ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

FallNeXt: A Deep Residual Model based on Multi-Branch Aggregation for Sensor-based Fall Detection

Sakorn Mekruksavanich[1] and Anuchit Jitpattanakul[2]

ABSTRACT

Falls are uncommon and pose a substantial health danger to adults and the elderly. These situations are a leading cause of severe injury. More harm could be averted if the faller could be located in time. The rising older population necessitates the rapid development of fall detection and prevention technologies. The burgeoning technology industry is focused on developing such technologies to improve the living conditions for the elderly in particular. A fall detection system monitors falls and provides an assistance notice to support mitigation of falls. This study proposes a sensor-based solution based on a deep learning network named FallNeXt to safeguard individual privacy and increase fall detection performance. This proposed network is a novel deep residual network that utilizes multi-branch aggregation to enhance fall detection capability. The detection effectiveness of this study was evaluated using three benchmark datasets for sensor-based fall detection: UpFall, SisFall, and UMAFall datasets. Compared to benchmark deep learning models on the three datasets, the experimental findings indicate that the proposed FallNeXt network scored the most significant overall accuracy and F1-score, with 96.16% and 99.12%, respectively. The benefit of the FallNeXt model's small but highly effective size for fall detection is its portability.

Article information:

Keywords: Fall Detection, Deep Residual Network, Wearable Sensors, Multi-branch Aggregation

Article history:
Received: March 10, 2022
Revised: April 23, 2022
Accepted: August 6, 2022
Published: September 3, 2022 (Online)

DOI: 10.37936/ecti-cit.2022164.248156

1. INTRODUCTION

The World Health Organization (WHO) reports that falls are a leading cause of injury in the elderly [1]. Additionally, medical evaluations have shown that treatment of fall-related injuries is significantly reliant on reaction and emergency time [2]. Consequently, it is critical for the elderly to preserve their well-being by immediately contacting the authorities and a physician if they fall.

Falls are caused by many factors [3]. They may occur as a result of an unpredicted health event, such as a heart condition, or as a result of impaired sensory capabilities, such as vision or cognition, or even due to long-term diseases that effect mobility, such as arthritis or Parkinson's disease. In any case, it is critical that an individual who falls gets treatment as soon as possible. Prompt treatment after a fall enhances a patient's post-fall living conditions. By notifying healthcare personnel, fall detection systems (FDS) play a critical role in ensuring timely treatment [4, 5].

Fall detection is a complex and challenging issue in human activity recognition (HAR) [6, 7]. The purpose of a FDS is to monitor a person's activity and identify falls to alert healthcare professionals or other caregivers. For example, fall detection systems are essential for older individuals with cognitive impairments who may be unable to stand for extended periods after a fall, leading to pressure sores and other complications [8].

Over the past decade, there has been a surge of studies on FDS, which may be classified into vision-based FDS and wearable-based FDS [9]. The disadvantage of vision-based FDS is that they are limited to confined spaces, such as a room or a care facil-

[1]The author is with the Department of Computer Engineering, School of Information and Communication Technology, University of Phayao, Phayao 56000, Thailand, E-mail : sakorn.me@up.ac.th
[2]The author is with the Department of Mathematics, Faculty of Applied Science, King Mongkut's University of Technology North Bangkok, Bangkok 10800, Thailand, E-mail : anuchit.j@sci.kmutnb.ac.th
[2]Intelligent and Nonlinear Dynamic Innovations Research Center, Science and Technology Research Institute, King Mongkut's University of Technology North Bangkok, Bangkok 10800, Thailand, E-mail : anuchit.j@sci.kmutnb.ac.th

ity. This is because they are expensive to build and maintain and because a permanent installation can restrict the individual's mobility. Wearable FDS uses motion sensors such as accelerometers, gyroscopes, and magnetometers, as well as pressure sensors and a variety of health monitoring sensors [10, 11]. The data collected by these sensors then used to ascertain whether or not a fall occurred. Wearable FDS typically make use of a large number of units connected to the body of the patient to record the individual's movement patterns more accurately.

To construct an effective model of sensor-based FDS, the researchers have combined a great many machine learning and deep learning techniques since these learning-based strategies can discover nonlinear patterns in motion or analyze fall information to recognize the occurrence.

To solve this issue, this article proposes a hybrid deep-learning-based strategy in which models are concurrently trained to identify fundamental and transitory human actions. Novel characteristics include implementing several deep learning models to better differentiate recognition findings and incorporate transitional activities to provide an effective activity identification strategy. The following is a summary of this work's primary contributions:

- A proposed deep residual network called FallNeXt is built on multi-branch aggregation and optimized for fall detection.
- Various experiments were conducted to evaluate the proposed network using three benchmark FDS datasets: SisFall, UpFall, and UMAFall.
- The results demonstrate that the proposed deep learning network's effectiveness for FDS has surpassed comparable approaches.

The remainder of this article is structured as follows: Section 2 summarizes the related literature. Section 3 details the FDS approach. Section 4 summarizes the experimental findings. The findings are discussed in Section 5. Lastly, Section 6 summarizes our study results and makes recommendations for future exploration.

2. RELATED WORK

Over the past decade, many of the studies in the field of FDS have focused on different methodologies utilizing machine learning and deep learning techniques. These studies can be classified into two broad categories: vision-based FDS and wearable-based FDS.

2.1 Vision-based Fall Detection Systems

Vision-based FDS acquires motion data by observing devices and extracting a person's body appearance inclination [12] or human bone annotations from received video or image data to determine if a fall has occurred. Agrawal et al. [13] employed experience subtraction to locate foreground items and selected the fall by computing the difference between the top and middle centers of the rectangle encompassing the human body. The participant is not required to wear any additional equipment. Nevertheless, it is simple to shut off and inevitably invades people's personal space. To overcome this challenge, Kong et al. [14] employed a depth camera to capture the skeletal photos of a human standing or falling. He encrypted the images and detected the fall by using Fast Fourier Transformation.

2.2 Wearable-based Fall Detection Systems

Wearable sensors have gained increasing interest as a result of their low cost [15]. The calf, spine, head, pelvis, and foot are the most often employed sites for wearable sensors to get 3D acceleration at various places and 3D rotation angular velocity with a gyroscope. Shahiduzzaman et al. [16] integrated wearable cameras, accelerometers, and gyroscope sensors into innovative headwear and processed data from multisensor cooperation on the edge. Mousavi et al. [17] suggested a technique for detecting falls using cellphones and acceleration signals by employing smartphone sensors and providing the user's location, with a 96.33% accuracy rate. Desai et al. [18] identified human falls using a modest 32-bit microcontroller embedded in a wearable waistband. Threshold approaches and machine classification algorithms are often employed to determine falls in wearable electronics [19].

Additionally, other studies have examined FDS using alternative sensors, such as environmental sensors. Usually, falls are detected by ambient sensors by gathering infrared [20], radar [21], or other signals from the scene sensor. Yang et al. [22] collected data using radars in three fixed sites and utilized the time-frequency distribution and range-time intensity as intake data for feature extraction to determine falls. While it does not violate one's privacy, it has a high cost, is vulnerable to noise, and has limited detection coverage.

2.3 Deep Learning Approaches for Fall Detection Systems

Machine learning techniques are broadly classified into two categories: traditional pattern identification and classification and identification based on deep learning [3,23]. Traditional recognition techniques (such as the support vector machine, the k-nearest neighbor algorithm, and others) all depend on human feature extraction. As a result, investigators' fall detection criteria must be increased. At first, it is vital to determine which human body elements are engaged in the falling procedure. Secondly, it is critical to analyze how these characteristics are separated from activity of daily living (ADL) such as seating and leaping since this will significantly slow down the

feature extraction process. Classification and identification algorithms based on deep learning are used increasingly more often utilized in fall detection approaches, because they can extract feature details directly. As a result of this benefit, deep learning strategies have grown in popularity in scientific society [24]. They have been utilized in various domains where they have accomplished performance levels comparable to human expertise [25-27]. In principle, deep learning algorithms for the data acquisition of wearable sensors entail preprocessing collected signals, extracting features from signal portions, and training a model employing features as input [28]. As a result, current wearable sensor studies in fall risk estimation mainly focus on technical feature optimization. The gathered characteristics are provided in several deep learning techniques used to indicate the recurrence of falls. Musci et al. [29] employed an LSTM-based fall detection scheme that used a long-time sequence as the source and then efficiently extracted temporal information.

3. THE PROPOSED DEEP LEARNING MODEL

As shown in Fig. 1, the framework used in this study is comprised of four main stages: data acquisition, data pre-processing, DL-based model training, and model evaluation.

First, we gathered FDS datasets, including wearable sensor data of ADLs and fall behaviors. Following analysis of the relevant literature, we have chosen three publicly accessible datasets to analyze in this research: UpFall, SisFall, and UMAFall. Three-dimensional accelerometer and three-dimensional gyroscope data are included in the sensor data. The sensor data was then denoised, normalized, and segmented using a sliding window method to create sample data for training and evaluating the deep learning algorithms. The output of the deep learning methods was examined using a 10-fold cross-validation methodology. Finally, four standard criteria were employed to assess and compare the trained models: accuracy, precision, recall, and F1-score. Each procedure is described in depth in the upcoming subsections.

3.1 Data Acquisition

According to their prevalence and scale, this study investigates fall detection using data from three FDS public datasets, namely UpFall, SisFall, and UMAFall. Each dataset is unique in terms of the number of ADLs and categories of falls, and the method used to gather sensor data. Many FDS investigations have made extensive use of these resources. Fig. 2 shows some accelerometer and gyroscope data samples from the three datasets, UpFall, SisFall, and UMAFall. The details of the three datasets are shown in Table 1.

The UpFall dataset [30] is the first FDS dataset we studied in this work. This dataset contains six activities of daily living (ADLs) (walking, standing, sitting, picking up an item, leaping, and lying down) as well as five distinct kinds of human falls (falling forward using hands, falling forward using knees, falling backward, falling sideways, and falling from sitting in an empty chair). The actions of 17 healthy young individuals were captured utilizing a multimodal method, including wearable sensors, ambient sensors, and vision equipment. For all activities, the sampling frequency is set to 100 Hz.

The SisFall dataset [31] is the second FDS dataset. This public dataset collects 3D accelerometer and 3D gyroscope data from subjects as they complete 19 ADLs and 15 fall actions. The wearable equipment was worn around the waist by 38 willing volunteers (23 young people and 15 older adults). The dataset was acquired from the sensors at a sampling rate of 200 Hz.

The UMAFall dataset [32] is the last FDS dataset used in this investigation. Three kinds of falls and twelve activities of daily living were gathered using a smartphone carried in the right thigh pocket and four wearable sensors worn on the ankle, left wrist, right wrist, and chest. Seventeen volunteers (seven females and ten males) completed a minimum of three sessions of each action in a household setting. Fall-backward, fall-forwards, and fall-lateral are the three distinct types of falls included in this dataset. Furthermore, the 12 ADLs are applauding, body bending (squatting), descending stairs, ascending stairs, lifting hands, hopping, light jogging, lying down on a bed, making a phone call, opening a door, sitting down, and walking. The sampling frequency for all activities in this dataset is harmonized at 20 Hz.

3.2 Data Pre-Processing

The raw sensor data contains system measurement noise or other unanticipated noise resulting from the individual's lively motions during the investigations. A noisy signal distorts the signal's usable information. As a result, it was critical to decrease the influence of the noise to recover relevant information from the signal for subsequent processing. The most frequently used filtering techniques are the mean filter, the low-pass filter, the Wavelet filter, and the Gaussian filter. Our investigation used an average smoothing filter for the accelerometer and gyroscope sensors to denoise the signals.

Then, as seen in Eq. (1), the raw sensor data is normalized to the range 0 to 1. This procedure assists in resolving the model learning issue by ensuring that all measured values are within a comparable degree. As a consequence, gradient descents could achieve a higher rate of convergence.

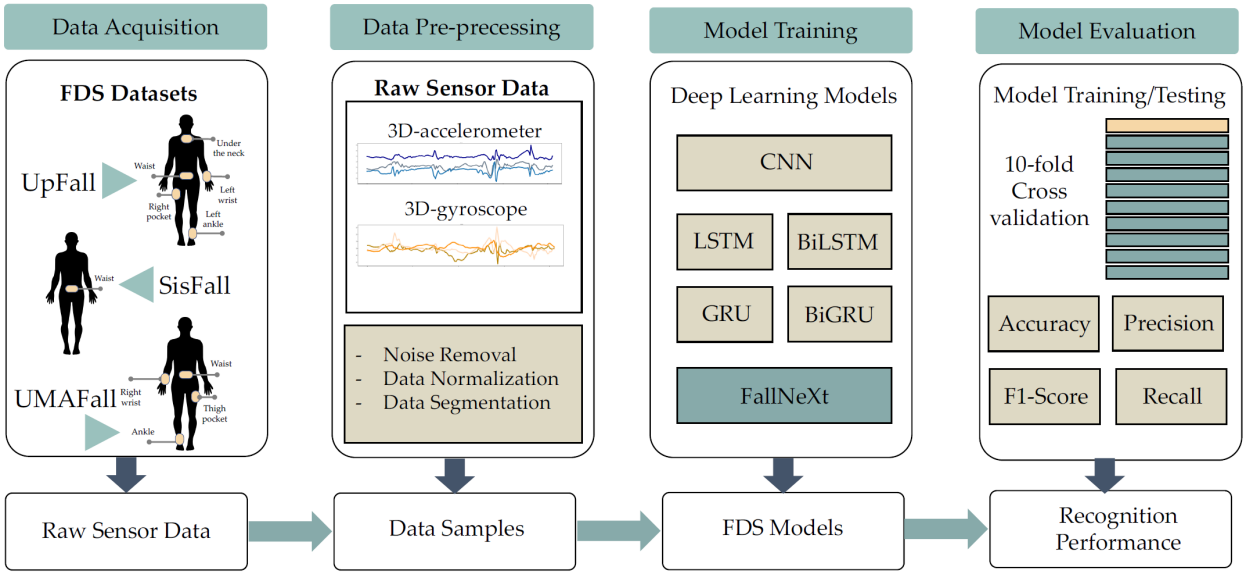

Fig.1: *The Framework for FDS used in this study..*

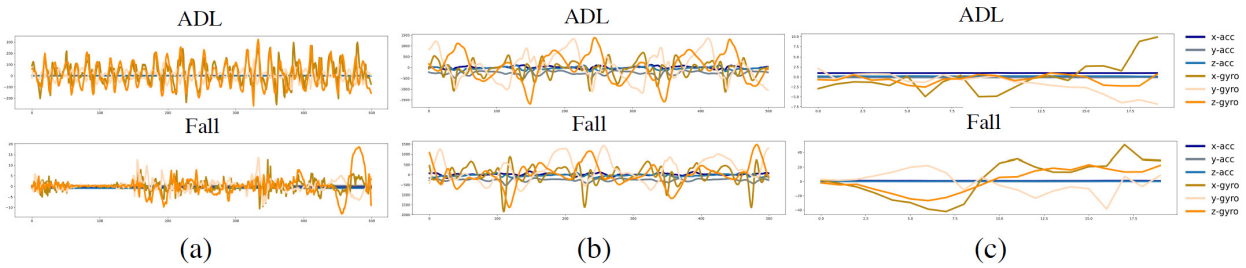

Fig.2: *Some samples of accelerometer and gyroscope data from the three datasets: (a) UpFall, (b) SisFall, and (c) UMAFall.*

Table 1: *Public benchmark datasets for FDS used in this work*

Dataset	Type of Sensors	Position	Subjects	Activities
UpFall [30]	1 accelerometer and 1 gyroscope	left wrist, under the neck, right pocket of pants, middle of waist (in the belt), and left ankle	17	5 types of fall and 6 ADLs
SisFall [31]	2 accelerometers and 1 gyroscope	waist	38	15 types of fall and 19 ADLs
UMAFall [32]	1 accelerometer and 1 gyroscope	thigh pocket, ankle, waist, and right wrist	72	3 types of fall and 8 ADLs

$$X_i^{norm} = \frac{X_i - x_i^{min}}{x_i^{max} - x_i^{min}}, \; i = 1, 2, ... \qquad (1)$$

X_i^{norm} denotes the normalized data, n denotes the number of channels, and x_i^{max} and x_i^{min} are the maximum and minimum values of the i-th channel, respectively.

Because of the high volume of signal data gathered by wearable sensors, it is hard to incorporate all the data into the FDS system at once. As a result, sliding window segmentation must be performed before feeding data into the system. The sliding window approach is widely employed in HAR for the identification of periodic activities (e.g., running, walking) and static actions (e.g., standing, sitting, and laying) [33]. The unprocessed sensor data was divided into fixed-length windows of 3 seconds. A fraction of the neighboring windows overlap to maximize the quantity of training data samples and prevent the loss of information during transition from one activity to the next. In this work, the overlapping proportion is 50%. The fixed-length sliding window technique used in this work is illustrated in Fig. 3.

3.3 Model Training

We implemented a multi-branch aggregation approach in this study influenced by ResNeXt [34]. Unlike InceptionNet [35], this approach utilizes the various-sized kernel feature maps rather than con-

Table 2: *The summary of hyperparameters for the FallNeXt network used in this work*

Stage	Hyperparameters		Values
Convolutional Block	Conv1D	Kernel Size	5
		Filters	64
	Batch Normalization		-
	Activation		ReLU
	Max Pooling		2
Multi Kernel Block x 5	Branch-1-1		
	Conv1D	Kernel Size	1
		Filters	10
	Conv1D	Kernel Size	3
		Filters	10
	Branch-1-2		
	Conv1D	Kernel Size	1
		Filters	10
	Conv1D	Kernel Size	5
		Filters	10
	Branch-1-3		
	Conv1D	Kernel Size	1
		Filters	10
	Conv1D	Kernel Size	7
		Filters	10
	Branch-1		
	Conv1D	Kernel Size	1
		Stride	1
		Filters	64
	Branch-2		
	Conv1D	Kernel Size	1
		Stride	1
		Filters	64
Classification Block	Global Average Pooling		-
	Flatten		-
	Dense		128
Training	Loss Function		Cross-entropy
	Optimizer		Adam
	Batch Size		64
	Number of Epochs		200

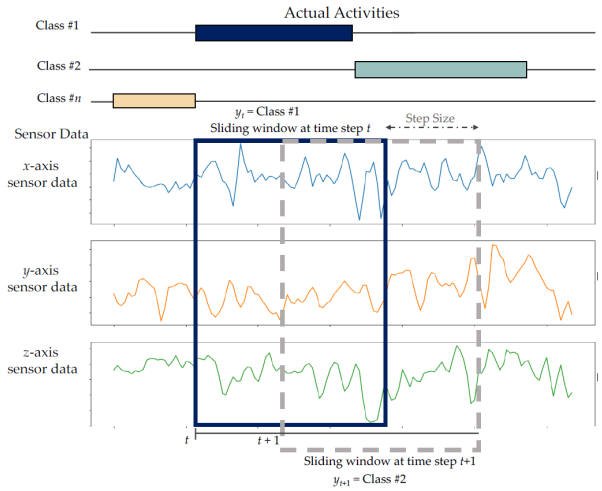

Fig.3: *Fixed-length sliding window technique used in this work.*

catenating them. This drastically reduces the model's parameter set, enabling adequate performance appropriate for edge and low latency operations. The summary of the hyperparameters of FallNeXt is given in Table 2. Fig. 4 illustrates FallNeXt in detail.

The FallNeXt model consists of four modules with convolutional kernels of varying sizes. Each MultiKernel (MK) component has three kernels of varying sizes: 1×3, 1×5, and 1×7. Additionally, 1×1 convolutions are performed to reduce the mode's overall complexity and parameter count before deploying these kernels. Fig. 5 illustrates the architecture of the MultiKernel component.

3.4 Model Evaluation

A member of the study population is assumed true positive (TP) if the classification action is correctly identified; false positive (FP) if the classification action is incorrectly identified; true negative (TN) if the classification action is correctly denied; and false negative (FN) if the classification action is incorrectly denied. The proposed methodology was assessed in this paper using the following four performance measures: accuracy, precision, recall, and F1-score. Formulae for these are shown in Eq. (2) through Eq. (5).

$$Precision(\%) = \frac{TP}{TP + FP} \times 100\% \qquad (2)$$

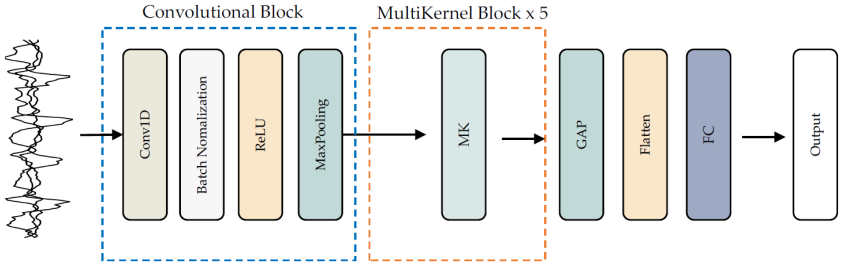

Fig.4: *The proposed FallNeXt model.*

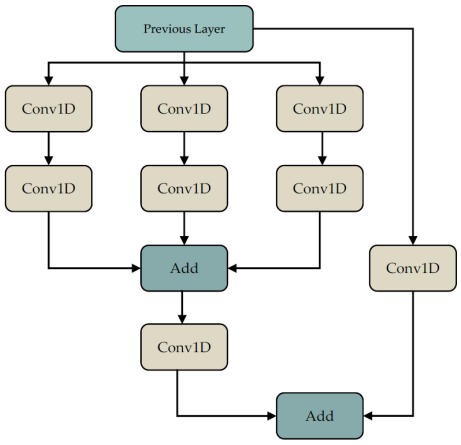

Fig.5: *Structure of a MultiKernel module.*

$$Recall(\%) = \frac{TP}{TP + FN} \times 100\% \tag{3}$$

$$F1 - score(\%) = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\% \tag{4}$$

$$Accuracy(\%) = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \tag{5}$$

4. EXPERIMENTAL RESULTS

This section discusses the experimental design used in this study and the observed outcomes for deep learning models, including the proposed FallNeXt model.

4.1 Experimental Setup

In this step, training and test data samples are partitioned for investigations, temporal windows from the signals were used to develop a model, and test data was used to evaluate the developed model. The standard procedure for splitting data into training and test sets is cross-validation. Numerous techniques, such as k-fold cross-validation, might segregate training data from testing data. This step evaluates the ability of the learning algorithm to generalize to new data. This step employed stratified 10-fold cross-validation inside of the smartwatch-based personal identification architecture. This validation technique divides the entire dataset into equal folds. Each repetition operates nine of these folds for training and one for testing. This step is conducted ten times, utilizing all data for both training and testing. Each fold has the same percentage of data from each participant when stratified.

The investigations in this research were done using the Google Colab Pro+ platform. The Tesla V100-SXM2-16GB graphics processing unit (GPU) was used to accelerate the training of the deep learning models. The proposed FallNeXt model was implemented in Python using the Tensorflow backend (version 3.9.1) and CUDA (version 8.0.6). The GPU accelerated deep learning model training and testing. The tests were conducted using the Python libraries listed below:

- Numpy and Pandas were used to read, manipulate, and analyze sensor data.
- Matplotlib and Seaborn were used to plot and present the results of knowledge discovery and model evaluation.
- Scikit-learn (Sklearn) was utilized in studies as a library for sampling and data production.
- TensorFlow, Keras, and TensorBoard were used to construct and train deep learning models.

4.2 Baseline Deep Learning Models

This research used five deep learning networks (CNN, LSTM, BiLSTM, GRU, and BiGRU) as baseline models to evaluate the proposed FallNeXt model's performance on fall detection.

The CNN model is a standard Convolutional Neural Network consisting of N convolutional layers activated by ReLU and a hidden dense layer. The CNN architecture employed in this study is shown in Fig. 6.

CNN hyperparameters include:

- The number of convolutional layers.
- The number of filters in each convolutional layer.

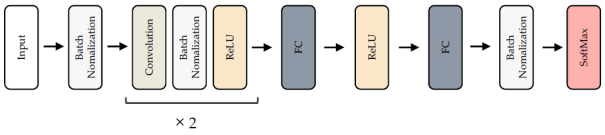

Fig.6: *The CNN architecture used in this work.*

- The number of neurons in the hidden dense layer.

We opted not to add pooling layers in this work since reducing the spatial size of the sequence is often unnecessary when there are sufficient convolutional layers.

The Long Short-Term Memory (LSTM) network architecture is a frequently used variation of the recurrent neural network (RNN) structure. Hochreiter and Schmidhuber [36] presented LSTM to address the vanishing gradient issue in long-term dependency learning. LSTM utilizes memory cells with three gates and parameters to describe long-range interdependence in temporal sequences. These gates regulate when states are modified, and previously hidden conditions are discarded, thereby governing the memory cells' overall functionality. The LSTM design employed in this investigation is shown in Fig. 7.

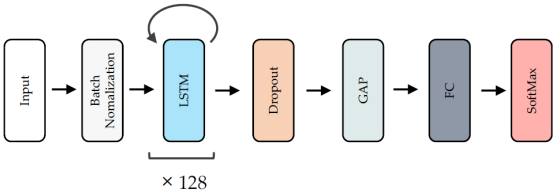

Fig.7: *The LSTM architecture used in this work.*

Schuster and Paliwal [37] proposed the Bidirectional LSTM (BiLSTM) to expand the level of knowledge provided in the LSTM network. The BiLSTM is connected in two directions to two hidden layers. This design will learn knowledge from both prior and future sequences concurrently. Fig. 8 illustrates the BiLSTM design used in this research.

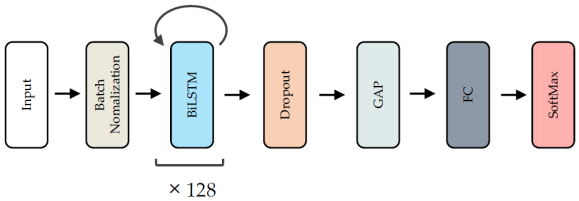

Fig.8: *The BiLSTM architecture used in this work.*

While LSTM is considered to be a reasonable alternative to RNNs' vanishing gradient challenge, the architecture's memory cells result in a rise in computational requirements. Cho et al. [38] published a paper in 2014 introducing the gate recurrent unit (GRU) network, a novel RNN-based model. The GRU is a simplified view of the LSTM that lacks a dedicated memory cell. A GRU network contains an update and reset gate that manages the level of adjustment of each hidden state. That is, it establishes which awareness should be transmitted and which should not. The GRU structure included in this study is depicted in Fig. 9.

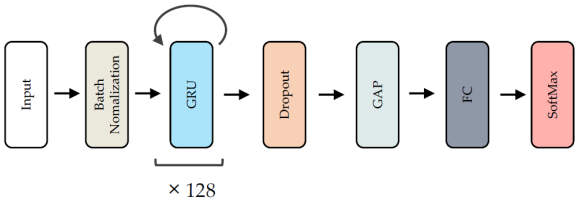

Fig.9: *The GRU architecture used in this work.*

One significant constraint of such a network is that it is unidirectional. That is, besides the input data, the output at a specific time step is determined solely by the data contained in the input sequence in the previous time step. Nevertheless, under certain circumstances, it could be advantageous to make forecasts by looking not only at the history but also at the future. Alsarhan et al. [39] suggested a framework for identifying human actions based on bidirectional gated recurrent units (BiGRUs). The findings suggest that using the BiGRU model to detect human activity from sensor data is quite efficient. The BiGRU structure used in this investigation is depicted in Fig. 10.

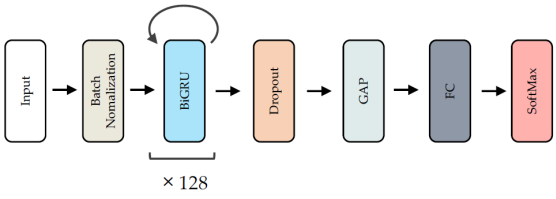

Fig.10: *The BiGRU architecture used in this work.*

4.3 Experimental Findings

The purpose of this research was to determine whether sensor-based FDS can be used to identify fall actions using deep learning models. We gathered ADLs and fall activities using three FDS datasets: UpFall, SisFall, and UMAFall. Preprocessed accelerometer and gyroscope data was used to train and then analyze the trained deep learning models using a 5-fold cross-validation procedure. Tables 3, 4, and 5 summarize the experimental findings using the UpFall, SisFall, and UMAFall datasets, respectively.

4.4 Comparison of Results with Previous Work

This section compares FallNeXt to earlier efforts that have influenced the recommended network. Previously, ResNeXt [34] was suggested for image categorization, and the ResNeXt model is based on the cross-layer link concept of ResNet and integrates the VGG and Inception networks. FallNeXt is based on the design concept of ResNeXt and is intended to distinguish falls from other daily life activities for reliable fall detection with low-memory computing. To assess the effectiveness of fall detection, we performed further investigations using three data sets (UpFall, SisFall, and UMAFall). In Table 6, the comparison findings are shown.

5. RESEARCH DISCUSSION

This section discusses the enhancements to the recommended FallNeXt model. The obtained results are compared utilizing three datasets depicted in Fig. 11, Fig. 12, and Fig. 13.

5.1 Comparison Results with Baseline Models

The comparative results of the training parameters of the deep learning models are illustrated in Fig. 14. When considering the number of training parameters used, notice that the FallNeXt model uses fewer parameters than all of the other deep learning models in the experiment. These findings make it clear that the proposed FallNeXt model is less complicated than the other deep learning models and consumes less energy when practically implemented.

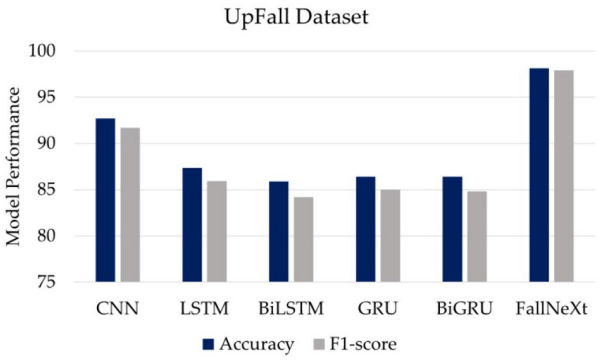

Fig.11: *Comparative results of deep learning models using the UpFall dataset.*

These comparative results indicate that the proposed FallNeXt model achieved the highest accuracy and F1-scores for the three datasets.

5.2 Fall Detection Performance of the FallNeXt Model

In this part, we investigate the effect of the FallNeXt model's multi-branch aggregation. Considering

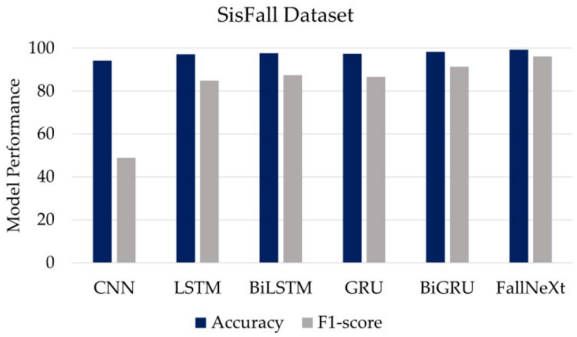

Fig.12: *Comparative results of deep learning models using the SisFall dataset.*

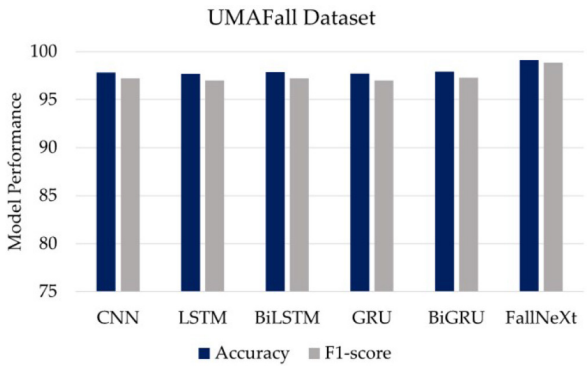

Fig.13: *Comparative results of deep learning models using the UMAFall dataset.*

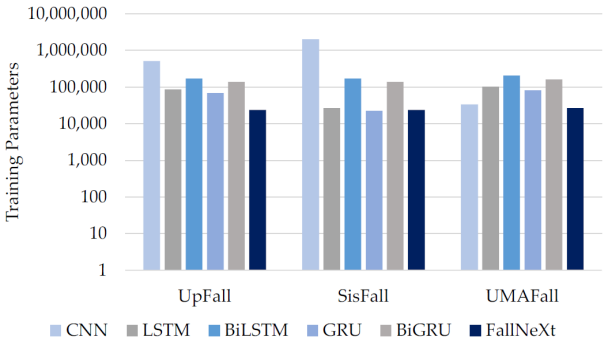

Fig.14: *Comparative results of training parameters of deep learning models.*

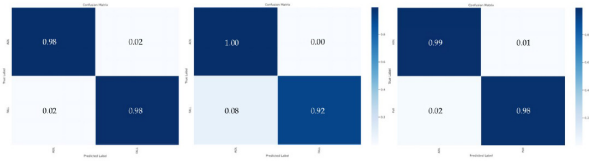

Fig.15: *Confusion matrices of the FallNeXt based on different datasets: (a) UpFall, (b) SisFll, and (c) UMAFall.*

Table 3: *Experimental result of deep learning models using the UpFall dataset*

Model	Parameter	Performance		
		Accuracy	Loss	F1-score
CNN	508,098	92.69% (±1.557%)	0.87 (±0.252)	91.70% (±1.962%)
LSTM	85,890	87.34% (±1.535%)	0.66 (±0.104)	85.94% (±1.743%)
BiLSTM	171,394	85.90% (±1.923%)	0.83 (±0.088)	84.22% (±2.170%)
GRU	68,994	86.41% (±0.960%)	0.98 (±0.092)	85.00% (±1.107%)
BiGRU	137,602	86.41% (±2.645%)	0.74 (±0.200)	84.81% (±3.095%)
FallNeXt	23,870	98.13% (±0.693%)	0.23 (±0.162)	97.92% (±0.775%)

Table 4: *Experimental result of deep learning models using the SisFall dataset*

Model	Parameter	Performance		
		Accuracy	Loss	F1-score
CNN	2,018,754	94.17% (±0.063%)	0.28 (±0.137)	48.89% (±0.529%)
LSTM	26,754	97.13% (±0.145%)	0.11 (±0.008)	84.78% (±0.851%)
BiLSTM	171,394	97.61% (±0.326%)	0.12 (±0.023)	87.39% (±2.174%)
GRU	22,402	97.47% (±0.244%)	0.10 (±0.018)	86.64% (±1.732%)
BiGRU	137,602	98.30% (±0.317%)	0.08 (±0.022)	91.44% (±1.759%)
FallNeXt	23,870	99.16% (±0.133%)	0.04 (±0.012)	96.14% (±0.641%)

Table 5: *Experimental result of deep learning models using the UMAFall dataset*

Model	Parameter	Performance		
		Accuracy	Loss	F1-score
CNN	33,474	97.84% (±0.324%)	0.14 (±0.034)	97.21% (±0.414%)
LSTM	102,786	97.69% (±0.496%)	0.09 (±0.015)	97.01% (±0.644%)
BiLSTM	205,186	97.85% (±0.185%)	0.08 (±0.011)	97.20% (±0.241%)
GRU	81,666	97.70% (±0.140%)	0.08 (±0.007)	97.01% (±0.184%)
BiGRU	162,946	97.91% (±0.231%)	0.07 (±0.006)	97.29% (±0.305%)
FallNeXt	26,972	99.12% (±0.197%)	0.07 (±0.015)	98.87% (±0.250%)

Table 6: *Comparative results with the ResNeXt model*

Dataset	Model	Parameter	Performance		
			Accuracy	Loss	F1-score
UpFall	ResNeXt	18,966,658	94.07% (±0.138%)	0.29 (±0.125)	94.41% (±0.177%)
	FallNeXt	23,870	98.13% (±0.693%)	0.23(±0.162)	97.92% (±0.775%)
SisFall	ResNeXt	18,955,458	98.81% (±0.351%)	0.05 (±0.011)	94.39% (±1.611%)
	FallNeXt	23,870	99.16% (±0.133%)	0.04(±0.012)	96.14% (±0.641%)
UMAFall	ResNeXt	18,957,954	93.75% (±1.883%)	0.19 (±0.048)	91.88% (±2.492%)
	FallNeXt	26,972	99.12% (±0.197%)	0.07(±0.015)	98.87% (±0.250%)

the confusion matrices in Fig. 15, the findings reveal that the proposed FallNeXt model performed well on the three benchmark datasets, as indicated by its high F1-scores.

To conduct a more in-depth analysis, we examined various samples for which FallNeXt generated false-negative findings. We discovered that the samples consisted of data segments pertaining to transitional behaviors, including a portion of fall activity. As seen in Fig. 16, 3D-accelerometer data (displayed in blue shading) and 3D-gyroscope data (displayed in orange shading) had both stable and dynamic portions within the same segment. These factors impeded FallNeXt's ability to identify fall actions.

5.3 Complexity Analysis

We performed a complexity study of the proposed FallNeXt model based on the analytical technique for human activity recognition described in [40]. The complexity of each model is measured by its memory consumption, mean prediction time, and the number of trainable parameters. The model was evaluated on the three benchmark datasets (UpFall, SisFall, and UMAFall) used in this study.

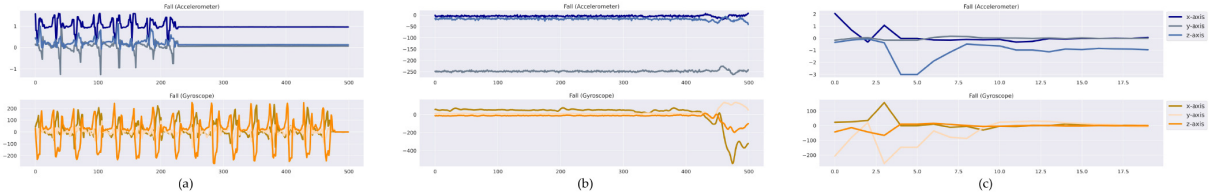

Fig.16: *Some samples that result in false negatives from different datasets: (a) UpFall, (b) SisFll, and (c) UMAFall.*

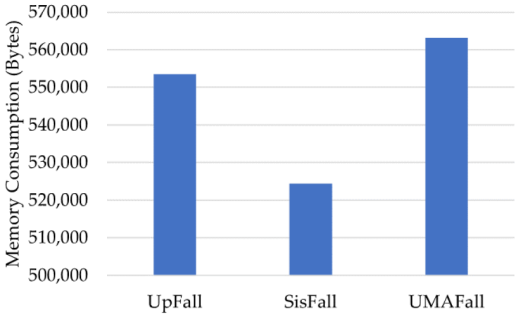

Fig.17: *Mean prediction time in seconds of of the FallNeXt model.*

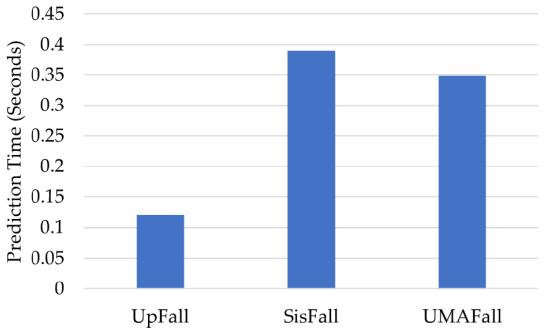

Fig.18: *Mean prediction time in seconds of of the FallNeXt model.*

5.3.1 Memory Consumption

Considering contemporary wearable technologies, memory usage is no longer a problem. Apple Watch Series 7 and later models already have 1 GB of RAM, whereas the Samsung Galaxy Watch 3 includes 1.5 GB of RAM. Consequently, designing smartphone or wristwatch applications to use FDS machine learning should not provide complications. If we were to create our own wearable device, memory usage would be more vital if we wanted to reduce the size of the hardware or enhance power consumption. As suggested in [40], the FallNeXt model is implemented on an iPhone XR using the Tensorflow Lite platform to monitor and compare the memory utilization in this study. Using an Xcode debug session, the memory utilization of each model can be estimated. Fig. 17 depicts the data obtained from monitoring memory utilization.

Based on the comparative findings in Fig. 17, FallNeXt uses less than 0.5 MB of RAM while processing the three datasets. The memory usage reveals that the FallNeXt model requires less than 1 MB of memory use to function on wearable devices.

5.3.2 Prediction Time

We continued comparing complexity and mean prediction time for efficiency purposes. To obtain the mean prediction, a series of test data samples were input into the Tensorflow Lite models, which were then averaged to determine the mean prediction time.

The research findings are shown in Fig. 18 as the mean prediction time in seconds for one window processed by the deep learning models on the three datasets. It can be seen that FallNeXt requires 0.12 to 0.38 seconds to produce a forecast.

5.3.3 Trainable Parameters

Considering memory usage and mean prediction time, we can examine our third model complexity indicator: the number of trainable parameters. A standard statistical indicator for deep neural networks is the number of trainable parameters. Each parameter's weight can be learned throughout model training, indicating one trainable parameter. The more parameters a deep learning network maintains, the more detailed data it can describe. However, it is more likely to overfit the training data if it is not very complicated.

Fig. 14 displays the findings of the trainable parameters of FallNeXt presented in this study. The values could be derived from the model summary of many tests using distinct HAR benchmark datasets. The findings demonstrate that FallNeXt has a substantially lower number of parameters than other deep learning models.

6. CONCLUSION AND FUTURE WORK

We proposed a deep learning network that is based on wearable sensor information and compared it to other networks in this research. The exploratory results indicate that the proposed FallNeXt network surpassed other benchmark models (CNN, LSTM, BiLSTM, GRU, and BiGRU). On the UpFall, SisFall, and UMAFall datasets, the proposed model had

identification accuracy rates of 98.13%, 99.16%, and 99.12%, respectively. The advantage of the FallNeXt model is its compact but highly effective model size for fall detection. It can potentially run on various wearable devices.

For future work, we plan to develop a multimodal fall detection system to detect falls and emit real-time alerts on wearable devices such as smartphones or smartwatches. We also plan to gather additional data on elderly falls and further train the network systems to enhance performance.

ACKNOWLEDGMENT

This research project was supported by the University of Phayao (Grant No. FF65-RIM041) and the Thailand Science Research and Innovation Fund.

References

[1] W. H. Organization, *World report on ageing and health*. World Health Organization, 2015.

[2] M. Mubashir, L. Shao and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144-152, 2013.

[3] K. Chaccour, R. Darazi, A. H. El Hassani, and E. Andrès, "From fall detection to fall prevention: A generic classification of fall-related systems," *IEEE Sensors Journal*, vol. 17, no. 3, pp. 812-822, 2017.

[4] A. L. S. de Lima, L. J. W. Evers, T. Hahn, L. Bataille, J. L. Hamilton, M. A. Little, Y. Okuma, B. R. Bloem and M. J. Faber, "Freezing of gait and fall detection in Parkinson's disease using wearable sensors: a systematic review," *Journal of Neurology*, vol. 264, pp. 1642-1654, 2017.

[5] N. el Halabi, R. A. Z. Daou, R. Achkar, A. Hayek and J. Börcsök, "Monitoring system for prediction and detection of epilepsy seizure," in 2019 Fourth International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), 2019, pp. 1-7.

[6] A. S. Syed, D. Sierra-Sosa, A. Kumar and A. Elmaghraby, "A deep convolutional neural network-xgb for direction and severity aware fall detection and activity recognition," *Sensors*, vol. 22, no. 7, 2022.

[7] S. Mekruksavanich, P. Jantawong, A. Charoenphol and A. Jitpattanakul, "Fall detection from smart wearable sensors using deep convolutional neural network with squeeze-and-excitation module," in *2021 25th International Computer Science and Engineering Conference (ICSEC)*, pp. 448-453, 2021.

[8] J. Fleming, C. Brayne and Cambridge collaboration, "Inability to get up after falling, subsequent time on floor, and summoning help: prospective cohort study in people over 90.," *The BMJ*, vol. 337, 11 2008.

[9] X. Wang, J. Ellul and G. Azzopardi, "Elderly fall detection systems: A literature survey," *Frontiers in Robotics and AI*, vol. 7, 2020.

[10] X. Xi, M. Tang, S. M. Miran and Z. Luo, "Evaluation of feature extraction and recognition for activity monitoring and fall detection based on wearable semg sensors," *Sensors*, vol. 17, no. 6, 2017.

[11] F. S. Butt, L. La Blunda, M. F. Wagner, J. Schäfer, I. Medina-Bulo and D. Gómez-Ullate, "Fall detection from electrocardiogram (ecg) signals and classi_cation by deep transfer learning," *Information*, vol. 12, no. 2, 2021.

[12] M. Bosch-Jorge, A.-J. Sánchez-Salmerón, A. Valera and C. Ricolfe-Viala, "Fall detection based on the gravity vector using a wide-angle camera," *Expert Syst. Appl.*, vol. 41, no. 17, pp.7980-7986, 2014.

[13] S. C. Agrawal, R. K. Tripathi and A. S. Jalal, "Human-fall detection from an indoor video surveillance," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1-5, 2017.

[14] X. Kong, Z. Meng, L. Meng and H. Tomiyama, "A privacy protected fall detection iot system for elderly persons using depth camera," in *2018 International Conference on Advanced Mechatronic Systems (ICAMechS)*, pp. 31-35, 2018.

[15] S. Mekruksavanich, A. Jitpattanakul, K. Sitthithakerngkiet, P. Youplao and P. Yupapin, "Resnet-se: Channel attention-based deep residual network for complex activity recognition using wrist-worn wearable sensors," *IEEE Access*, vol. 10, pp. 51142-51154, 2022.

[16] K. M. Shahiduzzaman, X. Hei, C. Guo, and W. Cheng, "Enhancing fall detection for elderly with smart helmet in a cloud-network-edge architecture," in *2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, pp. 1-2, 2019.

[17] S. A. Mousavi, F. heidari, E. Tahami and M. Azarnoosh, "Fall detection system via smart phone and send people location," in *2020 28th European Signal Processing Conference (EUSIPCO)*, pp. 1605-1607, 2021.

[18] K. Desai, P. Mane, M. Dsilva, A. Zare, P. Shingala and D. Ambawade, "A novel machine learning based wearable belt for fall detection," in *2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*, pp. 502-505, 2020.

[19] K. Singh, A. Rajput and S. Sharma, "Human fall detection using machine learning methods: A survey," *International Journal of Mathematical, Engineering and Management Sciences*, vol. 5, pp. 161-180, 11 2019.

[20] H.-W. Tzeng, M.-Y. Chen and J.-Y. Chen, "Design of fall detection system with floor pressure

and infrared image," in *2010 International Conference on System Science and Engineering*, pp. 131-135, 2010.

[21] H. Sadreazami, M. Bolic and S. Rajan, "Fall detection using standoff radar-based sensing and deep convolutional neural network," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 1, pp. 197-201, 2020.

[22] T. Yang, J. Cao and Y. Guo, "Placement selection of millimeter wave fmcw radar for indoor fall detection," in *2018 IEEE MTT-S International Wireless Symposium (IWS)*, pp. 1-3, 2018.

[23] S. Mekruksavanich and A. Jitpattanakul, "Deep residual network for smartwatch-based user identification through complex hand movements," *Sensors*, vol. 22, no. 8, 2022.

[24] S. Mekruksavanich, N. Hnoohom and A. Jitpattanakul, "A hybrid deep residual network for efficient transitional activity recognition based on wearable sensors," *Applied Sciences*, vol. 12, no. 10, 2022.

[25] S. Mekruksavanich and A. Jitpattanakul, "Sport-related activity recognition from wearable sensors using bidirectional gru network," *Intelligent Automation & Soft Computing*, vol. 34, no. 3, pp. 1907-1925, 2022.

[26] S. Mekruksavanich and A. Jitpattanakul, "Sport-related activity recognition from wearable sensors using bidirectional gru network," *Intelligent Automation & Soft Computing*, vol. 34, no. 3, pp. 1907-1925, 2022.

[27] N. Hnoohom, S. Mekruksavanich and A. Jitpattanakul, "An efficient resnetse architecture for smoking activity recognition from smartwatch," *Intelligent Automation & Soft Computing*, vol. 35, no. 1, pp. 1245-1259, 2023.

[28] I. Kiprijanovska, H. Gjoreski and M. Gams, "Detection of gait abnormalities for fall risk assessment using wrist-worn inertial sensors and deep learning," *Sensors*, vol. 20, no. 18, 2020.

[29] M. Musci, D. De Martini, N. Blago, T. Facchinetti and M. Piastra, "Online fall detection using recurrent neural networks on smart wearable devices," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1276-1289, 2021.

[30] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez and C. Peñafort-Asturiano, "Up-fall detection dataset: A multimodal approach," *Sensors*, vol. 19, no. 9, 2019.

[31] A. Sucerquia, J. D. López and J. F. Vargas-Bonilla, "Sisfall: A fall and movement dataset," *Sensors*, vol. 17, no. 1, 2017.

[32] E. Casilari, J. A. Santoyo-Ramón and J. M. Cano-García, "Umafall: A multisensor dataset for the research on automatic fall detection," *Procedia Computer Science*, vol. 110, pp. 32-39, 2017.

[33] 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, 2014.

[34] S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He, "Aggregated residual transformations for deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987-5995, 2017.

[35] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller and F. Petitjean, "Inceptiontime: Finding alexnet for time series classification," *Data Min. Knowl. Discov.*, vol. 34, no. 6, p. 1936-1962, Nov. 2020.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[37] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, 1997.

[38] K. Cho, B. van Merriënboer, D. Bahdanau and Y. Bengio, "On the properties of neural machine translation: Encoder_decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. Doha*, Qatar: Association for Computational Linguistics, pp. 103-111, Oct. 2014.

[39] T. Alsarhan, L. Alawneh, M. Al-Zinati and M. Al-Ayyoub, "Bidirectional gated recurrent units for human activity recognition using accelerometer data," in *2019 IEEE SENSORS*, 2019, pp. 1-4.

[40] S. Angerbauer, A. Palmanshofer, S. Selinger and M. Kurz, "Comparing human activity recognition models based on complexity and resource usage," *Applied Sciences*, vol. 11, no. 18, 2021.

Sakorn Mekruksavanich received the B.Eng. degree in computer engineering from Chiang Mai University, in 1999, the M.S. degree in computer science from the King Mongkut's Institute of Technology Ladkrabang, in 2004, and the Ph.D. degree in computer engineering from Chulalongkorn University, in 2012. He is currently a Faculty Member with the Department of Computer Engineering, School of Information and Communication Technology, University of Phayao, Phayao, Thailand. His current research interests include deep learning, human activity recognition, neural network modeling, wearable sensors, and applying deep learning techniques in software engineering.

Anuchit Jitpattanakul received the B.Sc. degree in applied mathematics from the King Mongkut'sInstituteof Technology NorthBangkok, Bangkok, Thailand, and the M.Sc. degree in computational science and the Ph.D. degree in computer engineering from Chulalongkorn University. He joined the Intelligent and Nonlinear Dynamic Innovations (INDI) Research Center, KMUTNB. He is currently a Faculty Member with the Department of Mathematics, King Mongkut's University of Technology North Bangkok. His current research interests include deep learning approaches applied to human activity recognition, wearable sensors, and healthcare applications.