science, engineering and health studies

Research Article

Science, Engineering and Health Studies
https://li01.tci-thaijo.org/index.php/sehs
ISSN (Online): 2630-0087

Convolutional neural network for wearable fall detection systems

Uttapon Khawnuan[1], Teppakorn Sittiwanchai[2] and Nantakrit Yodpijit[1]*

[1] Center for Innovation in Human Factors Engineering and Ergonomics (CIHFE[2]), Department of Industrial Engineering, Faculty of Engineering, King Mongkut's University of Technology North Bangkok, Bangkok 10800, Thailand
[2] Center for Innovation in Human Factors Engineering and Ergonomics (CIHFE[2]), Department of Production and Robotics Engineering, Faculty of Engineering, King Mongkut's University of Technology North Bangkok, Bangkok 10800, Thailand

***Corresponding author**:*
Uttapon Khawnuan
uttapon.kn@gmail.com

***Received**: 9 September 2022*
***Revised**: 3 January 2023*
***Accepted**: 17 January 2023*
***Published**: 24 October 2023*

***Citation**:*
Khawnuan, U., Sittiwanchai, T., and Yodpijit, N. (2023). Convolutional neural network for wearable fall detection systems. Science, Engineering and Health Studies, 17, 23040002.

ABSTRACT

Fall accidents are a common cause of critical injuries among older adults. Therefore, fall detection systems have garnered considerable attention in research and industry. Feature extraction is the key for detecting falls, but it is time-consuming and tedious process. Deep learning can autonomously extract features from raw sensor data. In this study, we proposed a fall detection algorithm for wearable devices using a convolutional neural network (CNN) to differentiate falls from activities of daily living. The proposed model achieved over 99% metrics (sensitivity, specificity, precision, accuracy, and F1 score) by evaluating two different public datasets and provided better classification performance compared to other fall detection models in the same dataset. The higher CNN performance was recognized without requiring complex data preparation and manual feature extraction. Results from this study could induce CNN through the application of classification problems in technological environments.

Keywords: classification algorithms; convolutional neural network; deep learning; machine learning; wearable fall detection systems

1. INTRODUCTION

The rapidly increasing aging populations in developed and developing countries amplify the demand for assisted living in elderly care (Hadjadji et al., 2022). Statistics indicate that older people over 65 years are prone to falls, resulting in severe health problems (Chen et al., 2019; Ren and Peng, 2019). Among various issues that affect the daily safety of the elderly, falls are a major problem affecting their health (Ren and Peng, 2019). Many elderlies tend to live alone. Automatic fall detection systems can serve as a solution to provide timely assistance to those who fall (Mubashir et al., 2013). Simply stated, fall detection devices can improve the quality of life and reduce the cost of medical care for people with fall risk (Özdemir and Barshan, 2014).

In the last two decades, owing to the availability of wearable devices, wearable-based fall detection systems have become popular and are extensively researched (Al Rakhami et al., 2021; Tucker et al., 2015). Several findings indicate that the accelerometer and gyroscope have been extensively used for detecting falls in wearable-based devices (Mubashir et al., 2013). Furthermore, the use of only one accelerometer utilizes less computational resources and energy than multi-sensors (Pires et al., 2020). This can increase the acceptance of fall detection devices as the frequency of recharging batteries is reduced (Özdemir and Barshan, 2014).

A critical challenge of wearable fall detection systems is to differentiate actual falls from other fall-like activities (Kwolek and Kepski, 2014). Unfortunately, the ineffective classification algorithm of the fall detection system makes the system unreliable and can pose a danger to the user (Xu et al., 2018). The threshold-based method is a basic wearable fall detection algorithm in which features are calculated from raw sensor data and evaluated to determine whether a fall has occurred (Razum et al., 2018). The use of only a threshold-based algorithm can result in many false alarms (Kwolek and Kepski, 2014).

Machine learning (ML) is a part of artificial intelligence in which mathematical algorithms (e.g., artificial neural network (ANN) and logistic regression (Abouzari et al., 2020) are used to improve performance by analyzing the existing data. ML methods extract features from the acquired data and classify the data to detect a fall (Panahi and Ghods, 2018). ML methods provide better performance than threshold-based methods for differentiating falls from activities of daily living (ADLs) (Khojasteh et al., 2018). However, ML performance relies on the quality of handcrafted features, which is a time-consuming process to develop and select the appropriate features for achieving optimal performance (Hossain et al., 2018).

Deep learning (DL) is a type of ML method that can extract features from the raw data (Souza et al., 2021), known as non-handcrafted features (Vernikos et al., 2019). Convolutional neural network (CNN) is a type of feedforward neural network in DL that uses convolution in place of matrix multiplication in at least one of the layers, inspired by how the human visual cortex processes information (Gao et al., 2021). CNN methods, which are well-known for their promising performance in human activity recognition (Gao et al., 2021), could be developed to classify falls and ADLs (He et al., 2019) using a 3-axis accelerometer and gyroscope for RGB bitmap.

The objective of this study was to develop a fall detection algorithm using CNN methods based on raw data from a single accelerometer. The performance of the proposed algorithm was compared to that of conventional ML algorithms using the same two datasets. The findings could be implemented into a wearable fall detection device as a conceptual model, which was presented in the conclusion section.

2. MATERIALS AND METHODS

2.1 Architecture of CNN

CNN transformed the input data into an output answer using a stack of different layers. Figure 1 shows that the CNN architecture contains convolutional layers, a rectified linear unit (ReLU) activation function, pooling layers, fully connected layers (FCL), and a softmax activation function. Convolutional layers comprise a series of convolution kernel, that extract local features from raw input data. Parameters of convolution kernels are size, number of kernels, padding, and stride, which should be customized and optimized according to input data and the architecture of the model (Patil and Rane, 2021).

The activation functions transformed the value of the input into the output. ReLU was used for all convolutional layers that allow more efficient training of deeper networks (Chen and Ho, 2019). Additionally, ReLU had fewer vanishing gradient problems as compared to sigmoidal activation functions (Ide and Kurita, 2017). ReLU is represented as follows:

$$f\left(z^{l(i.j)}\right) = \max\{0, z^{l(i.j)}\}, \quad (1)$$

where $f(.)$ is the activation function.

A pooling layer could reduce the dimensions of the extracted feature maps. Having few parameters enables the neural network model to train and run faster. Maxpooling is the most common function employed to reduce the data size from the convolution layer and accelerate the computing process (Souza et al., 2021).

FCL was the last layer in the network that has a multi-layer perceptron (MLP) neural network. The output data from the last convolution and pooling layer was flattened into a single dimension and sent to FCL. The model results were classified by FCL, where the number of output nodes was the same number of labeled data classes (Al Nahian et al., 2020).

The softmax activation function has been used in various classification methods to calculate the final output from the last FCL (Souza et al., 2021). This activation function represents the probability distribution of the targeted class over different classes (Rodríguez et al., 2018). This function provided the output of the class range of 0–1 and the sum of the outputs from all classes, which was 1.

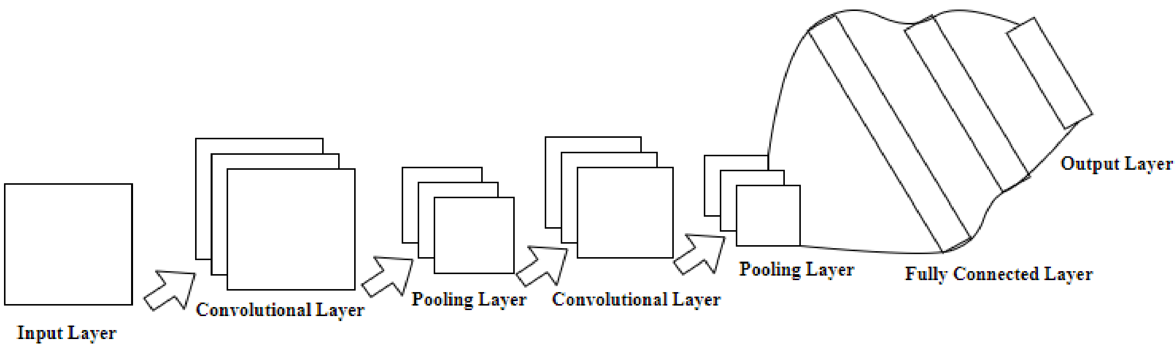

Figure 1. Typical CNN architecture

2.2 Datasets

The development of a wearable-based fall detection algorithm with CNN (FD-CNN) is based on the public fall datasets comprising tFall (Medrano et al., 2014) and SisFall (Sucerquia et al., 2017). This study utilized two public datasets of falls to evaluate their performance. The datasets were separated for training (60%), validating (10%), and testing (30%) to optimize the performance of the CNN model. Training data was used during the learning process for fitting weights of the CNN. Validation data was utilized to tune the hyperparameters of the CNN. Testing data was employed to assess the performance of the trained and validated CNN. The proposed model focused on the binary classification that detects falls and ADLs using the raw data from an accelerometer. All fall and ADL types were grouped into falls and ADL classes. Figures 2 and 3 show examples of acceleration data for fall and ADLs, respectively.

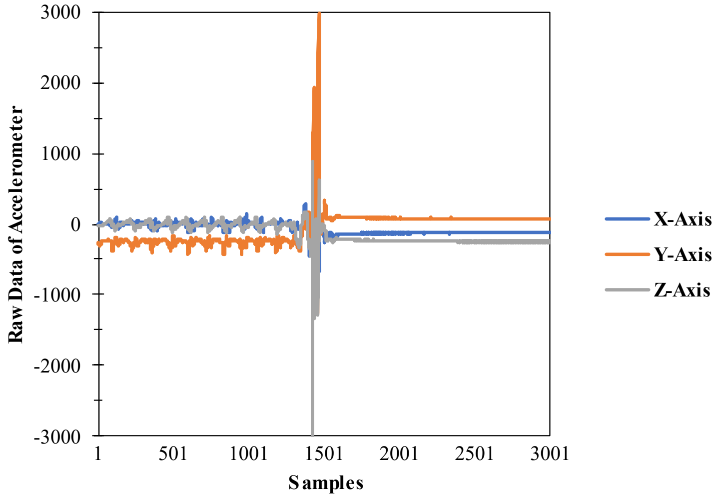

Figure 2. Fall acceleration data

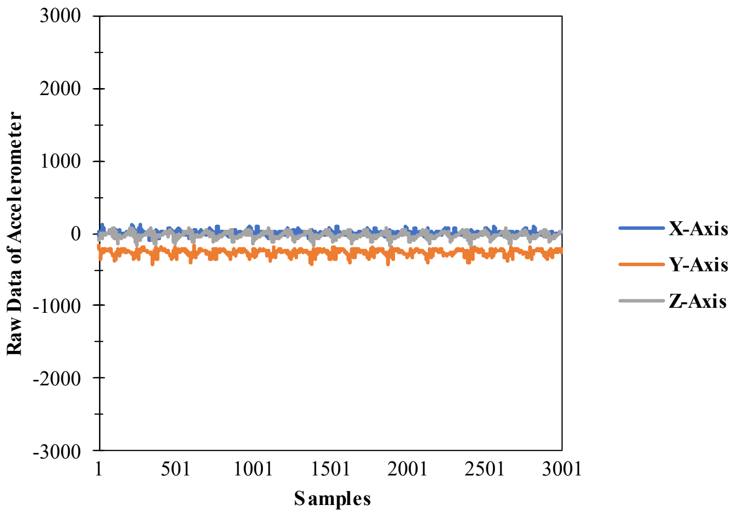

Figure 3. ADL acceleration data

2.2.1 tFall dataset

The tFall dataset contained 7816 activities (503 falls and 7313 ADLs) (Medrano et al., 2014). The data were gathered from a 3-axis accelerometer located in the pouches of 10 subjects. The data length was 6 s with a sampling frequency of 50 Hz. The synthetic minority oversampling technique was employed on the fall data to prevent imbalanced classification and overfitting due to the minority class of an imbalanced dataset (Wang et al., 2019).

2.2.2 SisFall dataset

The SisFall dataset consisted of 4505 activities (1798 falls and 2707 ADLs) (Sucerquia et al., 2017). The data were gathered from two 3-axis accelerometers and a 3-axis gyroscope installed at the waists of the subjects (23 young adults and 15 older adults). The data length was between 10 and 18 s with a sampling frequency of 200 Hz.

2.3 Data pre-processing

For wearable-based fall detection systems, there is no significant benefit in using human motion data with a sampling frequency higher than 25 Hz (Liu et al., 2018). Therefore, the original data of the two datasets was first down-sampled to 25 Hz. The impact point-based method was then used because a fall causes an impact (Wang et al., 2019). The acceleration magnitude (AM) (Equation 2) with a threshold of 1.6 g (Kau and Chen, 2015) and a window of 6 s (Hosseinian et al., 2019) was used to segment the down-sampled data at the peak of impact.

$$AM\left(a[n] = \sqrt{a[n]_x^2 + a[n]_y^2 + a[n]_z^2}\right) \tag{2}$$

2.4 Proposed CNN model

In this study, we proposed the FD-CNN model to differentiate falls from ADLs. This model used only a 3-axis accelerometer data as input vectors for training, validation, and testing, similar to previous studies (Torti et al., 2019; Wang et al., 2019). The FD-CNN model structure is presented as a flowchart in Figure 4. The input layer received segmented data from a 3-axis accelerometer. A batch normalization layer followed the input layer to make the data distribution consistent (Wang et al., 2022). Three convolution layers consisted of 16, 32, and 64 kernels, with a ReLU activation function connected to the batch normalization layer (Gadaleta and Rossi, 2018). Each kernel size was set to 1 × 3 with a stride of 1.

The maxpooling layer followed each convolution layer to reduce data dimension (Yadav et al., 2022). A dropout rate of 0.05 was used to avoid overfitting problems and gradient vanishing (Xu et al., 2021; Yadav et al., 2022). Two FCLs comprised 512 and 32 neurons, with a softmax activation function appended at the end of the model. In the development process, the hyperparameters of the model were manually adjusted and trained for 85 epochs with early stopping. The TensorFlow framework with Keras was used for developing CNN architectures on a computer equipped with Intel(R) Core(TM) i7-6700 at 3.40 GHz, 16 GB of RAM, and a GeForce GTX 1060 GPU with 6 GB of memory.

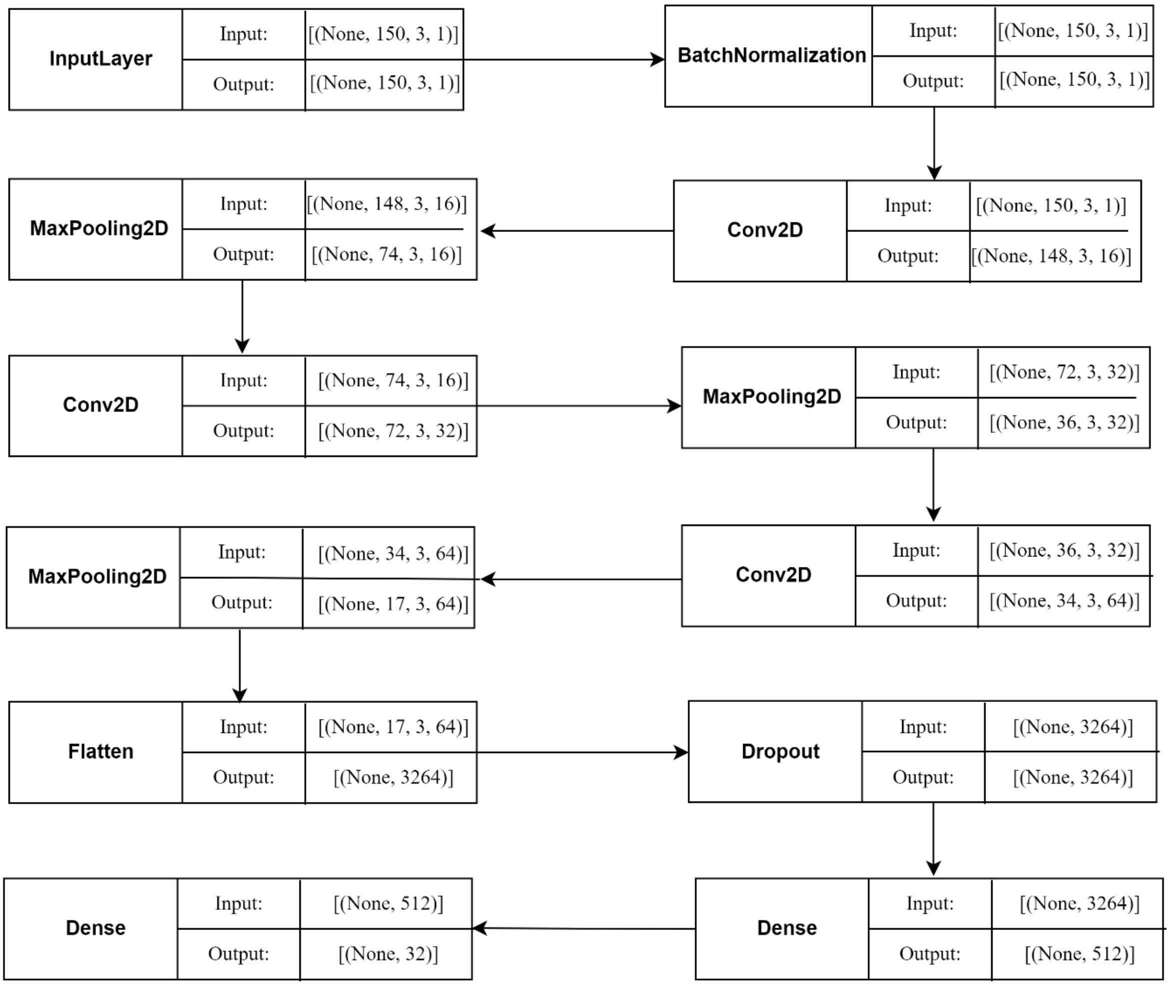

Figure 4. Structure of the FD-CNN model

3. RESULTS

The performance of the FD-CNN model was validated and compared to that of previous studies. Two datasets were used to test the FD-CNN model.

3.1 Experiments on tFall dataset

The proposed FD-CNN model was tested and validated on the tFall dataset. Figures 5(a) and 5(b) illustrate the accuracy and loss plots on the tFall dataset, respectively, achieving a steady state in 8 epochs. In the training process, the accuracy showed continual improvement, whereas the loss showed normal convergence. Thus, it could be implied that the model is reliable and valid (Zhang et al., 2022). The confusion matrix of this dataset is presented in Figure 6, where very few falls and ADLs were misclassified to other classes. The results of the testing set indicated that accuracy was 99.15%, precision was 99.15%, sensitivity was 99.06%, specificity was 99.25%, and the F1-score was 0.9915. The running time of the tFall dataset was 2 ms.

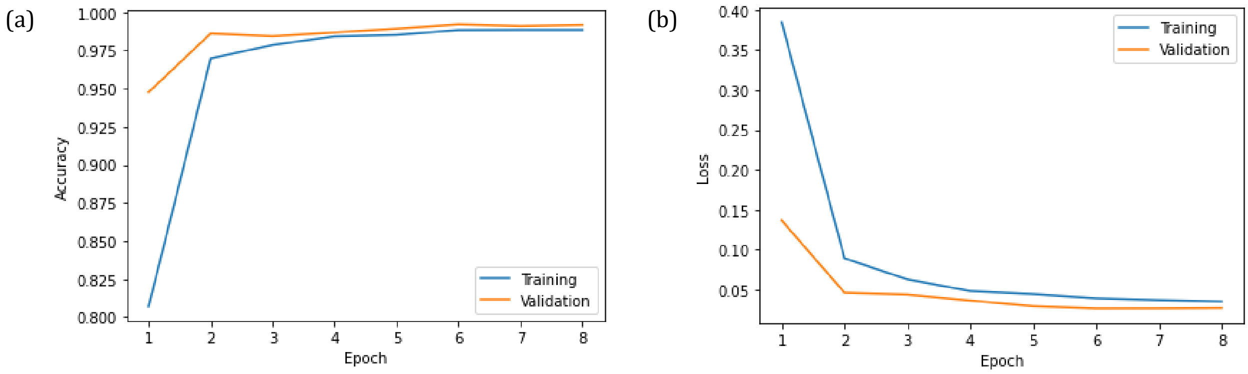

Figure 5. Training and validation using the tFall dataset: (a) accuracy plot, (b) loss plot

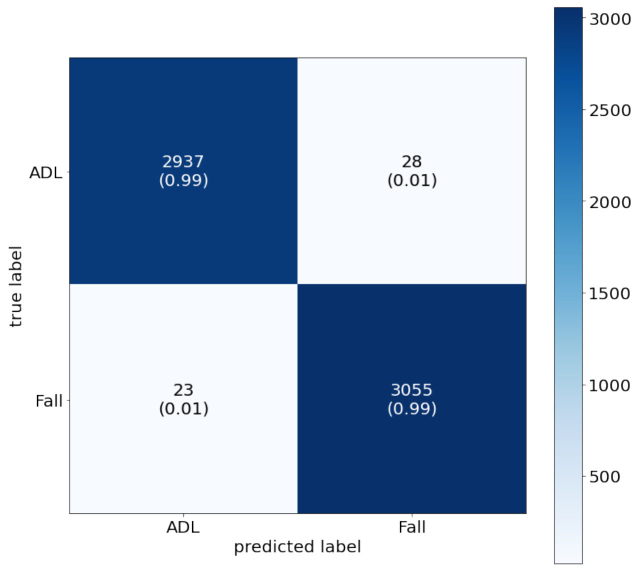

Figure 6. Confusion matrix for testing the data of the tFall dataset

3.2 Experiments on SisFall dataset

The proposed model was tested and validated for the SisFall dataset. Figures 7(a) and 7(b) illustrate the accuracy and loss plots, respectively, achieving a steady state in 11 epochs. The confusion matrix of this dataset is presented in Figure 8. The findings of the tFall dataset described in Section 3.1 are shown in Figures 7 and 8. In the testing phase, the results indicated accuracy (99.10%), precision (99.09%), sensitivity (98.54%), specificity (99.81%), and F1-score (0.9909). The running time of SisFall dataset is 2 ms.

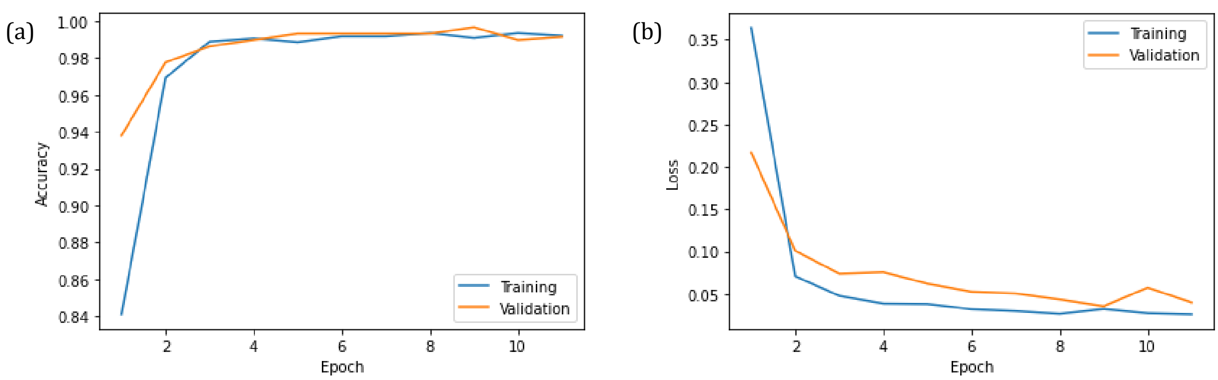

Figure 7. Training and validation using the SisFall dataset: (a) accuracy plot, (b) loss plot

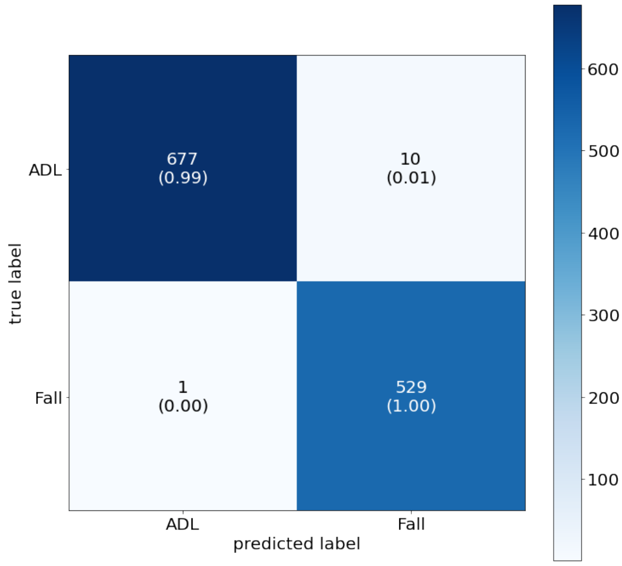

Figure 8. Confusion matrix for the testing data of the SisFall dataset

4. DISCUSSION

Fall can cause death and injury among the elderly. Fall detection is critical and has gained more attention among researchers. ML models are commonly used for recognizing falls among ADLs of the elderly, e.g., SVM (Carletti et al., 2017; Medrano et al., 2014), Kmeans + ANN (Medrano et al., 2014), MLP (Nguyen et al., 2018), k-nearest neighbor (KNN) (Nguyen et al., 2018), decision tree (DT) (Badgujar and Pillai, 2020), gradient boosting (GB) (Zurbuchen et al., 2020), and bidirectional long short-term memory (BiLSTM) (Waheed et al., 2021). The mentioned models have unsatisfactory performance and rely on the quality of handcrafted features. They are time-consuming and tedious (Hossain et al., 2018; Ronao and Cho, 2016) owing to the requirement of extensive human labor and knowledge of manual feature designs (Souza et al., 2021). The handcrafted features are calculated and fed into the ML model for training and testing. Examples are (1) sum vector magnitude on the horizon plane, (2) maximum peak to peak AM, (3) angle between the z-axis and vertical, (4) orientation change in the horizontal plane, and (5) jerk (rate of acceleration change) (Badgujar and Pillai, 2020).

We developed the FD-CNN model for a fall detection algorithm utilizing the raw accelerometer data. The proposed algorithm was based on DL techniques, which could extract features from raw data. The performance of the FD-CNN model was tested on two public datasets (tFall and SisFall) using five metrics (sensitivity, specificity, precision, accuracy, and F1 score). Tables 1 and 2 show the comparisons between the previous and current studies with conventional ML methods in the same datasets. The performance metrics in Tables 1 and 2 were obtained from the original publications. FD-CNN was the model that archives the highest accuracy for detecting falls and ADLs on each dataset. The accuracy of the FD-CNN model reached more than 99% on both datasets.

Table 1. Comparison of the tFall dataset results of previous studies and current study

Method	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)	F1 score
SVM (Medrano et al., 2014)	95.40	92.40	-	93.90	-
Kmeans + ANN (Medrano et al., 2014)	91.00	90.30		90.70	
SVM (Carletti et al., 2017)	95.20	95.40	-	-	-
This study	99.06	99.25	99.15	99.15	0.9915

Table 2. Comparison of SisFall dataset results of previous studies and this study

Method	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)	F1 score
MLP (Nguyen et al., 2018)	98.26	99.62	-	99.05	-
KNN (Nguyen et al., 2018)	97.06	97.89	-	97.54	-
SVM (Nguyen et al., 2018)	99.73	97.70	-	98.55	-
DT (Badgujar and Pillai, 2020)	-	-	-	96.00	-
GB (Zurbuchen et al., 2020)	98.06	99.21	-	98.70	-
BiLSTM (Waheed et al., 2021)	100	95.45	94.28	97.41	-
CNN (He et al., 2019)	98.62	99.80	-	98.61	-
This study	98.54	99.81	99.09	99.10	0.9909

In this study, the raw data of one accelerometer was used as the input for the FD-CNN model, which provided higher classification performance. In accordance with the research of Souza et al. (2021), we found that DL could work with raw data to avoid the transformation of raw data to vectors and statistical parameters needed in other models. The selection of statistical parameters is difficult because it significantly impacts the performance of the classification models (Souza et al., 2021; Jia et al., 2020). Moreover, the lowered number of motion sensors can reduce computational resources and energy consumption when implementing the model into the device (Pires et al., 2020). The proposed model focused on binary classification to identify falls and ADLs. It achieved higher performance compared to multi-class classification, as reported by He et al. (2019), by feeding the RGB bitmap of accelerometer and gyroscope data into the CNN model. The result agrees with the previous research findings of Kerdjidj et al. (2020), indicating that binary classification has better performance than multi-class classification. The hyperparameters of the FD-CNN model are shown in Table 3.

Table 3. Hyperparameters of the FD-CNN model

Hyperparameters	Value
Number of Batch Normalization Layer	1
Number of Convolutional and Maxpooling Layers	3
Number of Filters of Conv2D 1	16
Pool Size of Maxpooling 1	16
Number of Filters of Conv2D 2	32
Pool Size of Maxpooling 2	32
Number of Filters of Conv2D 3	64
Pool Size of Maxpooling 3	64
Dropout Rate	0.05
Number of Dense Layers	2
Dense Neurons 1	512
Dense Neurons 2	32

5. CONCLUSION

In this study, we developed a fall detection model based on wearable sensors. The findings demonstrated the effectiveness of using CNN-based methods on the raw data of an accelerometer to detect falls and ADLs. Complex data preparation and manual feature extraction were not necessary for this model. Overall, the detection performance of the FD-CNN model is suitable and demonstrates that CNN with the raw data of a single accelerometer could be implemented in a realistic fall detection system. The limitation of this study was the use of only two public datasets for evaluating the performance of the model. In the future, the proposed methods can be implemented into a real-time system to detect falls based on TensorFlow Serving and IoT platforms. The raw data from the accelerometer is sent to the FD-CNN model, which is implemented on a cloud service. When a fall is detected, the fall detection device will send an alert message to caregivers and healthcare services, as demonstrated in Figure 9.

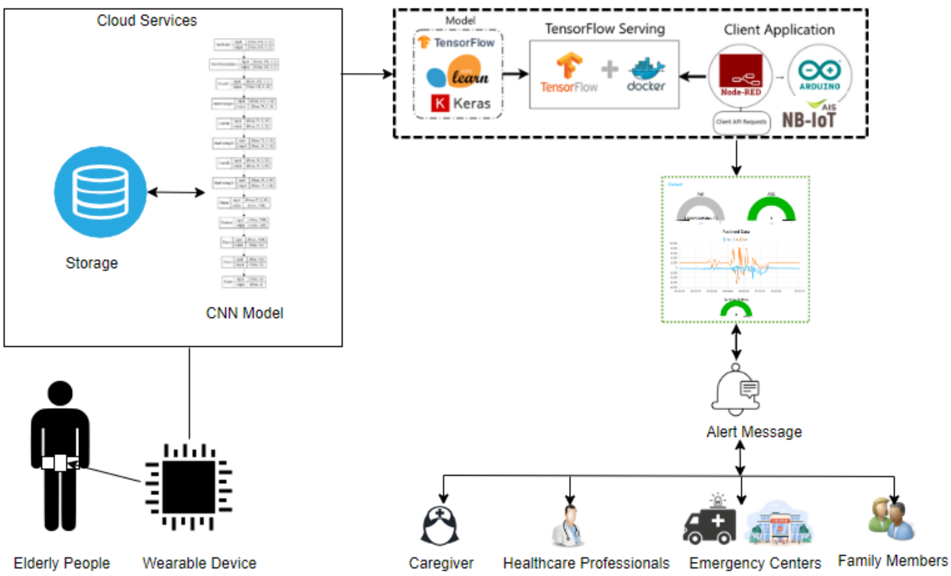

Figure 9. Conceptual model of the IoT-fall detection system

REFERENCES

Abouzari, M., Goshtasbi, K., Sarna, B., Khosravi, P., Reutershan, T., Mostaghni, N., Lin, H. W., and Djalilian, H. R. (2020). Prediction of vestibular schwannoma recurrence using artificial neural network. *Laryngoscope Investigative Otolaryngology*, 5(2), 278–285.

Al Rakhami, M. S., Gumaei, A., Altaf, M., Hassan, M. M., Alkhamees, B. F., Muhammad, K., and Fortino, G. (2021). FallDeF5: A fall detection framework using 5G-based deep gated recurrent unit networks. *IEEE Access*, 9, 94299–94308.

Al Nahian, M. J., Ghosh, T., Uddin, M. N., Islam, M. M., Mahmud, M., and Kaiser, M. S. (2020). Towards artificial intelligence driven emotion aware fall monitoring framework suitable for elderly people with neurological disorder. In *Brain Informatics* (Mahmud, M., Vassanelli, S., Kaiser, and M.S., Zhong, N., Eds.), pp. 275–286. Cham: Springer.

Badgujar, S., and Pillai, A. S. (2020). Fall detection for elderly people using machine learning. In *Proceedings of the 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT),* pp. 1–4. Kharagpur, India.

Carletti, V., Greco, A., Saggese, A., and Vento, M. (2017). A Smartphone-based system for detecting falls using anomaly detection. In *Image Analysis and Processing - ICIAP 2017* (Battiato, S., Gallo, G., Schettini, R., and Stanco, F., Eds.), pp. 490–499. Cham: Springer.

Chen, L., Li, R., Zhang, H., Tian, L., and Chen, N. (2019). Intelligent fall detection method based on accelerometer data from a wrist-worn smart watch. *Measurement*, 140, 215–226.

Chen, Z., and Ho, P. H. (2019). Global-connected network with generalized ReLU activation. *Pattern Recognition*, 96, 106961.

Gadaleta, M., and Rossi, M. (2018). IDNet: Smartphone-based gait recognition with convolutional neural networks. *Pattern Recognition*, 74, 25–37.

Gao, W., Zhang, L., Huang, W., Min, F., He, J., and Song, A. (2021). Deep neural networks for sensor-based human activity recognition using selective kernel convolution. *IEEE Transactions on Instrumentation and Measurement*, 70, 2512313.

Hadjadji, B., Saumard, M., and Aron, M. (2022). Multi-oriented run length based static and dynamic features fused with Choquet fuzzy integral for human fall detection in videos. *Journal of Visual Communication and Image Representation*, 82, 103375.

He, J., Zhang, Z., Wang, X., and Yang, S. (2019). A low power fall sensing technology based on fd-cnn. *IEEE Sensors Journal*, 19(13), 5110–5118.

Hossain, H. M. S., Khan, M. A. A. H., and Roy, N. (2018). DeActive: Scaling activity recognition with active deep learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(2), 66.

Hosseinian, S. M., Zhu, Y., Mehta, R. K., Erraguntla, M., and Lawley, M. A. (2019). Static and dynamic work activity classification from a single accelerometer: Implications for ergonomic assessment of manual handling tasks. *IISE Transactions on Occupational Ergonomics and Human Factors*, 7(1), 59–68.

Ide, H., and Kurita, T. (2017). Improvement of learning for CNN with ReLU activation by sparse regularization. In *Proceedings of the International Joint Conference on Neural Networks*, pp. 2684–2691. Alaska, USA.

Jia, L., Gu, Y., Cheng, K., Yan, H., and Ren, F. (2020). BeAware: Convolutional neural network (CNN) based user behavior understanding through WiFi channel state information. *Neurocomputing*, 397, 457–463.

Kau, L. J., and Chen, C. S. (2015). A smart phone-based pocket fall accident detection, positioning, and rescue system. *IEEE Journal of Biomedical and Health Informatics*, 19(1), 44–56.

Kerdjidj, O., Ramzan, N., Ghanem, K., Amira, A., and Chouireb, F. (2020). Fall detection and human activity classification using wearable sensors and compressed sensing. *Journal of Ambient Intelligence and Humanized Computing*, 11(1), 349–361.

Khojasteh, S. B., Villar, J. R., Chira, C., González, V. M., and De la Cal, E. (2018). Improving fall detection using an on-wrist wearable accelerometer. *Sensors*, 18(5), 1350.

Kwolek, B., and Kepski, M. (2014). Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer Methods and Programs in Biomedicine*, 117(3), 489–501.

Liu, K. C., Hsieh, C. Y., Hsu, S. J. P., and Chan, C. T. (2018). Impact of sampling rate on wearable-based fall detection systems based on machine learning models. *IEEE Sensors Journal*, 18(23), 9882–9890.

Medrano, C., Igual, R., Plaza, I., and Castro, M. (2014). Detecting falls as novelties in acceleration patterns acquired with smartphones. *PLoS ONE*, 9(4), e94811.

Mubashir, M., Shao, L., and Seed, L. (2013). A survey on fall detection: Principles and approaches. *Neurocomputing*, 100, 144–152.

Nguyen, L. P., Saleh, M., and Le Bouquin Jeannès, R. (2018). An efficient design of a machine learning-based elderly fall detector. In *Internet of Things (IoT) Technologies for HealthCare* (Ahmed, M., Begum, S., and Fasquel, J. B., Eds.), pp. 34–41. Cham: Springer.

Özdemir, A. T., and Barshan, B. (2014). Detecting falls with wearable sensors using machine learning techniques. *Sensors*, 14(6), 10691–10708.

Panahi, L., and Ghods, V. (2018). Human fall detection using machine vision techniques on RGB–D images. *Biomedical Signal Processing and Control*, 44, 146–153.

Patil, A., and Rane, M. (2021). Convolutional neural networks: An overview and its applications in pattern recognition. In *Information and Communication Technology for Intelligent Systems ICTIS 2020.* (Senjyu, T., Mahalle, P. N., Perumal, T., and Joshi, A., Eds.), pp. 21–30. Singapore: Springer.

Pires, I. M., Marques, G., Garcia, N. M., Flórez-Revuelta, F., Teixeira, M. C., Zdravevski, E., Spinsante, S., and Coimbra, M. (2020). Pattern recognition techniques for the identification of activities of daily living using a mobile device accelerometer. *Electronics*, 9(3), 509.

Razum, D., Seketa, G., Vugrin, J., and Lackovic, I. (2018). Optimal threshold selection for threshold-based fall detection algorithms with multiple features. In *Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO),* pp. 1513-1516. Opatija, Croatia.

Ren, L., and Peng, Y. (2019). Research of fall detection and fall prevention technologies: A systematic review. *IEEE Access*, 7, 77702–77722.

Rodríguez, P., Bautista, M. A., Gonzàlez, J., and Escalera, S. (2018). Beyond one-hot encoding: Lower dimensional target embedding. *Image and Vision Computing*, 75, 21–31.

Ronao, C. A., and Cho, S. B. (2016). Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications*, 59, 235–244.

Souza, R. M., Nascimento, E. G. S., Miranda, U. A., Silva, W. J. D., and Lepikson, H. A. (2021). Deep learning for diagnosis and classification of faults in industrial rotating machinery. *Computers and Industrial Engineering*, 153, 107060.

Sucerquia, A., López, J. D., and Vargas-Bonilla, J. F. (2017). SisFall: A fall and movement dataset. *Sensors*, 17(1), 198.

Torti, E., Fontanella, A., Musci, M., Blago, N., Pau, D., Leporati, F., and Piastra, M. (2019). Embedding recurrent neural networks in wearable systems for real-time fall detection. *Microprocessors and Microsystems*, 71, 102895.

Tucker, M. R., Olivier, J., Pagel, A., Bleuler, H., Bouri, M., Lambercy, O., Millán, J. R., Riener, R., Vallery, H., and Gassert, R. (2015). Control strategies for active lower extremity prosthetics and orthotics: A review. *Journal of NeuroEngineering and Rehabilitation*, 12, 1.

Vernikos, I., Mathe, E., Spyrou, E., Mitsou, A., Giannakopoulos, T., and Mylonas, P. (2019). Fusing handcrafted and contextual features for human activity recognition. In *Proceedings of the 14th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, pp. 1–6. Larnaca, Cyprus.

Waheed, M., Afzal, H., and Mehmood, K. (2021). Nt-fds-a noise tolerant fall detection system using deep learning on wearable devices. *Sensors*, 21(6), 1–26.

Wang, G., Li, Q., Wang, L., Zhang, Y., and Liu, Z. (2019). Elderly fall detection with an accelerometer using lightweight neural networks. *Electronics*, 8(11), 1354.

Wang, P., Li, Q., Yin, P., Wang, Z., Ling, Y., Gravina, R., and Li, Y. (2022). A convolution neural network approach for fall detection based on adaptive channel selection of UWB radar signals. *Neural Computing and Applications*, 35, 15967–15980.

Xu, T., Se, H., and Liu, J. (2021). A fusion fall detection algorithm combining threshold-based method and convolutional neural network. *Microprocessors and Microsystems*, 82, 103828.

Xu, T., Zhou, Y., and Zhu, J. (2018). New advances and challenges of fall detection systems: A survey. *Applied Sciences*, 8(3), 418.

Yadav, S. K., Luthra, A., Tiwari, K., Pandey, H. M., and Akbar, S. A. (2022). ARFDNet: An efficient activity recognition and fall detection system using latent feature pooling. *Knowledge-Based Systems*, 239, 107948.

Zhang, Z., Zhou, Z., Yang, X., Meng, H., and Wu, G. (2022). Convolutional neural network based on multi-directional local coding for finger vein recognition. *Information Sciences*, 623, 633–647.

Zurbuchen, N., Bruegger, P., and Wilde, A. (2020). A comparison of machine learning algorithms for fall detection using wearable sensors. In *Proceedings of the International Conference on Artificial Intelligence in Information and Communication (ICAIIC),* pp. 427–431. Fukuoka, Japan.