International Journal of Electrical and Computer Engineering (IJECE)
Vol. 12, No. 4, August 2022, pp. 4430~4438
ISSN: 2088-8708, DOI: 10.11591/ijece.v12i4.pp4430-4438

Accelerometer-based elderly fall detection system using edge artificial intelligence architecture

Osama Zaid Salah, Sathish Kumar Selvaperumal, Raed Abdulla
Faculty of Computing, Engineering and Technology, Asia Pacific University of Technology and Innovation, Kuala Lumpur, Malaysia

Article Info

Article history:

Received Mar 18, 2021
Revised Dec 17, 2021
Accepted Jan 11, 2022

Keywords:

Accelerometer
Deep learning
Edge artificial intelligence
Fall detection
LoRa

ABSTRACT

Falls have long been one of the most serious threats to elderly people's health. Detecting falls in real-time can reduce the time the elderly remains on the floor after a fall, hence avoiding fall-related medical conditions. Recently, the fall detection problem has been extensively researched. However, the fall detection systems that use a traditional internet of things (IoT) architecture have some limitations such as latency, high power consumption, and poor performance in areas with unstable internet. This paper intends to show the efficacy of detecting falls in a resource-constrained microcontroller at the edge of the network using a wearable accelerometer. Since the hardware resources of microcontrollers are limited, a lightweight fall detection deep learning model was developed to be deployed on a microcontroller with only a few kilobytes of memory. The microcontroller was installed in a low-power wide-area network based on long range (LoRa) communication technology. Through comparative testing of different lightweight neural networks and traditional machine learning algorithms, the convolutional neural network (CNN) has been shown to be the most suited, with 95.55% accuracy. The CNN model reached inference times lower than 37.84 ms with 61.084 kilobytes storage requirements, which implies the capability to detect fall event in real-time in low-power microcontrollers.

Corresponding Author:

Raed Abdulla
Faculty of Computing, Engineering and Technology, School of Engineering, Asia Pacific University of Technology and Innovation
Technology Park Malaysia, Bukit Jalil, Kuala Lumpur 57000, Malaysia
Email: raed@staffemail.apu.edu.my

1. INTRODUCTION

Since 2000, the medical field has significant advances leading to an increase in life expectancy by a rate of 5 years [1]. According to the National Institutes of Health (NIH), the current elderly population is 8.5% of the world’s population, and by 2050, the percentage will be increased to 20% [2]. Hence in that sense, providing healthcare services to the elder to reduce their daily life risks becomes increasingly demanding. The fall is one of the most common risk factors for elders that frequently occur in hospitals, nursing homes, or homes, with approximately 30% of falls causing injury [3]. According to the World Health Organization (WHO), about 30% of the elderly over 65 years get fell one time or more annually, and for the elderly over 80 years, this rate increased to 50% [4].

In such a scenario, automated fall detection can reduce the impact and consequences of falls among the elderly by detecting and reporting their occurrence [5]–[8]. Over the last decade, fall detection has become a hot topic of research where many researches have been carried on fall detection systems fall detection system (FDS) [9]–[15]. Based on the different sensors used in the detection, fall detection systems can be classified into two major classes context-aware and wearable systems [16]. However, context-aware systems are restricted to the deployment area, which usually implies installing devices in the different places where the user being monitored. Recently, interest in wearable sensor-based systems has increased rapidly due to the emergence of low-cost physical sensors [17]–[20]. The wearable-based devices provide real-time monitoring without the use of environment-based devices. Therefore, only user-related data are acquired. In such systems, simple and low-power devices are used, commonly microcontroller equipped with inertial measurement unit leading to minimize the device's size and increasing the battery life. Wearable devices also usually imply lower economic costs compared to context-aware systems [21].

For wearable-based fall detection devices, there are two main types of algorithm can be found which are threshold-based and machine learning-based approach. Although the threshold-based approach shows very low computational complexity, many difficulties are presented when adapted to new fall types. Machine and deep learning approaches are considered more advanced techniques where the possibility of fall prevention and damage mitigation is available. However, deep learning algorithms required high computational power due to the large performed algebraic operations. Therefore, running such models on limited resources microcontrollers can lead to high-power consumption and long response times. Consequently, develop a real-time wearable fall detection system based on deep learning is an open research problem.

Every day, the internet of thins (IoT) is the most creative resource in manufacturing, industrial, and residential systems, and it plays a critical role [22]–[24]. The research described in this paper aims to develop an IoT-based fall detection system to alleviate the elderly fear of not being discovered after a fall which can help them to live an active, normal life. Most of the previous works in the literature and existing systems utilize conventional IoT architectures where the raw data are collected from the sensing unit and sent to the cloud or local server to perform the fall prediction and detection. This type of IoT architecture causes latency, power wastage, and not allowing the use of a low-power wide-area network (LPWAN) as they have a low data rate. Also, most of the previous works in the literature utilizes conventional machine learning techniques such as support vector machine (SVM) and k-nearest neighbors (K-NN) which are not robust and could produce many false alarms and have a low detection rate. While the robust fall detection system should have fast inference time and long-range to reduce the time that elderly people remain lying on the floor following a fall that could have significant consequences. Also, the low-power operation should be ensured to reduce cost.

Torti *et al.* [25] proposed an embedded fall detection system based on deep learning. In the proposed system, long short-term memory (LSTM) (a special type of recurrent neural networks (RNN)) was trained on a public fall detection dataset called SisFall. SensorTile board was used in the system for fall detection. An accelerometer data was analyzed with the trained LSTM model. The proposed system had a fall detection accuracy of 98%. In terms of power consumption, the proposed device could run continuously for about 20 hours without recharging.

Yacchirema *et al.* [26] proposed an IoT-based fall detection system based on the ensemble machine learning algorithm. In the proposed methodology, simple moving average (SMA) and sliding-windows techniques were used to extract the features from the raw signal of a publicly accessible dataset "SisFall". The extracted features were used to train and test four machine learning classifiers, including decision trees, ensemble, logistic regression, and deepNet, to find the best model. Then, the IoT fall architecture consists of four main stages a wearable device, a wireless communication network, an IoT gateway, and cloud services were used. The results revealed that ensemble-random forest (RF) had the best performance with an average areas under the curve (AUC) of 0.995, 5.75s training time, 3.48s testing time, accuracy (98.72%).

Luna-Perejón *et al.* [27] proposed a wearable fall detector using recurrent neural networks (RNNs). In the proposed system, two RNN models were trained using the SisFall dataset. For the performance analysis and integration of the trained models, two STM32 32-bit microcontrollers were used. The achieved results in terms of accuracy and specificity are 96.3% and 96.4%, respectively. Besides, less than 40ms execution times were obtained.

The major contributions of the project were driven by the limitations of the previous works in the literature. The proposed system was based on edge artificial intelligence (AI) IoT architectures where a deep learning algorithm is used to process data acquired by a 3D accelerometer sensor at the local level (microcontroller) which is new features as the microcontrollers typically are limited in resources. The used edge AI architectures eliminate the privacy issue of transmitting millions of data and storing it in the cloud, as well as the bandwidth and latency limitations that reduce data transmission capacity.

2. RESEARCH METHOD

2.1. Proposed method

Figure 1 shows the proposed system architecture for elderly fall detection. The system architecture consists of three layers: edge layer, fog layer, and cloud layer. The edge layer is a wearable sensor node equipped with an inertial measurement unit (IMU) and long range (LoRa) transceiver to collect, analyze, and transmit the data to an IoT-gateway via LoRa communication technology. The Arduino Nano 33 Bluetooth low energy (BLE) Sense microcontroller will be used to implement the edge device. Data from the inertial measurement unit will be fed into a deep learning model deployed on the microcontroller to perform inference. Then, only information about the elderly's status and instant notification in case of a fall is transmitted to the IoT gateway over LoRa. The fog layer is a LoRa gateway, which will be implemented with a Raspberry Pi 4 single-board computer and a LoRa transceiver, directly connected to the internet. The LoRa gateway is responsible for receiving LoRa packets and hosting the cloud server. The last layer in the architecture is the cloud layer used for global storage, cloud services, and web/mobile application servers. The proposed architecture solved the research problem by moving the AI processing of the inference from the cloud to the edge. Running the inference directly on the microcontroller at the edge allows the system to function in poor or unstable internet areas. Also, reducing bandwidth which allows the use of communication technologies with high range but limited in bandwidth such as LoRa. Therefore, a high range will be achieved, which is essential for fall detection systems. Lastly, latency in conventional architecture is solved as the proposed architecture will lower centralized computing power and give more real-time response.

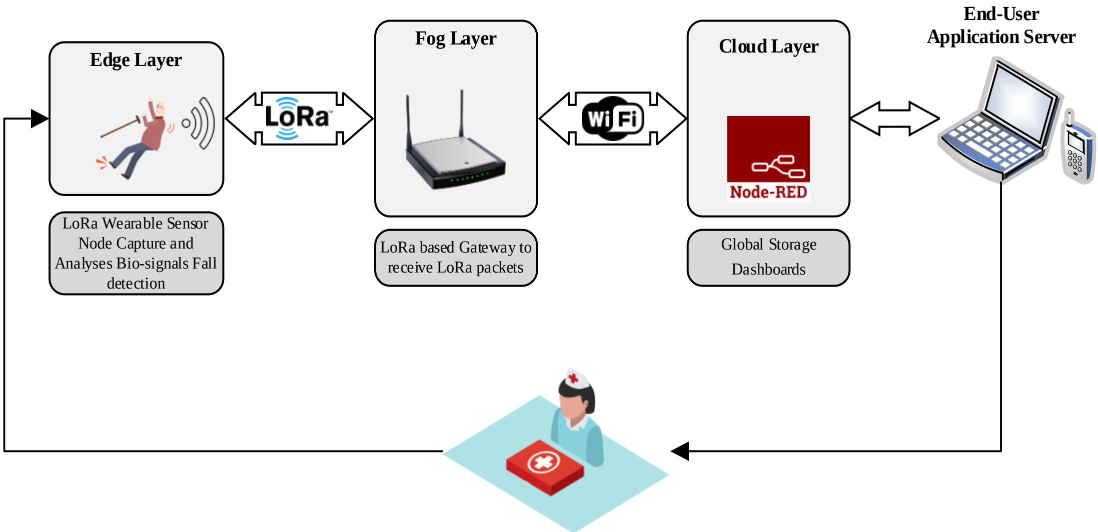

Figure 1. Proposed fall detection system architecture

2.2. Sesing unit

The core of the proposed FDS system is the Arduino Nano 33 BLE Sense board which is a completely new board. The board comes with a series of embedded sensors such as 9 axes inertial sensor which makes this board ideal for wearable devices. The main feature of this board, besides the impressive selection of sensors, is the possibility of running edge computing applications AI on it using tiny machine learning (TinyML). The Nano 33 BLE Sense board also has a communications chipset that can be both a BLE and Bluetooth client and host device which is something unique in the world of microcontroller platforms. Since the Nano 33 BLE Sense board has both embedded IMU and BLE communications chip, there is no external component needed to be connected to construct the BLE version of the sensing unit except for the battery. While the LoRa version of the proposed FDS system consists of Nano 33 BLE Sense board, NODEMCU ESP32 board, LoRa RFM95W transceiver module at 915 MHz band, two 18650 lithium batteries, a 5 V voltage regulator, and a 10 uF capacitor. The NODEMCU ESP32 board is acting as interface between Nano 33 BLE Sense board and LoRa RFM95W transceiver module as the LoRa Arduino library is not yet supported by the Nano 33 BLE Sense board. The NODEMCU ESP32 board and Nano 33 BLE Sense board are connected using UART serial communication protocol.

Figure 2 shows the detailed block diagram of the application deployed on the wearable sensing unit. The block diagram of the sensing unit can be divided into six main parts. The first part is the main loop of the system where the application runs in a continuous loop. The second part of the sensing unit application is the accelerometer handler. It is connected directly to the onboard accelerometer of the wearable sensing unit. This part of the application is responsible for managing the movement data capture from the accelerometer and writes them to the model's input tensor. Also, it uses a buffer to hold data during the inference process to avoid missing the movement data. The third part of the sensing unit application is the TensorFlow lite interpreter. It is connected directly to a TensorFlow lite model. This part of the application is responsible for running the inference based on the movement data being fed from the accelerometer handler. The convolutional neural network (CNN) model was included in the system in form of a C byte array. The fourth part of the sensing unit application is the activity predictor which takes the model's output and decides whether an activity has been detected, based on thresholds for both probability and the number of consecutive positive predictions. The final part of the sending unit application is the output handler which sends the data from the sensing units to the system gateway via either LoRa or BLE and prints output to the serial port depending on the recognized activity.

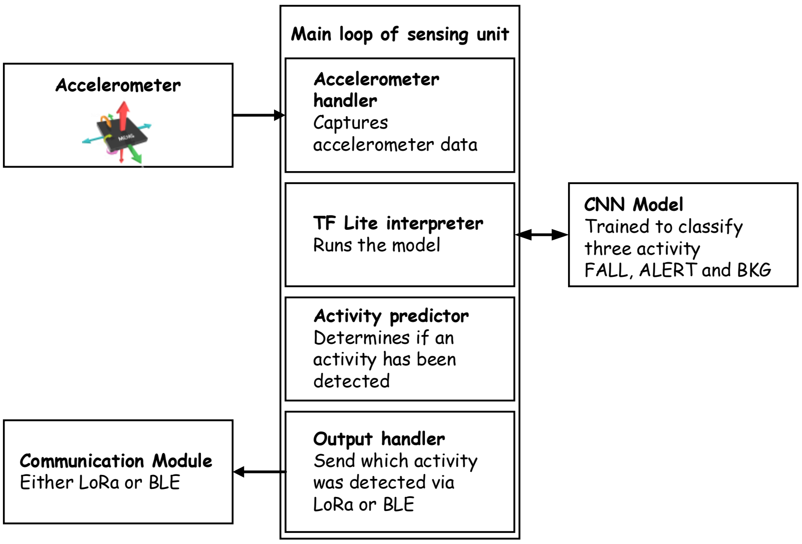

Figure 2. Detailed application block diagram of the wearable sensing unit

2.3. IoT gateway

The IoT gateway is a solution for enabling IoT communication. In the proposed system the gateway receives the data packets from the sensing unit and sends them to the cloud for global storage and displays the data on the system dashboard. The gateways have been implemented using a Raspberry Pi 4 model B running NOOBS software with a LoRa RFM95W transceiver module. The RFM95 LoRa transceiver module communicates with the Raspberry Pi using the serial peripheral interface (SPI) communication protocol.

Figure 3 shows the detailed block diagram of the system IoT gateway. The block diagram of the IoT gateway can be divided into three main parts. The first part is the communication initialization of the system communication protocol. The second part of the IoT gateway application is the data handler. This part is responsible for receiving the data packets that are sent by the edge device (sensing unit) and convert them from byte to float format so it can be sent to the node-red to be displayed in the system dashboard. The last part of the IoT gateway application is the nod red where the data is split and display in the system dashboard.

2.4. Machine learning for fall detection

The SisFall dataset [28] was used to train, test, and validate the proposed FDS, as it was deemed the most complete. To train the deep learning model with the chosen dataset, some pre-processing steps must be done. Firstly, the data was given in bits. In order to convert the acceleration data given in bits into gravity, the equation (1) was used.

$$Acceleration\ [g] = \frac{2 \times Range}{2^{Resolution}} \times AD \tag{1}$$

Since the used accelerometer in the SisFall dataset was ADXL345, the values +-16g for Range and 13 bits for resolution will be substituted in (1). Then, sliding windows have been produced because the Neural Network's inputs consist of a sequence of samples with a fixed length. Each window consists of three signals corresponding to X, Y, and Z accelerometer axes. The sequence of samples with a fixed width is referred to as a block. Therefore, to train the neural network's model, the block must be labeled according to the event class that belongs to it. The proposal established in [25] was used, in which each block is classified according to the appearance percentage of most the relevant class. The classes were labeled as FALL for the fall event, ALERT for the risk of falling and BKG for other activities. The BKG class includes daily life activities that not fall-related, such as walking and jumping. This process was applied for the whole dataset, and each block contains 256 samples, equal to 1.28 seconds. The blocks were 50% overlapped.

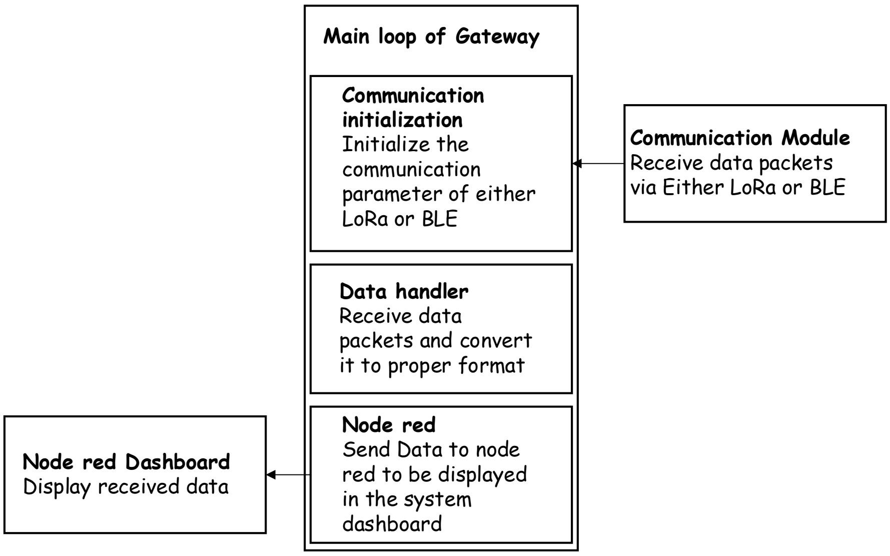

Figure 3. Detailed application block diagram of the system IoT gateway

The requirement that drove the design of deep learning architecture is that it should be deployed on a resource-constrained and relatively cheap device (a microcontroller equipped with an IMU sensor). As a result, five machine and deep learning models were trained on a SisFall dataset for evaluating the fall detection accuracy. Firstly, the dataset produced using the sliding window technique was split into training (60%), validation (20%), and testing (20%) sets based on data for subjects, e.g. 6 subjects for training, 2 for validation, and 2 for testing. For the training process, a graphic processor unit NVIDIA Tesla T4 provided by Google Colab was used, implemented in the Keras library. The architecture of the machine and deep learning models used is shown in Table 1.

Table 1. Architecture of the machine and deep learning models

Classifiers	Architecture
K-NN	15 neighbors and 5 neighbors
SVM	-
LSTM	Two Stacked LSTM layers with 32 neurons each, batch normalization [29], [30]
CNN	Two convolutional layers with 8 filters each, dropout [31]

3. RESULTS AND DISCUSSION

3.1. Fall detection performance of machine learning models

Performance comparative experiments were conducted between the developed lightweight neural networks (CNN and LSTM) and traditional machine learning algorithms (SVM and K-NN) where the experimental results are listed in Table 2. Among the traditional methods shown in Table 2, the SVM got the best accuracy with 82.72%. k-nearest neighbors (K-NN) (5 neighbors) accuracy was 3.61 lower than SVM, while the K-NN (15 neighbors) performs the worst with an accuracy of 78.64%. Much better performance can be obtained using lightweight neural networks where the CNN neural network has achieved accuracy higher than 95.5% which is even higher than the best traditional algorithms (82.27% of SVM). The LSTM neural network has achieved the highest performance in detecting falls with an accuracy of 96.78%. This significant improvement in accuracy shows the superiority of neural networks over traditional algorithms. The extraordinary results of neural networks are partially attributable to their advanced capability of modeling, but fundamentally to the high ability of neural networks to extract features and discover patterns.

Table 2. The fall detection performance of machine learning models

Classifiers	Conventional methods			Neural networks	
Metrics	K-NN (15 neighbors)	K-NN (5 neighbors)	SVM	CNN	LSTM
SEN. (%)	81.07	80.06	87.21	95.1	97.87
SPE. (%)	76.57	78.21	78.48	94.86	95.21
ACC. (%)	78.64	79.11	82.27	95.55	96.78

The proposed solution for this problem was to deploy the deep learning model on a microcontroller to run inference locally rather than transmitting the full sequences of raw data to the cloud. Thus, the developed CNN model was deployed in the Arduino Nano 33 BLE Sense board using the TensorFlow lite Arduino library because it is the only model supported by the board. Hence, further analysis was performed on the CNN model. Figure 4 shows the graph of the training history with respect to the loss function. As can be seen in the figure, the classification loss was drastically reduced. Finally, Figure 5 shows the confusion matrix resulting from training of the CNN model with the optimal hyperparameter. As can be seen, the model has achieved high accuracy in classifying the three classes in the training dataset.

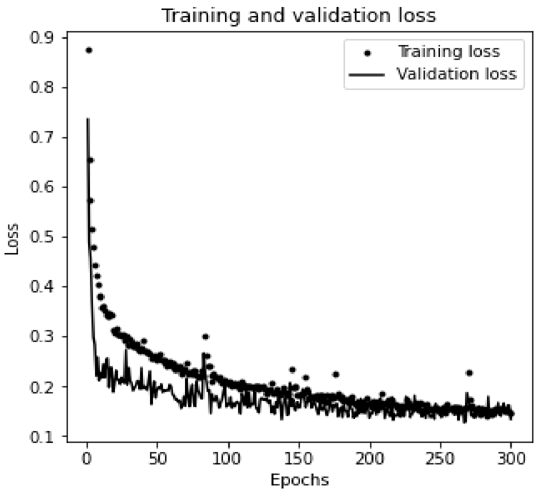

Figure 4. Graph of the training history

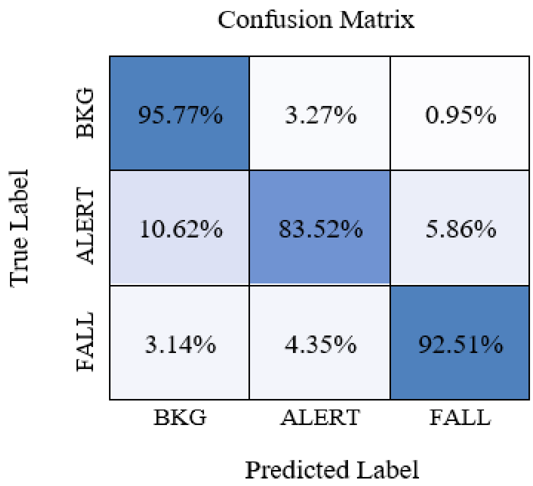

Figure 5. Confusion matrix

3.2. Inference time (response time) with different sampling rates

The latency in the traditional IoT architectures was one of the problems that the research intended to solve. Therefore, the proposed solution for this problem was to deploy the deep learning model on a microcontroller to run inference locally rather than transmitting the full sequences of raw data to the cloud. Thus, the developed CNN model was deployed in the Arduino Nano 33 BLE Sense board using the TensorFlow lite Arduino library because it is the only model supported by the board.

The inference time could be further enhanced where it depends on the inputs of the model such as the sampling rate and the window widths of the input. Hence, the fall detection inference time test was conducted on four different sampling rates. Table 3 shows the achieved inference time at each sampling rate. As can be seen from Table 3, the inference time was significantly influenced by the reduction of the sampling rate hence, high computational efficiency was achieved. This is due to the decrease in sampling rate leads to a decrease in the number of samples to form a 1.28s window of data that is required to make inferences. Therefore, the amount of data needed to be processed, computational power, and performed algebraic operations would be reduced. By this, the microcontroller would take less time to perform inference.

Table 3. Achieved inference time at each sampling rate

Sampling rate (Hz)	Window size(samples)	Inference time (μs)
25	32	37842
50	64	72347
100	128	123078
150	192	223782
200	256	302887

3.3. Power consumption of the different sensing unit versions

Figure 6 shows the graph which was plotted for the sensing unit power consumption test which was conducted on three sensing unit versions. The x-axis on the graph is representing the battery life in an hour, while the x-axis is representing the number of events being transmitted by the sensing unit. It can be noticed that with considering a large number of events, up to 40 K, the sensing unit's battery life is over 53 h when implementing the CNN model on the LoRa version of the sensing unit, over 38 h if the model implemented on the BLE version and only 8 h in the BLE sensing unit without embedded CNN model. This shows that the proposed solution has shown a huge improvement in terms of battery life over the traditional fall detection systems. This due to the fact that the proposed system only transmits information about the status of the elderly and battery condition rather than continuously transmitting the full sequences of raw data to the cloud.

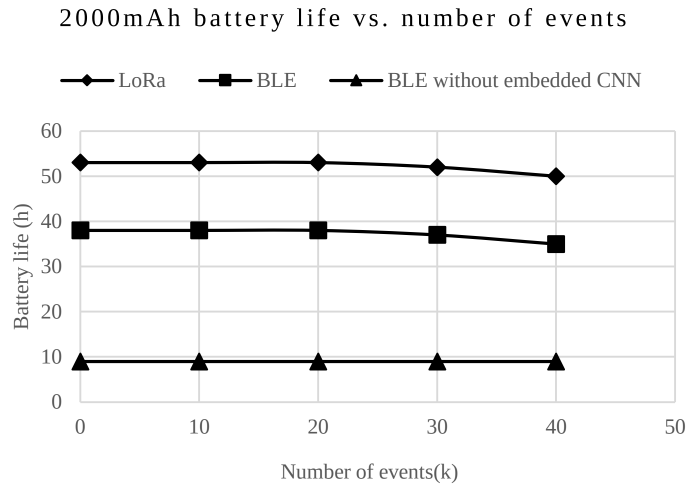

Figure 6. Sensing unit battery life

3.4. Range of the different sensing unit versions

Table 4 shows the achieved communication range of the two technologies used for each scenario. It can be observed that the LoRa technology has achieved a communication range sixty times larger compared to BLE. For instance, in the line of sight (LOS) scenario, LoRa has achieved a communication range of 180 meters while only 2 meters has been achieved with BLE in the same scenario. This is mainly due to the lower bandwidth and data rates used in LoRa, as well as the robustness of the LoRa-modulated signal. Also, it can be observed that when moving from LOS to the non-line of sight (NLOS) scenario, the communication range decreases. For example, the communication range decreased from 180 meters in the LoRa LOS scenario to a 150 meter in the NLOS scenario. Thus, it can be concluded that as the number of obstacles or barriers between the sensing unit and IoT gateway decreases, the communication range increases. This is due to the fact that the link budget is deducted by all sorts of obstacles between the sender and receiver. Thus, if the line budget is used up the receiver will only create some noise and no data will be received.

Table 4. Achieved range by each communication technology

Technology	Scenario	Achieved range (m)
BLE	LOS	3
	NLOS	2
LoRa	LOS	180
	NLOS	150

4. CONCLUSION

This work provides a study of the development of an enhanced accelerometer-based elderly fall prediction and detection system using deep learning with edge AI architecture. The obtained results reveal that the developed lightweight fall prediction and detection deep learning model was only 61.084 kilobytes, allowing the model to be executed into a microcontroller with only a few kilobytes of memory in real-time.

The model architecture with two convolutional layers has achieved 95.55% accuracy. Additionally, the testing of the inference time (response time) of the deep learning models with different sampling rates reveals that the fastest response time was obtained with a CNN model at a 25 Hz sampling rate which was (37.84 ms). The tested consumption and range with two versions of the sensing unit LoRa and the BLE version indicate that it is possible to use small batteries where the LoRa version has achieved the best performance. The LoRa version was able to achieve a range of 180 m, with only 37.72 mA current consumption during transmission mode which meets the low-power wide-area network (LPWAN) requirements.

REFERENCES

[1] WHO, "Life expectancy increased by 5 years since 2000, but health inequalities persist," *World Health Organization*, 2016. https://www.who.int/news/item/19-05-2016-life-expectancy-increased-by-5-years-since-2000-but-health-inequalities-persist (accessed Mar. 04, 2021).

[2] National Institute on Aging, "World's older population grows dramatically," *National Institutes of Health*, 2016. https://www.nia.nih.gov/news/worlds-older-population-grows-dramatically (accessed Mar. 09, 2021).

[3] R. L. Berg and J. S. Cassells, *Falls in older persons: risk factors and prevention*. National Academies Press, 1992.

[4] WHO, "WHO global report on falls prevention in older age," *World Health Organization*, 2008. https://apps.who.int/iris/handle/10665/43811 (accessed Mar. 10, 2021).

[5] D. Ajerla, S. Mahfuz, and F. Zulkernine, "A real-time patient monitoring framework for fall detection," *Wireless Communications and Mobile Computing*, vol. 2019, pp. 1–13, Sep. 2019, doi: 10.1155/2019/9507938.

[6] 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, Apr. 2019, doi: 10.3390/s19091988.

[7] G. Santos, P. Endo, K. Monteiro, E. Rocha, I. Silva, and T. Lynn, "Accelerometer-based human fall detection using convolutional neural networks," *Sensors*, vol. 19, no. 7, Apr. 2019, doi: 10.3390/s19071644.

[8] G. Wang, Q. Li, L. Wang, Y. Zhang, and Z. Liu, "Elderly fall detection with an accelerometer using lightweight neural networks," *Electronics*, vol. 8, no. 11, Nov. 2019, doi: 10.3390/electronics8111354.

[9] L. Liu, Y. Hou, J. He, J. Lungu, and R. Dong, "An energy-efficient fall detection method based on FD-DNN for elderly people," *Sensors*, vol. 20, no. 15, Jul. 2020, doi: 10.3390/s20154192.

[10] M. Panhwar, S. M. S. Shah, S. M. Z. S. Shah, S. M. Z. A. Shah, and B. S. Chowdhry, "Smart phone based fall detection using auto regression modeling in a non-restrictive setting," *Indian Journal of Science and Technology*, vol. 10, no. 5, pp. 1–6, Feb. 2017, doi: 10.17485/ijst/2017/v10i5/111274.

[11] A. Shahzad and K. Kim, "FallDroid: an automated smart-phone-based fall detection system using multiple kernel learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 35–44, Jan. 2019, doi: 10.1109/TII.2018.2839749.

[12] G. Fortino and R. Gravina, "Fall-MobileGuard: a smart real-time fall detection system," *Proceedings of the 10th EAI International Conference on Body Area Networks*, 2015, doi: 10.4108/eai.28-9-2015.2261462.

[13] G. G. Torres, R. V. Bayan Henriques, C. E. Pereira, and I. Müller, "An EnOcean wearable device with fall detection algorithm integrated with a smart home system," *IFAC-PapersOnLine*, vol. 51, no. 10, pp. 9–14, 2018, doi: 10.1016/j.ifacol.2018.06.228.

[14] J. He, C. Hu, and X. Wang, "A smart device enabled system for autonomous fall detection and alert," *International Journal of Distributed Sensor Networks*, vol. 12, no. 2, Feb. 2016, doi: 10.1155/2016/2308183.

[15] S. Gharghan *et al.*, "Accurate fall detection and localization for elderly people based on neural network and energy-efficient wireless sensor network," *Energies*, vol. 11, no. 11, Oct. 2018, doi: 10.3390/en11112866.

[16] X. Wang, J. Ellul, and G. Azzopardi, "Elderly fall detection systems: a literature survey," *Frontiers in Robotics and AI*, vol. 7, Jun. 2020, doi: 10.3389/frobt.2020.00071.

[17] A. Kamilaris and A. Pitsillides, "Mobile phone computing and the internet of things: a survey," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 885–898, Dec. 2016, doi: 10.1109/JIOT.2016.2600569.

[18] Y. Zhang, L. Sun, H. Song, and X. Cao, "Ubiquitous WSN for healthcare: recent advances and future prospects," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 311–318, Aug. 2014, doi: 10.1109/JIOT.2014.2329462.

[19] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, "Context-aware computing, learning, and big data in internet of things: a survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 1–27, Feb. 2018, doi: 10.1109/JIOT.2017.2773600.

[20] M. Chen, Y. Li, X. Luo, W. Wang, L. Wang, and W. Zhao, "A novel human activity recognition scheme for smart health using multilayer extreme learning machine," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1410–1418, Apr. 2019, doi: 10.1109/JIOT.2018.2856241.

[21] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *BioMedical Engineering OnLine*, vol. 12, no. 1, 2013, doi: 10.1186/1475-925X-12-66.

[22] T. Eldemerdash, R. Abdulla, V. Jayapal, C. Nataraj, and M. K. Abbas, "IoT based smart helmet for mining industry application," *International Journal of Advanced Science and Technology*, vol. 29, no. 1, pp. 373–387, 2020.

[23] R. Abdulla, A. Abdillahi, and M. K. Abbas, "Electronic toll collection system based on radio frequency identification system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 3, pp. 1602–1610, Jun. 2018, doi: 10.11591/ijece.v8i3.pp1602-1610.

[24] R. Lakshmanan, M. Djama, S. Perumal, and R. Abdulla, "Automated smart hydroponics system using internet of things," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, pp. 6389–6398, Dec. 2020, doi: 10.11591/ijece.v10i6.pp6389-6398.

[25] E. Torti *et al.*, "Embedded real-time fall detection with deep learning on wearable devices," in *2018 21st Euromicro Conference on Digital System Design (DSD)*, Aug. 2018, pp. 405–412, doi: 10.1109/DSD.2018.00075.

[26] D. Yacchirema, J. S. de Puga, C. Palau, and M. Esteve, "Fall detection system for elderly people using IoT and ensemble machine learning algorithm," *Personal and Ubiquitous Computing*, vol. 23, no. 5–6, pp. 801–817, Nov. 2019, doi: 10.1007/s00779-018-01196-8.

[27] F. Luna-Perejón, M. J. Domínguez-Morales, and A. Civit-Balcells, "Wearable fall detector using recurrent neural networks," *Sensors*, vol. 19, no. 22, Nov. 2019, doi: 10.3390/s19224885.

[28] A. Sucerquia, J. López, and J. Vargas-Bonilla, "SisFall: a fall and movement dataset," *Sensors*, vol. 17, no. 12, Jan. 2017, doi: 10.3390/s17010198.

[29] F. Luna-Perejon *et al.*, "An automated fall detection system using recurrent neural networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11526, Springer International Publishing, 2019, pp. 36–41.

[30] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?," *Advances in Neural Information Processing Systems*, pp. 2483–2493, 2018.

[31] S. Nitish, H. Geoffrey, K. Alex, S. Ilya, and S. Ruslan, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

BIOGRAPHIES OF AUTHORS

Osama Zaid Salah received the B.Eng. degree in Mechatronics Engineering from Asia Pacific University of Technology and Innovation (APU), Malaysia, in 2021, and now he is studying Master of Science (Data Science and Analytics) at Universiti Kebangsaan Malaysia. His research interests include artificial intelligence, robotic, and IoT. He can be contacted at email: osama.z.a.salah7@gmail.com.

Sathish Kumar Selvaperumal received his Ph.D. from Saraswathi Viswa Mahavidyalaya University, Chennai, India in the year 2014. He completed his B.E degree in Electronics and Communication Engineering in the year 2001 and M.E Applied Electronics in the year 2006, at Arulmigu Meenakshi Amman College of Engineering, Kanchipuram, Chennai, India. He has 17 years of teaching experience, and he is currently working as Associate Professor and Program leader for Telecommunication Engineering in Asia Pacific University (APU) of Technology and Innovation, Technology Park of Malaysia, Kuala Lumpur, Malaysia. He is the Final Year Project Manager in the school of Engineering, APU. He is a charted Engineer (C.Eng, U.K) and member of the Institution of Engineering and Technology (IET), Institute of Electrical and Electronics Engineers (IEEE). His research interests are image enhancement, image compression, optical communication, internet of things (IoT), antenna design, and artificial intelligence and robotics. He can be contacted at email: sathish@staffemail.apu.edu.my.

Raed Abdulla received the bachelor's degree in Electrical Engineering from AL-Mustansiria University, Baghdad, Iraq in 1997, the master's degree in Electronic Systems Design Engineering from Universiti Sains Malaysia, Malaysia in 2008 and he received his Ph.D. in Wireless and Mobile Systems from Universiti Sains Malaysia, Malaysia in 2012. He served Universiti Sains Malaysia, Malaysia from 2008 until 2011 as Postgraduate Teaching Assistant and in the year 2012 he works at School of Engineering, Asia Pacific University of Technology and innovation (APU), Malaysia, as a Senior lecturer until now. He is a charted Engineer (C.Eng, U.K) and member of the Institution of Engineering and Technology (IET), He has contributed in research and in the areas of radio frequency identification (RFID) and wireless sensor network (WSN) and internet of things (IoT). He can be contacted at email: raed@staffemail.apu.edu.my.