensor data on sliding windows of both hands. Since the asymmetric motion pattern is reflected in the matrix image according to the window size, we conduct a comprehensive evaluation of classification performance to determine the optimal window size.
- We develop a new DL framework, called EDIS framework. EDIS framework is a network architecture combined with homogeneous multi-CNN models using blending ensemble learning. The design of the network architecture includes multiple inputs to improve classification performance and aligns the cascade windows at the current time to reduce detection time.
- Utilizing a network architecture incorporating multiple CNN models increases the number of parameters and increases inference time, but the EDIS framework improves classification performance and reduces detection time compared to single CNN models in extensive evaluation results.

The rest of the paper is organized as follows. Section II introduces sliding window-based techniques to classify time-series data in related work. Section III describes experimental setup, data pre-processing, and EDIS-framework in detail. In Section IV, three performance evaluations are performed for classification performance, computation performance, and detection time. Finally, this paper concludes in Section V.

II. RELATED WORK

In this section, we discuss a sliding window based approach for detection of time-series motion patterns. A sliding window approach uses sub-sequences of streams data from accelerometer sensors, called a window, and classifies activities as the window moves. Window length directly impacts performance on recognition performance and latency. That is, a small-sized window speeds up activity recognition but can slow recognition performance. For Human Activity Recognition (HAR), it is recommended to use on-body wearable sensors [32] and smartphone sensors [33] for 1∼2 seconds and 2.5∼3.5 seconds, respectively.

Accelerometer-based wrist-worn devices (e.g., smartwatch and smartband) have been widely developed for monitoring physical activities, such as walking, running, biking, and sleeping, due to their portability and relatively low cost [34]. Among them, sleep monitoring is a representative application that uses smart bands/watches to manage health and sleep. For example, activity-based sleep-wake monitoring, a.k.a., actigraphy, has become an important assessment tool in sleep research and sleep medicine [35]. In addition, functions such as sleep apnea detection [36], sleep posture estimation [37], and sleep-wake cycles estimation [38] have also been developed.

HAR is a problem of adequately matching time-series sensor data with well-defined movements. There are several challenges, such as massive sensor data generated per second, the temporal nature of sensor data, and the unclear relationship between sensor data and activity patterns. Classical approaches for HAR extract handcrafted features from times-series sensor data in a fixed-size window and train machine learning models, such as a decision tree and

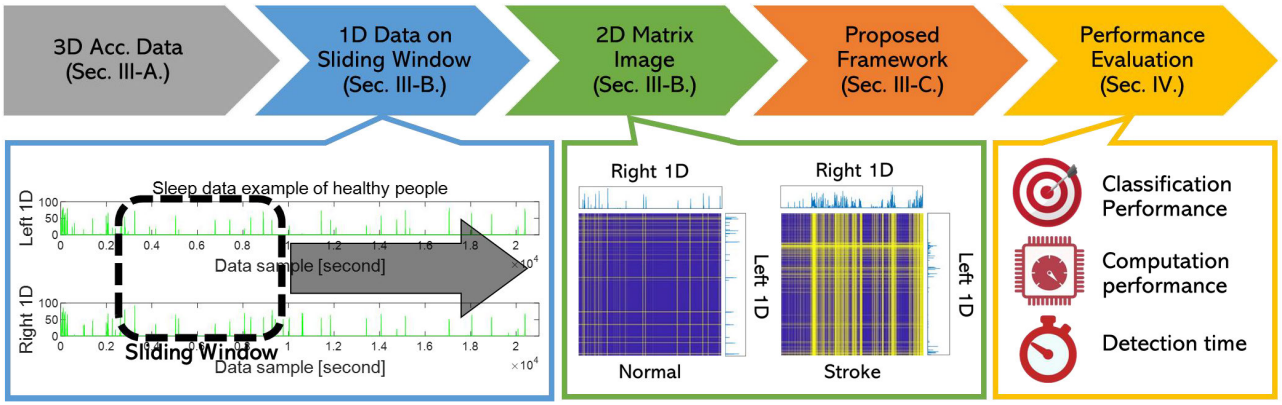

FIGURE 2. Overview of EDIS framework development process.

a support vector machine. The difficulty is that feature engineering requires expertise in the field, i.e., background and knowledge in the application domain.

With advances in deep learning technology, DL models have significantly been improved by designing only model architectures. Deep learning introduces a concept of end-to-end learning where results are obtained by only providing input data to the DL model without feature engineering.

Convolutional Neural Networks (CNNs), a representative model structure of DL for computer vision, outperform conventional machine learning models such as support vector machines and k-nearest neighbors [39]. By converting time-series sensor data into activity image data, CNNs can be used for HAR [40]. CNNs are also used a feature extractor to improve classification performance [41].

Recurrent neural networks (RNNs) has been used for sequential data, such as text, audio, and video. A long short-term memory (LSTM) network is a model that improves the problems of long-term dependencies of RNNs by including a memory to store temporal dependencies [42]. LSTM is more suitable for time series classification than CNNs because CNNs require a fixed-size window of sensor data, but LSTMs do not strictly require a fixed-size window [43].

Nowadays, hybrid deep learning model architectures have been developed based on CNNs and LSTM, such as CNN-LSTM [44] and DeepConvLSTM [45]. The models combines the strengths of CNN and LSTM. That is, CNNs are used to extract local features as feature extractors and LSTM is used to capture time-dependent relationships from the local features. However, the hybrid model design has a lot of freedom and requires expertise in model design.

Compared to existing works, conventional approaches uses short windows to extract specific motion patterns. However, sleep motion occurs unconsciously, making it challenging to classify abnormal motion patterns using a short window. To address the challenge, we use a long sliding window to extract abnormal sleep motion patterns (hemiparesis) and make 2D images. We also propose a new DL framework using multiple CNNs with the goal of detecting stroke during sleep within the deadline (3 hours). We note that the proposed framework uses a CNN approach because the input data is a 2D image using a fixed long sliding window.

III. METHOD

This section outlines the EDIS framework development process, as shown in Fig. 2. We first describe the experimental setup and data preprocessing based on sliding windows. We propose EDIS framework using two-dimensional matrix images. Finally, we assess the performance of the framework by considering classification performance, computational performance, and detection time.

A. EXPERIMENTAL SETUP

This section describes the experimental setup of the hardware & software prototype for designing the EDIS framework and also describes the sleep dataset we collected.

1) HARDWARE AND SOFTWARE PROTOTYPE

Our prototype consists of hardware in the form of a wristband, as shown in Figure 1(b), and software that collects motion data during sleep.

For the hardware setup, we configure a set of wearable devices consisting of two transmitters (9-axis Inertial Measurement Unit (IMU) module), one receiver (2.4 GHz wireless communication module), and a laptop. The embedded modules were developed by the E2BOX company in the Republic of Korea. EBIMU24GV2 module is used for the two transmitters, and EBRF24GRCV module is used for the receiver. The transmitter size is 32mm×24mm, and the sampling rate is 100Hz.

In the software setup, the data logging software is developed using an open source java library that provides serial port communication, *RxTx*. The software is installed on the laptop and configured to launch the program when a user initiates a sleep experiment. Healthy participants perform the sleep experiment at their home, and stroke patients are conducted in hospital with the help of medical staff.

2) SLEEP DATASET

We collect sleep data[1] using our prototype. Our sleep data consists of 46 healthy people and 43 stroke patients with hemiparesis. We noted that the EDIS framework uses a sliding window-based approach. For our dataset to train and test the DL models, we generate motion patterns from the sleep data by shifting two sliding windows by 30 minutes.

Depending on the severity of the stroke, the degree of paralysis on one side of the body varies. This severity can be assessed using manual muscle testing (MMT). MMT is a commonly used assessment tool to assess muscle dysfunction and deficits in stroke patients [46]. The MMT evaluates three upper and three lower extremity muscles (shoulder abduction, elbow flexion, wrist extension, hip flexion, knee extension, and ankle dorsiflexion) on a six-point scale (0: None, 1: Trace, 2: Poor, 3: Fair, 4: Good, 5: Normal) [47]. Among these evaluation indicators, the wrist sensor directly detects the movement of the forearm and is greatly affected by the elbow joint. Therefore, we use the elbow flexion score as an indicator of stroke severity.

The MMT score is 2.558 (2.094, 3.023) on the affected side and 4.7442 (4.596, 4.892) on the unaffected side in the stroke dataset. The score is mean (95% confidence Interval). Healthy people have an MMT score of 5 on both sides.

3) DATASET CLEANING

Before processing sleep motion data, data cleaning task is necessary for reducing errors in performance.

It is important to use high-quality data to improve performance in making deep learning models. An origin sleep data was generated by manually checking sleep data and removing the data after participants woke up. Besides, we remove the sleep data at the initial 30 minutes and the last 30 minutes to prevent noise caused by experimental setup and motions before getting asleep.

During normal REM sleep, we experience temporary muscle paralysis [48]. Because of the REM sleep, we sometimes observe few motion data on sliding windows in the sleep datasets for both stroke patients and healthy people. This is natural sleep rhythms. An abnormal sleep pattern, i.e., stroke, is detected by comparing the differences in left and right movements in these movement patterns.

We define a new metric called Motion Contention Rate (MCR) to evaluate data quality in a sliding window ω to remove low-quality data, as shown in Eq. 1. We collect sleep data for different window sizes to train a DL model. Data sets less than 25% MCR are deleted for each window size. The 25% criterion has been established empirically.

$$\text{MCR} = \frac{\text{length of nonzero } (\omega^L + \omega^R)}{\text{length of } \omega} \tag{1}$$

[1]All participants in this work were approved by the Institutional Review Board (IRB) of DGIST and the IRB of Kyungpook National University Hospital, respectively. Sleep data collection was performed from May 1, 2015 to December 31, 2019. The IRB approval number is KNUMC_13-1057, KNUH_2013-12-030-001, DGIST_160114HR00404, and DGIST_171221HR04005.

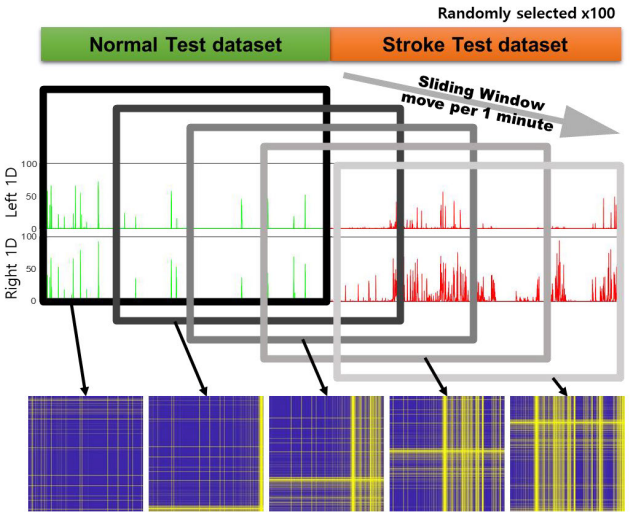

FIGURE 3. Simulation study for detection time in real-time in-sleep stroke detection.

4) SIMULATION STUDY FOR DETECTION TIME

Detection time is a critical metric for evaluating the DL model in early-detection applications. However, it is difficult to verify the developed model by directly applying it to stroke patients. Therefore, we simulate a stroke onset event during sleep by using the collected sleep dataset.

After training the deep learning model with the training dataset, we can simulate a stroke onset event with the remaining test dataset. We can generate a simulation data, a total length of $2 \times \omega$, for the stroke onset by concatenating normal test dataset and stroke test dataset, as shown in Fig. 3.

By moving a sliding window by 1 minute, we investigate the minimum time that the trained DL model detects a stroke. If detection is failed by ω minutes, the detection time is recorded as ω minutes. This detection time test is repeated 100 times.

B. DATA PRE-PROCESSING

Sleep data consists of stream data over a long time. We generate a motion dataset by moving a sliding window on sleep data in 30-minute increments. We construct several motion datasets with different window sizes to find the optimal sliding window size. This section describes how to perform pre-processing on sleep data.

1) 1-DIMENSIONAL MAGNITUDE ACCELEROMETER DATA

We use 9-axis Inertial Measurement Unit (IMU) sensors (3-axis accelerometer sensors, 3-axis gyroscope sensors, and 3-axis magnetometer sensors) to capture motion patterns. The IMU sensors are on the wristbands. This work uses only raw data from 3-axis accelerometer sensors. The first process is to convert 3-dimensional accelerometer data ξ to 1-dimensional (1D) magnitude data ξ_m using Euclidean distance. Because sleep motion occurs unconsciously, there is no uniform pattern along any particular axis. This

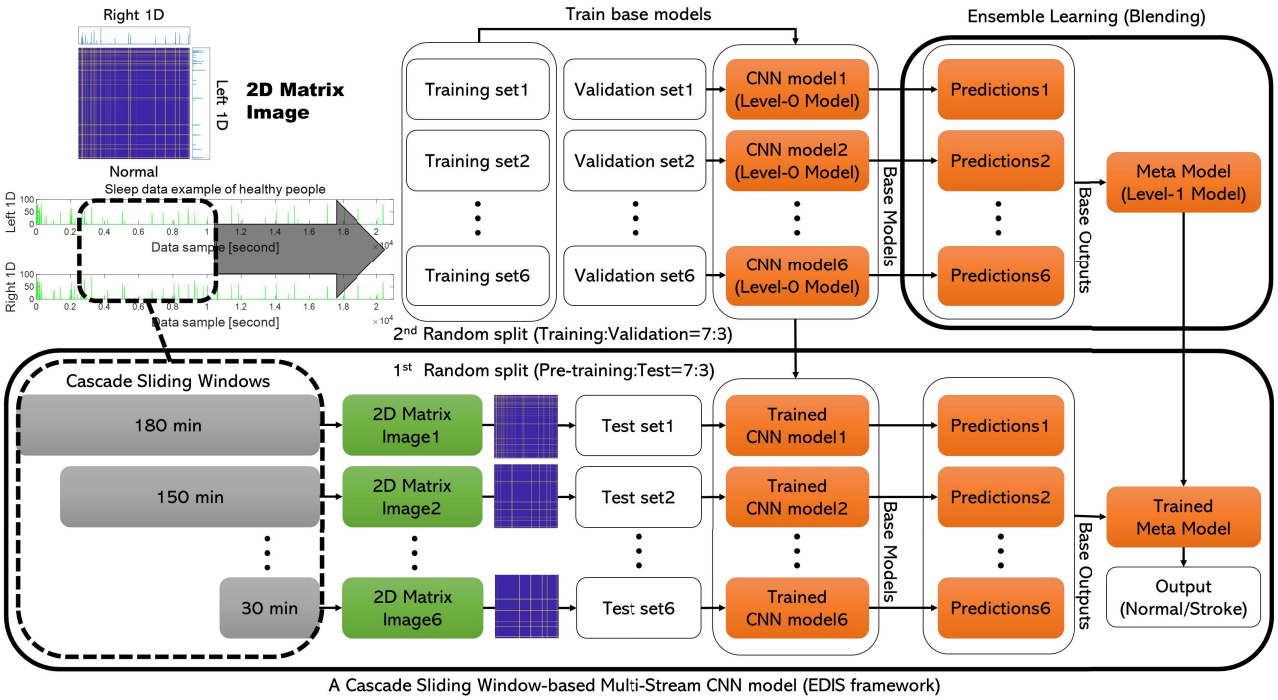

FIGURE 4. EDIS framework using a blending ensemble learning.

transformation removes data noise along the axis and focuses solely on the intensity and frequency of motion.

$$\xi_m(i) = \sqrt{(\xi_x(i))^2 + (\xi_y(i))^2 + (\xi_z(i))^2} \tag{2}$$

where x, y, and z denote each axis on the 3-axis accelerometer sensor data, and i denotes the data point. The transformed 1D magnitude data ξ_m is sampled at the rate of 100Hz. Because this work observes long-term time-series data patterns, the transformed data is also resampled at 1Hz to reduce the computational workload required for the following pre-processing procedures.

2) SLIDING WINDOW

We use a long-sized sliding window to effectively extract the differences in sleep motion between healthy people and stroke patients.

As a sliding window moves with time, it collects sequences of sensor data. A sliding window ω can be defined as:

$$\omega(t) = [\xi_m(t-s+1), \ldots, \xi_m(t-1), \xi_m(t)] \tag{3}$$

where t represents the current date point, and s denotes the sliding window size.

We use the minimum length of sliding window with 30 minutes, empirically. Because we observed that asymmetric sleep motion patterns were well expressed when sliding windows with window sizes greater than 30 minutes were used. We increase the size in 30-minute increments to find the optimal window size that provides the best performance.

3) 2-DIMENSIONAL ASYMMETRIC MATRIX

There are 1-dimensional sequences of sensor readings ω^L and ω^R, on Left sliding window and Right sliding window respectively. We create 2-dimensional Matrix M by inner product of the two 1D sequences as follow.

$$\omega^L = \{d(t-s+1), \ldots, d(t-1), d(t)\} \tag{4}$$
$$\omega^R = \{d(t-s+1), \ldots, d(t-1), d(t)\} \tag{5}$$
$$M = \omega^L \cdot [\omega^R]^{-1} \tag{6}$$

where d_t denotes the sensor reading at time t.

4) IMAGE DATA

For the use in the pre-trained CNNs as input data, we need to transform 2D matrix data into image data. Matlab provides functions for the transformation. *imagesc* displays the array data as an image data I_{img} using the full range of colors in the colormap. *ind2rgb* converts image data I_{img} into RGB image format data I_{rgb} as input image data of the CNN model.

$$I_{img} = imagesc(M) \tag{7}$$
$$I_{rgb} = ind2rgb(I_{img}) \tag{8}$$

In pre-trained CNN models, the size of input images for each model is different. Therefore, it is necessary to adjust the input size of the dataset. We resize the input image file into the input size of pre-trained CNN models to apply our dataset. We perform resizing using a *imresize* function provided by Matlab.

C. PROPOSED FRAMEWORK

The basic idea of the EDIS framework comes from the two-stream multi-channel convolutional neural networks [49], [50], [51]. We aim at building a DL model with improved performance similar to these approaches using multiple inputs to improve performance.

In order to detect in-sleep stroke accurately and quickly, we propose a new framework, called EDIS framework, as shown in Fig. 4. EDIS framework has a network architecture consisting of several homogeneous pre-trained CNN models and one meta-model. We note that the multiple CNNs are retrained using our dataset in the EDIS-framework, i.e., transfer-learning.

For making the EDIS framework model using multiple CNN inputs, we use a *Blending* ensemble learning. Blending is an ensemble machine learning technique that combines predictions from multiple base models (Level-0 models) and builds an ensemble model (Level-1 model). Blending is called stacked generalization, and it has the advantage of shortening the optimization processing time for deep learning because the training procedure is simple. Blending trains a meta-model using predictions from the validation dataset in one holdout set (Single-split) [52].

We construct the DL datasets (training, validation, and test set) by performing two random splits with a 7:3 ratio. The first random split is performed on the motion dataset to make pre-training and test sets. The second random split is performed on the pre-training set to make the final training set and validation set.

1) BASE MODELS

EDIS framework uses a pre-trained CNN model as a base learner $\mathcal{L}_{base}$ in blending. The framework uses multiple base learners $\mathcal{L}^i_{base}$, where i = 1, 2, ..., k, and k is the total number of sliding windows.

$$\bar{h}_i = \mathcal{L}^i_{base}(x_i) \tag{9}$$

where x_i is a 2D matrix image on i^{th} sliding window. h_i is prediction results of i^{th} base learner $\mathcal{L}^i_{base}$, i.e., probabilities for *Normal* class and *Stroke* class.

In our observations on the sleep motion dataset, motion data with less than 25% percentile of MCR is not suitable for classification performance. Thus, we perform a conditional probability filter f on the prediction results of base learners, as shown in Algorithm 1. When the input data is less than predefined thresholds (less than 25% MCR of the training dataset, i.e., Threshold $\mathcal{T}$), the conditional probability function f modifies the prediction results.

$$\bar{h}^*_i = f(\bar{h}_i) \tag{10}$$

2) META MODEL

To aggregate prediction results from the base learner models, we use a softmax layer as a meta model. Meta model uses a filtered prediction result set $\bar{H}^*$ from base learners $\mathcal{L}_{base}$ to make new predictions.

Algorithm 1 Conditional Probability Function

Input: ω^L, ω^R, MCR_{train}, $M_{trained}$
Output: $\bar{H}$

```
I_matrix ⇐ image(ω^L · ω^R)
P ⇐ predictProb(M_trained, I_matrix)
T ⇐ percentile(MCR_train, 25)
MCR ⇐ element(ω^L + ω^R)/length(ω)
if MCR ≤ T then
    H̄* ← [1/2, 1/2] ▷ *Probability for [Class0, Class1]
else
    H̄* ← P
    N ⇐ n
end if
```

$$\bar{H}^* = \mathcal{L}_{base}(X) \tag{11}$$

A meta learner $\mathcal{L}_{meta}$ makes the final predictions a_m for each class, where m = 0, 1, tacking into account both prediction results of base learners h^*_i and weights w_i, as follows.

$$\mathcal{L}_{meta}(C = m|\bar{H}^*) = a_m = \sum_i h^*_i w_i \tag{12}$$

where a_m is the outcome of a meta learner $\mathcal{L}$. C denotes a set of classes: normal(0) and stroke(1).

Softmax is used to make predicted probabilities as the standard approach for classification problems using deep learning. Softmax $\mathcal{S}$ has 2 nodes denoted by p_m, where m = 0, 1.

$$\mathcal{S} = p_m = \frac{exp(a_m)}{\sum_j^c exp(a_j)} \tag{13}$$

where c is number of class. p_m specifies a discrete probability distribution, therefore, $\sum_m p_m = 1$.

For optimizing the softmax, we use the cross entropy function as the loss function. The cross entropy function E is given by:

$$E = \mathcal{D}(\mathcal{S}, \mathcal{D}) = -\sum_j^c \mathcal{D}_i log(\mathcal{S}_j) \tag{14}$$

where $\mathcal{D}$ is target matrix and $\mathcal{S}$ is output matrix from the softmax. For implementation to optimize the softmax, we use a function of *trainSoftmaxLayer* in Matlab.

After optimizing the softmax, we can use the trained weights w to make the final predictions (probabilities) p_m for each class m in the meta model $\mathcal{L}_{meta}$. The final predicted class $\bar{y}$ is classified by using the argmax function.

$$\bar{y} = \underset{m}{\text{argmax}}\, p_m = \underset{m}{\text{argmax}}\, a_m \tag{15}$$

IV. EVALUATION

This section evaluates algorithm performance in terms of three considerations: classification performance, computation performance, and detection time.

A. EVALUATION SETUP

In this section, we first describe how to set up the evaluation settings in terms of classification performance, computational performance, and detection time.

A confusion matrix is widely used for evaluation of classification performance. The confusion matrix can visualize the performance of an algorithm. Based on the values in the table, the table generates four basic performance metrics: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Using these four basic performance metrics, the table also provides more specific performance metrics: Precision, Sensitivity, Specificity, Accuracy, and F1-score. A detailed description of the performance metrics follows.

Precision (Positive Predictive Value): Measures the number of correct positive predictions out of the total number of positive predictions.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{16}$$

Sensitivity (True Positive Rate): Measures the number of correct positive predictions out of the total number of positives.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{17}$$

Specificity (True Negative Rate): Measures the number of correct negative predictions out of the total number of negatives.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{18}$$

Accuracy: Measures the number of all correct predictions out of the total number of the dataset.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{19}$$

F1-score: A weighted harmonic mean of precision and sensitivity

$$\begin{aligned}\text{F1-score} &= \frac{2}{\frac{1}{\text{Sensitivity}} + \frac{1}{\text{Precision}}} \\ &= 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}}\end{aligned} \tag{20}$$

Because the sleep duration of each individual is slightly different, it was difficult to collect the same number and sleep time for the normal and patient groups. That is, our sleep dataset is an imbalanced dataset in which the classes are not evenly distributed. The appropriate performance metric is vital for algorithm evaluation in the imbalanced dataset. We evaluate classification performance using F1-score and select the best algorithm. F1-score is widely used as an important metric for evaluating model performances for an imbalanced dataset [53], [54]. We also assess the classification performance with other performance metrics.

1) CLASSIFICATION PERFORMANCE METRICS

DL models classify two labels in our dataset: *Stroke (True)* and *Normal (False)* classes. This classification problem is a binary classification problem (boolean-typed labels). We evaluates the performance of detecting *Stroke (True)* in the classification performance evaluation.

2) COMPUTATION PERFORMANCE METRICS

For computational performance evaluation, we consider two computational metrics [55]: model complexity and inference time. Model complexity is calculated by counting the total of learnable parameters in DL models. Model complexity is proportional to the total memory usage of the GPU. Thus, a high-complexity DL model (with many parameters) requires GPUs with large memory sizes.

Inference time is calculated using the average time per image inference time over 100 runs. Inference time is critical to real-time computing performance. We evaluate the inference time with two environmental conditions: CPU and GPU. We measure inference time using the same batch size (64) and run on using a desktop. The desktop is equipped with an Intel Core i9-10900F, CPU @ 2.80GHZ, 64GB DDR4 RAM 3200MHZ, and NVIDIA Geforce RT 3090. The operating system is Window 10.

3) SIMULATION STUDY FOR DETECTION TIME

When applying the developed DL algorithm to sleep data in real-time, it is crucial to check how quickly it performs in-sleep stroke detection. Unfortunately, the DL algorithm cannot be applied to patients who are likely to have a stroke. Thus, we simulate an event of in-sleep stroke using the collected sleep dataset. For example, we randomly select a test dataset (ω minutes) from normal people and stroke patients, respectively. Then, two test datasets are concatenated to generate one in-sleep stroke event (2ω minutes). We check how quickly DL models detect the stroke by applying the trained DL algorithm to the newly created in-sleep stroke event data, i.e., simulation data. This simulation test is performed 100 times. The sliding window moves every minute until the ω minutes and records the first time of five consecutive detections (5 minutes threshold) on the simulation data. The 5-minute time threshold is determined heuristically. If DL models does not detect until ω minutes, the detection time is marked as ω minutes.

B. CLASSIFICATION PERFORMANCE EVALUATION

We first evaluate the classification performance of DL models according to the sliding window time. For this evaluation, we perform n-repeated hold-out cross-validation, where n is 30. We use the F1-score metric for model evaluation and selection.

Fig. 5 is a bar graph with error bar (95% confidence interval) showing the classification performance of ten DL models considering six sliding window times. Overall, the classification performance improves as the sliding window

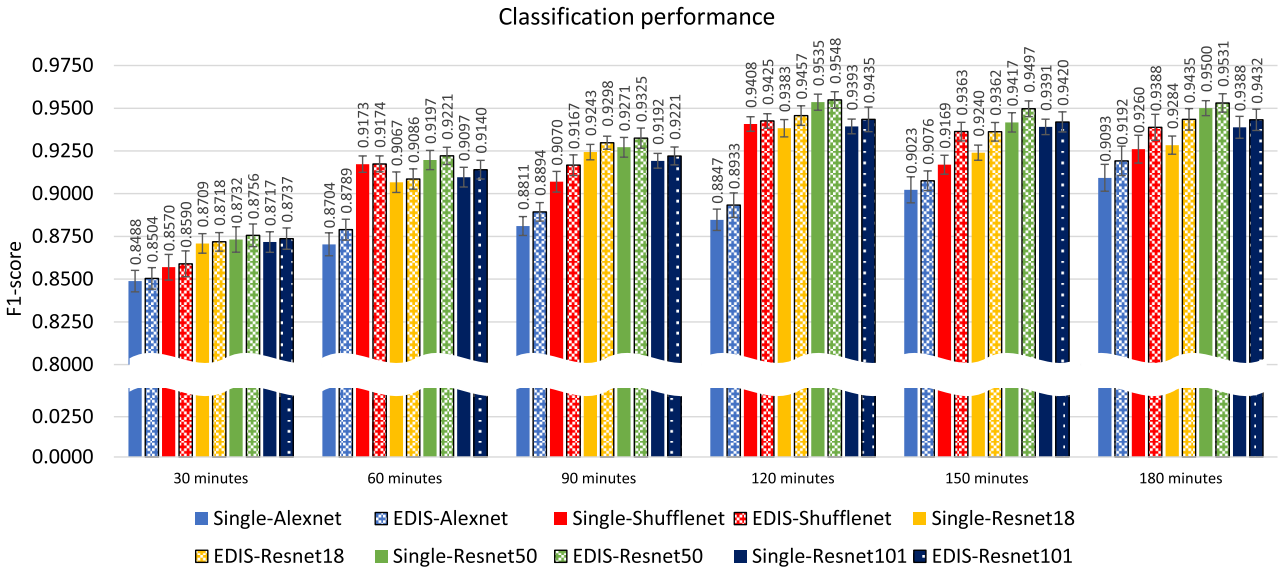

FIGURE 5. In-sleep stroke detection performance evaluation depending on sliding window size.

TABLE 1. Classification performance evaluation.

Model	Selected SW	F1-score	Accuracy	Precision	Sensitivity	Specificity
Single-Alexnet	180 minutes	0.909 (0.901, 0.917)	0.872 (0.862, 0.883)	0.913 (0.903, 0.923)	0.907 (0.892, 0.923)	0.787 (0.759, 0.816)
EDIS-Alexnet	180 minutes	0.919 (0.911, 0.928)	0.885 (0.873, 0.897)	0.918 (0.907, 0.929)	0.922 (0.908, 0.935)	0.797 (0.768, 0.826)
Single-Shufflenet	120 minutes	0.941 (0.937, 0.945)	0.920 (0.914, 0.926)	0.924 (0.915, 0.932)	0.959 (0.952, 0.966)	0.845 (0.825, 0.865)
EDIS-Shufflenet	120 minutes	0.943 (0.938, 0.947)	0.923 (0.917, 0.929)	0.930 (0.921, 0.938)	0.956 (0.950, 0.962)	0.857 (0.838, 0.876)
Single-Resnet18	120 minutes	0.938 (0.933, 0.943)	0.918 (0.911, 0.924)	0.928 (0.922, 0.935)	0.949 (0.941, 0.957)	0.857 (0.842, 0.872)
EDIS-Resnet18	120 minutes	0.946 (0.940, 0.951)	0.927 (0.920, 0.935)	0.934 (0.927, 0.941)	0.959 (0.949, 0.968)	0.866 (0.852, 0.881)
Single-Resnet50	120 minutes	0.954 (0.949, 0.958)	0.937 (0.931, 0.944)	0.935 (0.928, 0.942)	0.973 (0.965, 0.981)	0.867 (0.852, 0.883)
EDIS-Resnet50	120 minutes	**0.955 (0.950, 0.960)**	0.939 (0.933, 0.946)	0.943 (0.936, 0.949)	0.968 (0.960, 0.975)	0.885 (0.871, 0.898)
Single-Resnet101	120 minutes	0.939 (0.935, 0.944)	0.918 (0.912, 0.924)	0.917 (0.910, 0.924)	0.963 (0.956, 0.971)	0.829 (0.812, 0.846)
EDIS-Resnet101	120 minutes	0.943 (0.936, 0.951)	0.924 (0.915, 0.933)	0.929 (0.922, 0.936)	0.959 (0.947, 0.970)	0.857 (0.843, 0.872)

Mean (95% Confidential Interval)

time increases. More precisely, the classification performance peaks at 120 minutes and then degrades, indicating that 120 minutes is the optimal window time for ten DL models. The other two models (Single-Alexnet and EDIS-Alexnet) have the best classification performance at 180 minutes.

When the EDIS framework is applied to the five pre-trained CNN models, the classification performance is improved in all five models. The results show that blending ensemble learning enhances the classification performance by aggregating the prediction results of base models.

In addition, the performance improvement at 30 minutes in EDIS framework is due to a softmax layer. In the EDIS framework, a softmax layer is added to combine the prediction results of base models, and the trained softmax layer eventually leads to better performance.

By examining all sliding window sizes, we can select the optimal sliding window size for each model that shows the best classification performance. Among the ten DL models, EDIS-Resnet50 shows the highest classification performance with F1-score of 0.955 (0.950, 0.960), while Single-Alexnet shows the lowest performance with F1-score of 0.909 (0.901, 0.917). The performance metric of Accuracy also shows the same trend as the F1-score. The other performance metrics, such as Precision, Sensitivity, and Specificity, are also shown in Table 1. As the final model, we select the EDIS-Resnet50 with a 120-minute sliding window.

C. COMPUTATION PERFORMANCE EVALUATION

Computational performance is a crucial factor to consider when deploying a developed AI model on a target GPU embedded platform for real-time systems. We compare and evaluate the computation performance of ten DL models in terms of *Model complexity* and *Inference time*. Table 2 examines the number of learnable parameters of ten DL models and CPU and GPU inference times, respectively.

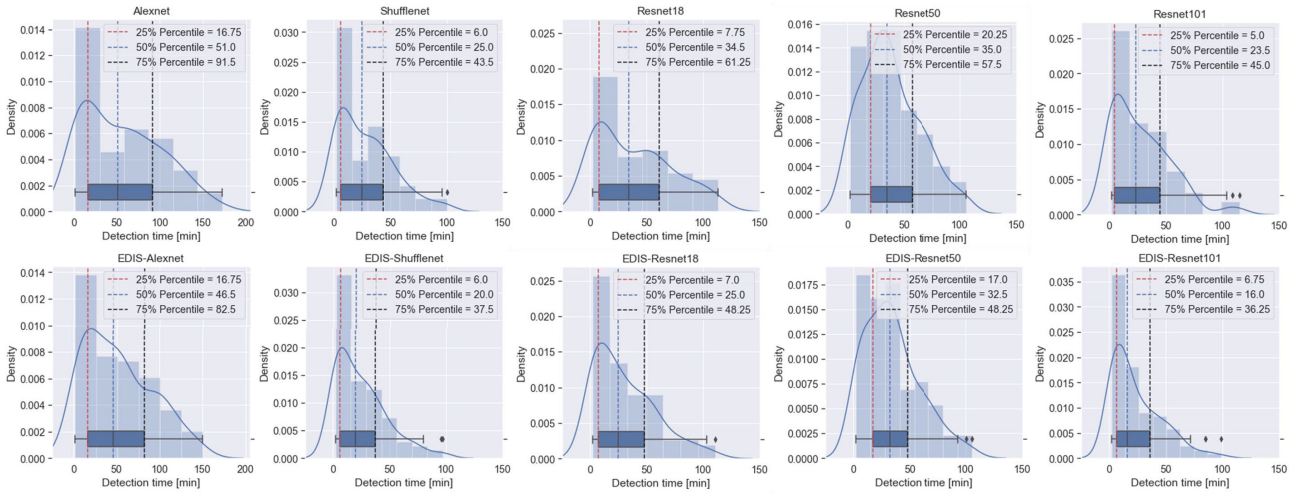

FIGURE 6. Histogram of detection time in DL models.

1) MODEL COMPLEXITY

Model complexity is related to the number of learnable parameters of the DL model. Since model complexity is proportional to the total memory usage of the GPU, this complexity is used to determine the GPU specifications required to run the developed model on an actual real-time embedded hardware platform.

We use five basic pre-trained CNN models: Alexnet, Shufflenet, Resnet18, Resnet50, and Resnet101. EDIS framework consists of homogeneous multi-DL models with different window sizes. Depending on the number of parameters of the selected pre-trained CNN model and the window size, the number of parameters in the EDIS framework increases proportionally.

The sliding window size showing the best performance for EDIS-Alexnet is 180 minutes. EDIS-framework-Alexnet consists of six duplicated Alexnet models. AlexNet shows relatively weak classification performance compared to other models, and the performance improvement over the ensemble-based architecture is up to 180 minutes. On the other hand, other models show that the effect of the architecture is maximized up to 120 minutes, and the performance decreases after that. EDIS framework for the four models (Shufflenet, Resnet18, Resnet50, and Resnet101) consists of four duplicated models each.

The final model chosen for the EDIS framework, EDIS-Resnet50, has 94,154,760 trainable parameters.

2) INFERENCE TIME

Inference time is measured by using CPU and GPU in a desktop environment (CPU: Intel i9-10900F, GPU: NVIDIA GeForce RTX 3090). We put a tic-tok function before and after the code that runs the prediction of the DL model. The prediction task is then repeated 100 times to measure the average elapsed time with CPU and GPU.

In general, the inference time of deep learning is shorter when using GPU than CPU. CNN models with relatively large parameter sizes (Alexnet, Resnet18, and Resnet101) have shorter inference times when using GPU than CPU. However, pre-trained CNN models with relatively small parameters, such as Shufflenet and Resnet18, show shorter inference times using CPU than GPU.

EDIS framework has more parameters than single CNN models because the network architecture consists of several homogeneous CNN models.

The inference time of the EDIS framework is more affected by the parameter size of a single CNN model than the overall parameter size of the framework. For example, EDIS-Resnet18 has relatively large parameters but shows that CPU inference time is shorter than GPU inference time. Because the inference time of single Resnet18, one of the basic CNN models on the EDIS framework, greatly influences the overall inference time of the framework.

The final model chosen for the EDIS framework, EDIS-Resnet50, shows that the inference time with GPU is 0.0807 milliseconds, while the time with CPU is 0.0923 milliseconds.

D. DETECTION TIME EVALUATION

Detection time is important performance metric because it tells how quickly a real-time monitoring system can detect a stroke. Because the developed model is difficult to apply and test at the moment of an actual stroke, we simulate a stroke event using collected sleep data from healthy people and stroke patients. A stroke event is generated by randomly selecting test datasets of healthy people and stroke patients, resulting in a total of 100 simulated data. Every minute, a sliding window shifts to determine how rapidly DL models can detect a stroke from its onset.

Fig. 6 shows a histogram plot of the simulation results for detecting a stroke during sleep in real time. Simulation

TABLE 2. Computation performance evaluation.

Model	# Parameters [number]	Time withCPU [millisecond]	Time withGPU [millisecond]
Single-Alexnet	56,868,224	0.0156	0.0073
EDIS-Alexnet	341,258,508	0.1076	0.0412
Single-Shufflenet	862,802	0.0141	0.0179
EDIS-Shufflenet	3,451,208	0.0612	0.0784
Single-Resnet18	11,182,338	0.0116	0.0123
EDIS-Resnet18	44,729,352	0.0515	0.0569
Single-Resnet50	23,538,690	0.0216	0.0199
EDIS-Resnet50	94,154,760	0.0923	0.0807
Single-Resnet101	42,556,930	0.0379	0.0344
EDIS-Resnet101	170,227,720	0.1570	0.1554

CPU: Intel i9-10900F, GPU: NVIDIA GeForce RTX 3090

results show that the EDIS framework shortens the detection time on five basic pre-trained CNN models. The detection time is reduced by 1 to 9 minutes at the 50% percentile (median), and 2.2 to 12.65 minutes at the 95% percentile, respectively. The detection times of the selected DL model (EDIS-Resnet50) are 36 minutes at the 55% percentile (median) and 81.1 minutes at the 95% percentile, respectively.

Note that ten DL models show that 100% detection rate in all simulation results within 180 minutes (the deadline). This simulation study for detection time checks whether EDIS framework helps reduce the detection time on single pre-trained CNN models. We select the most accurate detection model for the final DL model in the EDIS framework than the fastest detection model.

V. CONCLUSION

We propose a system that utilizes two wristbands to mitigate the risk of WUS and ensure timely stroke treatment. The core of this system is the development of an EDIS framework, which enables accurate and rapid detection of in-sleep strokes.

The primary objectives of the EDIS framework are twofold: enhancing detection performance and minimizing detection time. To achieve these goals, we introduce a novel network architecture known as the EDIS framework. This architecture comprises several homogeneous CNN models with varying window sizes. The framework combines the prediction results from these models using the *Blending* method to generate a final prediction. By employing multi-image data and aligning windows of different sizes with the current time (aligned cascading windows), the network architecture improves classification accuracy and reduces detection time.

To classify the asymmetrical sleep motion patterns (hemiparesis) exhibited by stroke patients, we devised a method for generating two-dimensional matrix images from accelerometer data captured from both hands. For evaluating the performance of the proposed framework, we employ a total of ten DL models: five pre-trained CNN models and five enhanced models incorporating the EDIS framework. Each model utilizes the optimal window size that yields the highest classification accuracy.

The EDIS framework, due to its multiple CNNs, possesses a greater number of parameters and requires a certain amount of inference time compared to pre-trained CNN models. Nonetheless, extensive evaluation results demonstrate that the EDIS framework improves classification performance and shortens detection time in all five pre-trained models.

This study primarily focuses on the development of a deep learning algorithm for in-sleep stroke detection in the domain of wearable computing. In future research, we intend to explore optimization challenges pertaining to hardware and software embedded systems, including reliable real-time computing within limited hardware resources, model compression, and the reduction of false alarms during practical use. Additionally, since the EDIS framework uses 2D image-based classification, we plan to apply the framework to other applications, such as human Electrocardiogram (ECG) Classification: Cardiac Arrhythmia and Congestive Heart Failure.

Given that brain cells begin to deteriorate immediately after a stroke, every second counts in saving the life of a stroke victim. We envision that the EDIS framework plays an important role in saving lives affected by stroke.

REFERENCES

[1] S. L. Murphy, J. Xu, K. D. Kochanek, and E. Arias, "Mortality in the United States, 2017," Tech. Rep., 2018.

[2] S. S. Virani, A. Alonso, E. J. Benjamin, M. S. Bittencourt, C. W. Callaway, A. P. Carson, A. M. Chamberlain, A. R. Chang, S. Cheng, and F. N. Delling, "Heart disease and stroke statistics-2020 update: A report from the American heart association," *Circulation*, vol. 141, no. 9, pp. 139–596, 2020.

[3] D. O. Kleindorfer, A. Towfighi, S. Chaturvedi, K. M. Cockroft, J. Gutierrez, D. Lombardi-Hill, H. Kamel, W. N. Kernan, S. J. Kittner, E. C. Leira, O. Lennon, J. F. Meschia, T. N. Nguyen, P. M. Pollak, P. Santangeli, A. Z. Sharrief, S. C. Smith, T. N. Turan, and L. S. Williams, "2021 guideline for the prevention of stroke in patients with stroke and transient ischemic attack: A guideline from the American heart association/American stroke association," *Stroke*, vol. 52, no. 7, pp. 1–2, Jul. 2021.

[4] W. L. Baker, J. A. Colby, V. Tongbram, R. A. Talati, I. E. Silverman, C. M. White, J. Kluger, and C. I. Coleman, "Neurothrombectomy devices for treatment of acute ischemic stroke," Tech. Rep., 2011.

[5] D. Kleindorfer, C. J. Lindsell, C. J. Moomaw, K. Alwell, D. Woo, M. L. Flaherty, O. Adeoye, T. Zakaria, J. P. Broderick, and B. M. Kissela, "Which stroke symptoms prompt a 911 call? A population-based study," *Amer. J. Emergency Med.*, vol. 28, no. 5, pp. 607–612, Jun. 2010.

[6] D. Song, E. Tanaka, K. Lee, S. Sato, M. Koga, Y. D. Kim, K. Nagatsuka, K. Toyoda, and J. H. Heo, "Factors associated with early hospital arrival in patients with acute ischemic stroke," *J. Stroke*, vol. 17, no. 2, p. 159, 2015.

[7] M. I. Weintraub, "Thrombolysis (tissue plasminogen activator) in stroke: A medicolegal quagmire," *Stroke*, vol. 37, no. 7, pp. 1917–1922, Jul. 2006.

[8] W. Hacke, M. Kaste, E. Bluhmki, M. Brozman, A. Davalos, D. Guidetti, V. Larrue, K. R. Lees, Z. Medeghri, and T. Machnig, "Thrombolysis with alteplase 3 to 4.5 hours after acute ischemic stroke," *New England J. Med.*, vol. 359, no. 13, pp. 1317–1329, 2008.

[9] E.-J. Lee, S. J. Kim, J. Bae, E. J. Lee, O. D. Kwon, H.-Y. Jeong, Y. Kim, and H.-B. Jeong, "Impact of onset-to-door time on outcomes and factors associated with late hospital arrival in patients with acute ischemic stroke," *PLoS ONE*, vol. 16, no. 3, Mar. 2021, Art. no. e0247829.

[10] A. R. Seo, W. J. Lee, S. H. Woo, J. Moon, and D. Kim, "Pre-hospital delay in patients with acute stroke during the initial phase of the coronavirus disease 2019 outbreak," *J. Korean Med. Sci.*, vol. 37, no. 6, 2022, Art. no. 1158793.

[11] D. L. Rimmele and G. Thomalla, "Wake-up stroke: Clinical characteristics, imaging findings, and treatment option—An update," *Frontiers Neurol.*, vol. 5, p. 35, Mar. 2014.

[12] M. N. Rubin and K. M. Barrett, "What to do with wake-up stroke," *Neurohospitalist*, vol. 5, no. 3, pp. 161–172, Jul. 2015.

[13] A. Wouters, R. Lemmens, P. Dupont, and V. Thijs, "Wake-up stroke and stroke of unknown onset: A critical review," *Frontiers Neurol.*, vol. 5, p. 153, Aug. 2014.

[14] J. W. Dankbaar, H. P. Bienfait, C. van den Berg, E. Bennink, A. D. Horsch, T. van Seeters, I. C. van der Schaaf, L. J. Kappelle, and B. K. Velthuis, "Wake-up stroke versus stroke with known onset time: Clinical and multimodality CT imaging characteristics," *Cerebrovascular Diseases*, vol. 45, nos. 5–6, pp. 236–244, 2018.

[15] G. Thomalla, J. B. Fiebach, L. Østergaard, S. Pedraza, V. Thijs, N. Nighoghossian, P. Roy, K. W. Muir, M. Ebinger, B. Cheng, I. Galinovic, T.-H. Cho, J. Puig, F. Boutitie, C. Z. Simonsen, M. Endres, J. Fiehler, and C. Gerloff, "A multicenter, randomized, double-blind, placebo-controlled trial to test efficacy and safety of magnetic resonance imaging-based thrombolysis in wake-up stroke (WAKE-UP)," *Int. J. Stroke*, vol. 9, no. 6, pp. 829–836, Aug. 2014.

[16] W. J. Powers, A. A. Rabinstein, T. Ackerson, O. M. Adeoye, N. C. Bambakidis, K. Becker, J. Biller, M. Brown, B. M. Demaerschalk, B. Hoh, E. C. Jauch, C. S. Kidwell, T. M. Leslie-Mazwi, B. Ovbiagele, P. A. Scott, K. N. Sheth, A. M. Southerland, D. V. Summers, and D. L. Tirschwell, "Guidelines for the early management of patients with acute ischemic stroke: 2019 update to the 2018 guidelines for the early management of acute ischemic stroke: A guideline for healthcare professionals from the American heart association/American stroke association," *Stroke*, vol. 50, no. 12, pp. 344–418, Dec. 2019.

[17] S.-L. Liew, K. A. Garrison, K. L. Ito, P. Heydari, M. Sobhani, J. Werner, H. Damasio, C. J. Winstein, and L. Aziz-Zadeh, "Laterality of poststroke cortical motor activity during action observation is related to hemispheric dominance," *Neural Plasticity*, vol. 2018, pp. 1–14, May 2018.

[18] M. Marangon, K. Priftis, M. Fedeli, S. Masiero, P. Tonin, and F. Piccione, "Lateralization of motor cortex excitability in stroke patients during action observation: A TMS study," *BioMed Res. Int.*, vol. 2014, pp. 1–7, 2014.

[19] M. Ajcević, G. Furlanis, A. Miladinović, A. B. Stella, P. Caruso, M. Ukmar, M. A. Cova, M. Naccarato, A. Accardo, and P. Manganotti, "Early EEG alterations correlate with CTP hypoperfused volumes and neurological deficit: A wireless EEG study in hyper-acute ischemic stroke," *Ann. Biomed. Eng.*, vol. 49, pp. 2150–2158, Sep. 2021.

[20] A. A. Anastasi, O. Falzon, K. Camilleri, M. Vella, and R. Muscat, "Brain symmetry index in healthy and stroke patients for assessment and prognosis," *Stroke Res. Treatment*, vol. 2017, pp. 1–9, Jan. 2017.

[21] M. J. A. M. van Putten, "The revised brain symmetry index," *Clin. Neurophysiol.*, vol. 118, no. 11, pp. 2362–2367, Nov. 2007.

[22] G. D. Cascino and F. W. Sharbrough, "Chapter 9—Intraoperative electroencephalographic monitoring during carotid endarterectomy and cardiac surgery," in *Aminoff's Electrodiagnosis in Clinical Neurology*, M. J. Aminoff, Ed., 6th ed. London: W. B. Saunders, 2012, pp. 207–218.

[23] C. M. Wilkinson, J. I. Burrell, J. W. P. Kuziek, S. Thirunavukkarasu, B. H. Buck, and K. E. Mathewson, "Predicting stroke severity with a 3-min recording from the muse portable EEG system for rapid diagnosis of stroke," *Sci. Rep.*, vol. 10, no. 1, pp. 1–11, Oct. 2020.

[24] K. B. Walsh, "Non-invasive sensor technology for prehospital stroke diagnosis: Current status and future directions," *Int. J. Stroke*, vol. 14, no. 6, pp. 592–602, Aug. 2019.

[25] G. J. Kim, A. Parnandi, S. Eva, and H. Schambra, "The use of wearable sensors to assess and treat the upper extremity after stroke: A scoping review," *Disability Rehabil.*, vol. 44, no. 20, pp. 6119–6138, 2021.

[26] M. Rabuffetti, P. Meriggi, C. Pagliari, P. Bartolomeo, and M. Ferrarin, "Differential actigraphy for monitoring asymmetry in upper limb motor activities," *Physiological Meas.*, vol. 37, no. 10, pp. 1798–1812, Oct. 2016.

[27] C. Iacovelli, P. Caliandro, M. Rabuffetti, L. Padua, C. Simbolotti, G. Reale, M. Ferrarin, and P. M. Rossini, "Actigraphic measurement of the upper limbs movements in acute stroke patients," *J. NeuroEng. Rehabil.*, vol. 16, no. 1, pp. 1–10, Dec. 2019.

[28] G. Reale, S. Giovannini, C. Iacovelli, S. F. Castiglia, P. Picerno, A. Zauli, M. Rabuffetti, M. Ferrarin, G. Maccauro, and P. Caliandro, "Actigraphic measurement of the upper limbs for the prediction of ischemic stroke prognosis: An observational study," *Sensors*, vol. 21, no. 7, p. 2479, Apr. 2021.

[29] S. Datta, C. K. Karmakar, A. S. Rao, B. Yan, and M. Palaniswami, "Automated scoring of hemiparesis in acute stroke from measures of upper limb co-ordination using wearable accelerometry," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 4, pp. 805–816, Apr. 2020.

[30] S. Datta, C. K. Karmakar, B. Yan, and M. Palaniswami, "Novel measures of similarity and asymmetry in upper limb activities for identifying hemiparetic severity in stroke survivors," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 6, pp. 1964–1974, Jun. 2021.

[31] S. Jeon, T. Park, Y. S. Lee, S. H. Son, H. Lee, and Y. Eun, "RISK-sleep: Real-time stroke early detection system during sleep using wristbands," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 4333–4339.

[32] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, Apr. 2014.

[33] G. Wang, Q. Li, L. Wang, W. Wang, M. Wu, and T. Liu, "Impact of sliding window length in indoor human motion modes and pose pattern recognition based on smartphone sensors," *Sensors*, vol. 18, no. 6, p. 1965, Jun. 2018.

[34] J. Qi, P. Yang, A. Waraich, Z. Deng, Y. Zhao, and Y. Yang, "Examining sensor-based physical activity recognition and monitoring for healthcare using Internet of Things: A systematic review," *J. Biomed. Informat.*, vol. 87, pp. 138–153, Nov. 2018.

[35] A. Sadeh, "The role and validity of actigraphy in sleep medicine: An update," *Sleep Med. Rev.*, vol. 15, no. 4, pp. 259–267, Aug. 2011.

[36] Y. J. Jeon and S. J. Kang, "Wearable sleepcare kit: Analysis and prevention of sleep apnea symptoms in real-time," *IEEE Access*, vol. 7, pp. 60634–60649, 2019.

[37] S. Jeon, T. Park, A. Paul, Y.-S. Lee, and S. H. Son, "A wearable sleep position tracking system based on dynamic state transition framework," *IEEE Access*, vol. 7, pp. 135742–135756, 2019.

[38] S. Lüdtke, W. Hermann, T. Kirste, H. Benes, and S. Teipel, "An algorithm for actigraphy-based sleep/wake scoring: Comparison with polysomnography," *Clin. Neurophysiol.*, vol. 132, no. 1, pp. 137–145, Jan. 2021.

[39] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, "Deep learning vs. traditional computer vision," in *Proc. Sci. Inf. Conf.* Cham, Switzerland: Springer, 2019, pp. 128–144.

[40] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 1307–1310.

[41] S. Jeon, J. Son, M. Park, B. S. Ko, and S. H. Son, "Driving-PASS: A driving performance assessment system for stroke drivers using deep features," *IEEE Access*, vol. 9, pp. 21627–21641, 2021.

[42] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.

[43] A. Murad and J.-Y. Pyun, "Deep recurrent neural networks for human activity recognition," *Sensors*, vol. 17, no. 11, p. 2556, Nov. 2017.

[44] H. Wang, J. Zhao, J. Li, L. Tian, P. Tu, T. Cao, Y. An, K. Wang, and S. Li, "Wearable sensor-based human activity recognition using hybrid deep learning techniques," *Secur. Commun. Netw.*, vol. 2020, pp. 1–12, Jul. 2020.

[45] F. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, Jan. 2016.

[46] K. M. Steele, C. Papazian, and H. A. Feldner, "Muscle activity after stroke: Perspectives on deploying surface electromyography in acute care," *Frontiers Neurol.*, vol. 11, Sep. 2020, Art. no. 576757.

[47] N. Ciesla, V. Dinglas, E. Fan, M. Kho, J. Kuramoto, and D. Needham, "Manual muscle testing: A method of measuring extremity muscle strength applied to critically ill patients," *J. Visualized Exp.*, no. 50, p. e2632, Apr. 2011.

[48] J. J. Fraigne, K. P. Grace, R. L. Horner, and J. Peever, "Mechanisms of REM sleep in health and disease," *Current Opinion Pulmonary Med.*, vol. 20, no. 6, pp. 527–532, Nov. 2014.

[49] R. Ke, W. Li, Z. Cui, and Y. Wang, "Two-stream multi-channel convolutional neural network for multi-lane traffic speed prediction considering traffic volume impact," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2674, no. 4, pp. 459–470, Apr. 2020.
[50] H. Ye, Z. Wu, R.-W. Zhao, X. Wang, Y.-G. Jiang, and X. Xue, "Evaluating two-stream CNN for video classification," in *Proc. 5th ACM Int. Conf. Multimedia Retr.*, Jun. 2015, pp. 435–442.
[51] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.
[52] T. Wu, W. Zhang, X. Jiao, W. Guo, and Y. A. Hamoud, "Evaluation of stacking and blending ensemble learning methods for estimating daily reference evapotranspiration," *Comput. Electron. Agricult.*, vol. 184, May 2021, Art. no. 106039.
[53] S. Wu, W.-C. Yau, T.-S. Ong, and S.-C. Chong, "Integrated churn prediction and customer segmentation framework for Telco business," *IEEE Access*, vol. 9, pp. 62118–62136, 2021.
[54] S. Bej, N. Davtyan, M. Wolfien, M. Nassar, and O. Wolkenhauer, "LoRAS: An oversampling approach for imbalanced datasets," *Mach. Learn.*, vol. 110, no. 2, pp. 279–301, Feb. 2021.
[55] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64270–64277, 2018.

SANGHOON JEON (Member, IEEE) received the B.S. degree from the School of Electronics Engineering, Kyoungpook National University, Daegu, South Korea, in 2012, and the M.S. degree from the Department of Information and Communication Engineering, DGIST, Daegu, in 2014. He is currently a Research Assistant Professor with the Department of Emergency Medicine, College of Medicine, Hanyang University. His research interests include wearable computing, human–computer interface, healthcare and the IoT applications, embedded systems, and Cyber-physical systems.

YANG-SOO LEE received the B.S. and M.S. degrees from the School of Medicine, Kyungpook National University, Daegu, South Korea, in 1986 and 1988, respectively, the Ph.D. degree in physiology from Yeungnam University, Daegu, in 2001, and the Rehabilitation Medicine degree from Kangdong Sacred Heart Hospital, in1994. Since 1994, he has been a Professor with the Department of Rehabilitation Medicine, School of Medicine, Kyungpook National University.

SANG HYUK SON (Life Fellow, IEEE) received the B.S. degree in electronics engineering from Seoul National University, the M.S. degree from KAIST, and the Ph.D. degree in computer science from the University of Maryland, College Park, MD, USA.

He was a Professor with the Computer Science Department, University of Virginia. He was a WCU Chair Professor with Sogang University. He was a Visiting Professor with KAIST; The City University of Hong Kong; École Centrale de Lille, France; Linköping University, Sweden; and the University of Skövde, Sweden. He is currently a Chair Professor with the Department of Electrical Engineering and Computer Science, DGIST. His research interests include cyber-physical systems, real-time and embedded systems, database and data services, and wireless sensor networks. He has authored or coauthored more than 340 papers and edited/authored four books in these areas. He is a IEEE Fellow. He is a member of the Korean Academy of Science and Technology and the National Academy of Engineering of Korea. He is a Founding Member of the ACM/IEEE CPS Week. He served as a member of the Steering Committee for the IEEE RTCSA and Cyber Physical Systems Week. He received the Outstanding Contribution Award from Cyber Physical Systems Week, in 2012. He served on the Editorial Board of *ACM Transactions on Cyber Physical Systems*, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and *Real-Time Systems Journal*.