

Design of an Internet of Things-based Fall Detection System and Feedforward Neural Network with Edge Impulse

Wahyu Kurniawan*, Nike Dwi Grevika Drantantiyas, Ahmad Suaif

Engineering Physics Department, Institut Teknologi Sumatera, Lampung Selatan

*Corresponding author: wahyuwan5671@gmail.com

Received: June 4, 2025

Approved: June 11, 2025

Abstract

The increasing elderly population in Indonesia presents a significant challenge in ensuring safety, particularly concerning the risk of falls, which can lead to serious injuries. This study aims to design and develop a fall detection device based on the Internet of Things and a Feedforward Neural Network algorithm. The system was built using an ESP32 microcontroller and ADXL345 accelerometer sensor, and integrated with firebase and nodered to enable real time fall notifications via WhatsApp. The SisFall dataset was utilized, consisting of two primary classification labels fall and activities of daily living . Data processing and model training were carried out using the Edge Impulse platform. Three different models were evaluated, and Model 1 was selected as the best-performing model after hyperparameter tuning, achieving the highest accuracy of 99.7% and the lowest loss of 0.02. The model was quantized to int8 format for efficient edge deployment. Real world testing of the device involved 30 movement variations, each tested five times. Results showed that the system achieved a classification accuracy of 95.76%, precision of 97.22%, sensitivity of 93.33%, and specificity of 97.78%. These metrics indicate that the fall detection system was successfully developed and demonstrates excellent and reliable performance for practical applications.

Keywords: *fall detection, internet of things, feedforward neural network, edge impulse, tinyml*

Abstrak

Pertumbuhan populasi lansia di Indonesia membawa tantangan dalam hal keselamatan, khususnya risiko jatuh yang dapat menyebabkan cedera serius. Penelitian ini bertujuan untuk merancang dan mengembangkan alat deteksi jatuh berbasis Internet of Things dan algoritma Feedforward Neural Network. Sistem dibangun menggunakan mikrokontroler ESP32 dan sensor akselerometer ADXL345, serta diintegrasikan dengan firebase dan nodered untuk pengiriman notifikasi jatuh secara real time melalui WhatsApp. Dataset yang digunakan adalah SisFall, dengan dua label klasifikasi utama yaitu fall dan activities of daily living. Data diproses menggunakan platform Edge Impulse untuk pelatihan model feedforward. Tiga model dengan arsitektur berbeda diuji, dan Model 1 dipilih sebagai model terbaik berdasarkan hasil tuning hyperparameter dengan akurasi tertinggi 99,7% dan nilai loss terendah 0,02. Model ini kemudian dikuantisasi ke format int8 untuk efisiensi pada perangkat edge. Pengujian alat dilakukan dalam kondisi nyata dengan 30 variasi gerakan, masing-masing diuji sebanyak lima kali. Hasil menunjukkan bahwa sistem memiliki akurasi klasifikasi sebesar 95,76%, presisi 97,22%, sensitivitas 93,33%, dan spesifisitas 97,78%. Nilai-nilai ini mengindikasikan bahwa alat deteksi jatuh berhasil dibangun dan memiliki performa yang sangat baik serta dapat diandalkan untuk digunakan di lingkungan nyata.

Kata Kunci: *deteksi jatuh, internet of things, feedforward neural network, edge impulse, tinyml*

1. Introduction

The global growth of the elderly population shows a significant trend. According to data from the World Health Organization (WHO), the percentage of elderly people is estimated to increase from 12% in 2015 to 22% in 2050, equivalent to around 2 billion people[1]. In Indonesia, the number of elderly people reached more than 27 million in 2021 and is predicted to increase to 33.7 million by 2025[2]. This increase poses challenges in terms of health and safety, particularly regarding the risk of falls, which can cause serious injuries or even death[3]. Although approximately 75% of falls do not result in serious injury, the risk of falling increases with age, especially among individuals over 75 years old. Nearly 5% of falls result in fractures, typically in the hip, pelvis, humerus, or wrist. One of the factors contributing to this is osteoporosis. Osteoporosis causes bones to become brittle and can lead to fractures[4].

Falls among the elderly are a serious public health issue. Data shows that approximately 30% of elderly individuals living at home experience a fall each year, with half of them experiencing recurrent

falls[5]. In Indonesia, the prevalence of fall-related injuries among individuals over the age of 65 reaches 67.1% [1]. Causes include declines in physiological functions such as muscle strength, balance, and vision and hearing impairments[6]. Delays in providing first aid after a fall often exacerbate the impact of the injury.

Manual monitoring by family members for 24 hours is not always effective due to time and resource constraints. Therefore, an Internet of Things (IoT) technology solution is needed that can monitor the condition of the elderly in real-time and provide immediate notifications when a fall occurs, thereby minimizing the risk of more severe injuries from falls by reducing the time between the incident and first aid or assistance.

This study aims to design and develop a fall detection device that can monitor and send fall notifications via mobile devices and attempt to apply machine learning in detecting falls. Additionally, this study seeks to evaluate the performance of the designed device and the performance of the Feedforward Neural Network (FFNN) algorithm in classifying types of fall movements and regular activities.

Several studies have adopted the threshold approach as a fall detection method[7][8][9]. This approach has advantages in terms of simplicity of implementation and computational efficiency; however, it has limitations in distinguishing complex movement patterns or transitional activities that resemble fall characteristics, such as sudden sitting or bending, which can lead to false positives. The reason for this study is to explore the use of the Feedforward Neural Network algorithm, which can learn from data and recognize more complex patterns compared to the threshold method.

2. Material and Methods

This research designs an IoT-based fall detection tool with tinyML application using ESP32 and ADXL345 sensors. The system classifies two types of classes, namely fall and activities of daily living (ADL), using an Artificial Neural Network (ANN) algorithm implemented through Edge Impulse. The classification results are sent to Firebase, which can be monitored through a real-time database. Data from Firebase is then sent to Node-RED for display on the dashboard for monitoring purposes. If the classification result is a fall, Node-RED will send a notification and location to the designated WhatsApp account.

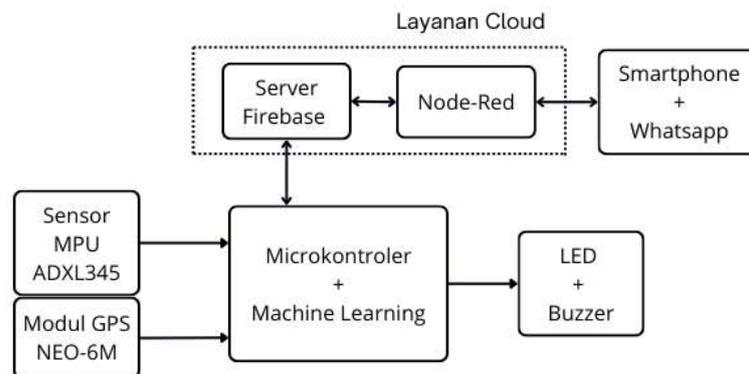


Fig. 1: diagram block system

Hardware Design

The fall detection device for the elderly developed in this study consists of an ADXL345 sensor[10], ESP32, LED indicator, buzzer, and neo-6m GPS module. The ADXL345 sensor is used to read acceleration data (ax, ay, az), while the Neo-6M module is used to obtain location information. The LED and buzzer serve as indicators that will light up and emit an audible alert when the device detects a fall. All components of the device are connected to the ESP32 microcontroller.

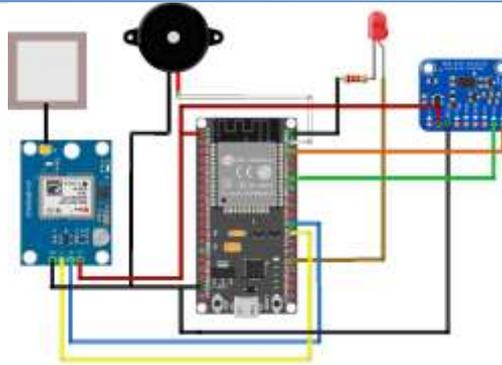


Fig. 2: Electronic Circuit

Dataset

The dataset was obtained from research conducted by[11], which introduced the SisFall dataset, containing recordings of falls and activities of daily living (ADL). The data was collected using a device equipped with two types of accelerometers and gyroscopes at a sampling frequency of 200 Hz. The dataset includes a total of 18 different daily activities and 15 different types of falls, as shown in **Table 1** and **Table 2**. The recordings were taken from 23 young adults, 14 older adults aged over 62 years performing 15 types of ADL, and one participant aged 60 years performing all daily activities and falls.

Table 1. Daily activities

Code	Activities	Trials	Duration
D01	Walk slowly	1	100 s
D02	Walking fast	1	100 s
D03	Slow jogging	1	100 s
D04	Fast jogging	1	100 s
D05	Walking up and down the stairs slowly	5	12 s
D06	Walking up and down stairs quickly	5	12 s
D07	Sit slowly on the half-high chair, wait a moment, and get up slowly	5	12 s
D08	Sit down quickly on the half-high chair, wait a moment, and get up quickly	5	12 s
D09	Sit down slowly on a low chair, wait a moment, and stand up slowly	5	12 s
D10	Sit down quickly on the low chair, wait a moment, and get up quickly	5	12 s
D11	Sat down for a moment, tried to get up, and fell back into the chair	5	12 s
D12	Sit down for a moment, lie down slowly, wait a moment, and sit down again	5	12 s
D13	Sit down for a moment, lie down quickly, wait a moment, and sit down again	5	12 s
D14	Lie on your back, turn onto your side, wait a moment, and then return to the supine position	5	12 s
D15	Stand up, slowly bend your knees, and get up	5	12 s
D16	Stand up, slowly bend over without bending your knees, and get up	5	12 s
D17	Tripped while walking	5	12 s
D18	Jumping slowly without falling (trying to reach something high)	5	12 s

Table 2. Fall activities

Code	Activities	Trials	Duration
F01	Jalling forward while walking due to slipping	5	15 s
F02	Falling backward while walking due to slipping	5	15 s
F03	Falling sideways while walking due to slipping	5	15 s
F04	Falling forward while walking due to tripping	5	15 s
F05	Fell forward while jogging due to tripping	5	15 s
F06	Falling forward while walking due to fainting	5	15 s
F07	Fell while walking, using hands on the table to support body and then fell	5	15 s
F08	Falling forward while trying to get up	5	15 s
F09	Falling to the side while trying to get up	5	15 s
F10	Falling forward while trying to sit down	5	15 s
F11	Falling backward while trying to sit down	5	15 s
F12	Falling sideways while trying to sit down	5	15 s
F13	Falling forward while sitting, due to fainting or falling asleep	5	15 s
F14	Falling backward while sitting, due to fainting or falling asleep	5	15 s
F15	Falling sideways while sitting, due to fainting or falling asleep	5	15 s

Data acquisition

The data obtained from the SisFall dataset was originally in txt file format, while the Edge Impulse platform only supports .json or .csv file formats. Therefore, the dataset files needed to be converted to .json format first. From the dataset, only the three-axis accelerometer data was taken, and then the acceleration values were converted to gravitational units.

The processed data was then uploaded to the Edge Impulse platform. However, data labeled ‘Fall’ contained mixed activity before and after the fall event, so it needed to be trimmed to focus only on the moment of the fall. Data labeled ‘ADL’ was used without modification. Data trimming was performed with a window length of 4000 ms, which was chosen as the most optimal for detecting fall events.

The total duration of the dataset uploaded to Edge Impulse is 14 hours, 41 minutes, and 5 seconds, divided into two labels: ADL and Fall. The dataset is then split into two parts: 80% for training and 20% for testing, as shown in **Figure 3**.

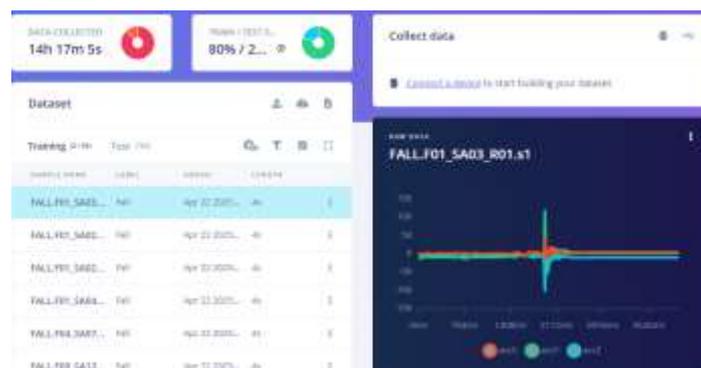


Fig. 3: Fall Dataset After Data Acquisition

As mentioned earlier, Edge-Impulse is used to create a feedforward neural network model. The “time series data” input block is used with a window size of 4,000 ms, a window increment of 2,000 ms, and a frequency of 200 Hz. The time series data is fed into the spectral analysis block to calculate features from the data, which are then input into the model. The classifier block takes the features generated by the spectral analysis block as input, and the output of the classifier is the class ADL or Fall, as shown in **Figure 4**.

In this classification task, raw accelerometer data collected from sensors captures complex movement patterns indicating a fall (Fall) or regular activity (ADL). The data includes information in both the time and frequency domains.

By applying spectral analysis to this dataset, the goal is to extract frequency-related features that can help distinguish between falls and regular activities. This approach is highly relevant because the frequency characteristics of specific fall-related movement patterns differ from those of common ADLs. The analysis is performed individually on each axis using the Fast Fourier Transform (FFT), specifically with an FFT length of 64. Through this spectral analysis, a set of features is obtained for the Accx, Accy, and Accz axes, totaling 18 features.

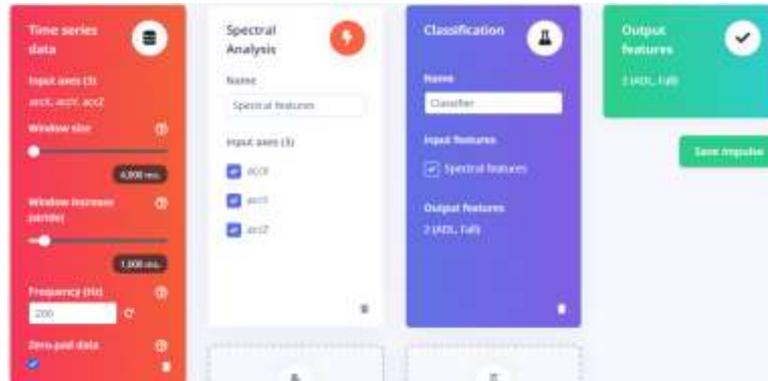


Fig. 4: Impulse Design on the Edge Impulse Platform

Arsitektur Model

The model was built using the Feedforward Neural Network (FFNN) algorithm for the model architecture, which has a small memory footprint and achieves excellent results[12]. It was designed to classify two main types of activities, namely Activities of Daily Living (ADL) and Falls. The model creation process was carried out entirely using the Edge Impulse platform.

The three models are Model 1 (3 dense layers with 64, 32, and 16 neurons and two dropouts of 0.2), Model 2 (3 dense layers with 32, 16, and 8 neurons without dropout), and Model 3 (3 dense layers with 20, 10, and 5 neurons and a dropout of 0.2).

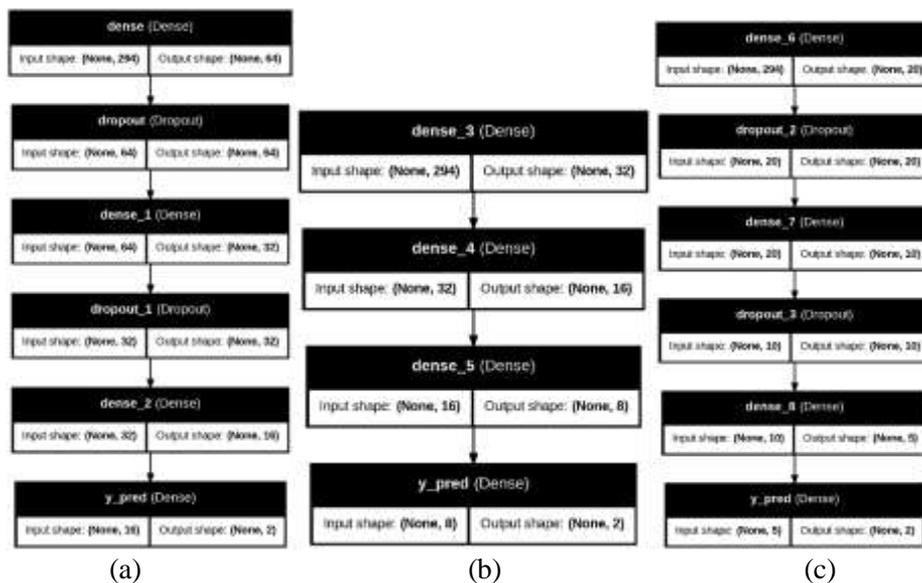


Fig. 5: (a). Model 1, (b). Model 2, (c). Model 3

Interface Design

This interface is designed using integration between Firebase Realtime Database, Node-RED as a monitoring dashboard platform, and WhatsApp services for notifications[13].

3. Results and Discussion

Hardware Design Results

The design of a fall detection device based on the Internet of Things (IoT) and Feedforward Neural Network (FFNN) has been successfully realized. This device is designed to be worn on the user’s body,

specifically on the waist, using a belt as the medium for placing the device. The waist area was chosen because this location is considered the most representative for accurately detecting changes in body position during normal activities or falls. The device is designed to be portable and easy to use, with all components assembled with consideration for space efficiency and physical safety. **Figure 6** shows the physical form and installation of the device on the user.



Fig. 6: (a). Physical Form of the Tool, (b). Device Placement on Users

This device consists of several main components, namely esp32, which functions as the main microcontroller that processes data from sensors and sends classification results to Firebase, and ADXL345 sensor, which measures acceleration values (accelerometer) on the x, y, and z axes. This sensor is positioned with the Z-axis facing forward. The NEO-6M GPS module provides location information in the form of latitude and longitude coordinates when the device detects a fall. The buzzer and LED provide immediate alerts when a fall is detected.

Interface Design Results

This interface was designed using integration between Firebase Realtime Database, Node-RED as a monitoring dashboard platform, and WhatsApp for notifications. Firebase is used as a real-time data storage medium. **Figure 7** shows the data stored in the Realtime database, which includes the ADL or Fall activity classification label, the confidence level of the model classification results, the status of the buzzer and LED actuators, and the latitude and longitude coordinates read from the GPS module.



Fig. 7: Firebase User Interface

The data from Firebase is then sent to Node-RED to be displayed on the dashboard for monitoring, and the latitude and longitude data obtained from Firebase is converted into a link to Google Maps, as shown in **Figure 8**.



Fig. 8: Nodered Monitoring

When the red node label indicates a fall, it will send a notification via the WhatsApp platform using the CallMeBot API. The message sent includes activity information, LED and buzzer status, and a link to the location of the incident in the form of Google Maps, as shown in **Figure 9**.



Fig. 9: WhatsApp notifications

Model Design Results

Hyperparameter Tuning

In this study, the author created three architectural models as mentioned in the research methods section, which were then subjected to hyperparameter tuning and tested in the design of the three models. This was done to determine the most effective and accurate model to be used for testing data.

Table 3 shows the training results of the three proposed models for activity classification (ADL and Fall) using data from the accelerometer on the x, y, and z axes. Tuning was performed at epochs 25, 50, 75, and 100 to observe the model's performance development as the number of training epochs increased, and an epoch value of 0.0005 was used for each model.

Table 3. Comparison of Accuracy and Loss of the Three Models

Model	Epoch 25		Epoch 50		Epoch 75		Epoch 100	
	accuracy	loss	accuracy	loss	accuracy	loss	accuracy	loss
Model 1	99.5%	0.03	99.7%	0.02	99.6%	0.03	99.6%	0.03
Model 2	99.6%	0.04	99.6%	0.03	99.7%	0.03	99.7%	0.04
Model 3	98.4%	0.04	99.6%	0.03	98.5%	0.03	98.6%	0.03

Based on the results in **Table 3**, Model 1 shows excellent and stable performance across all epochs. The highest accuracy of 99.7% was achieved at epoch 50, accompanied by the lowest loss value of 0.02. This indicates that the model has been able to learn optimally at that epoch without overfitting. After epoch 50, accuracy tends to remain stable but does not show significant improvement. Model 2 also has high accuracy performance, with the highest value of 99.7% at epoch 75 and 100. However, the loss value is relatively higher than Model 1, especially at epoch 100 with a value of 0.04, indicating the possibility of overfitting or inefficiency in the further training process. Model 3 has the highest accuracy of 99.6% and the minimum loss value of 0.03. Compared to the previous two models, Model 3's performance is the lowest, both in terms of accuracy and stability. This model does not show significant improvement even when the number of epochs is increased.

Based on the overall results obtained, Model 1 at epoch 50 was selected as the best model because it provides the best balance between high accuracy and low loss values and will be used for testing data.

Best Model Analysis

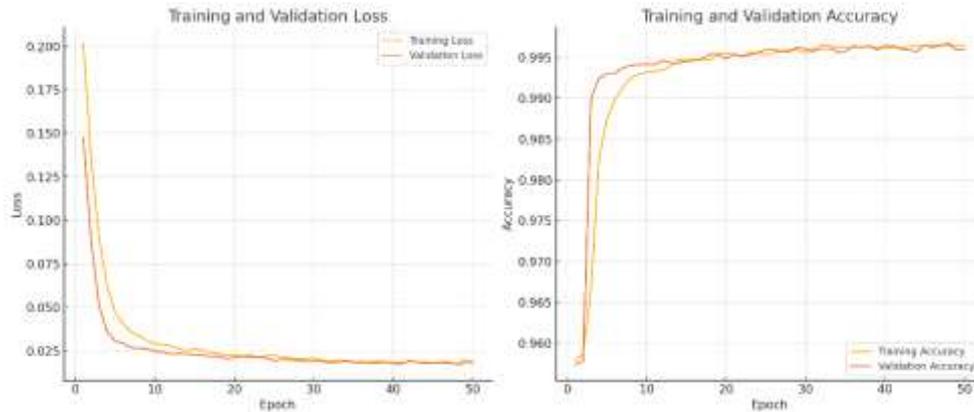


Fig.10: Training Results and Validation Model 1

Model 1 was trained using 25,972 training data and validated against 6,494 validation data. The training process was conducted over 50 epochs with a batch size of 128 and using a Tesla T4 GPU. The training results showed that the training accuracy increased consistently from 95.06% in the first epoch to around 99.47% in the 34th epoch, indicating a stable learning process. The validation accuracy also increased from 95.89% at the beginning of training to 99.66% at epoch 34, then remained stable at that range until the end of the epoch. This shows that the model is not only able to learn well from the training data but also generalizes well to the validation data.

The loss value on the training data decreased sharply from 0.3417 in the initial epoch to around 0.0254 in epoch 34, then leveled off in that range. The validation loss value also decreased significantly from 0.2045 to around 0.0187 to 0.0203, reflecting that overfitting did not occur significantly.

Model 1 shows excellent training performance, with high accuracy and low loss on both training and validation data. This indicates that the model does not experience significant underfitting or overfitting and is ready for use in the testing phase.

Model evaluation

Based on the training results of the three models with varying numbers of epochs, Model 1 was selected as the best model. Model testing was performed using the model quantized to int8 format, with the aim of enabling the model to run efficiently on micro devices such as ESP32. The model evaluation results are shown in Figure 11.



Fig. 11: Results from model testing

Based on the results of testing the developed classification model, excellent performance was achieved with an overall accuracy rate of 99.62%. This figure indicates that the model is capable of classifying data with a very low error rate and continues to perform very well even after being compressed to int8 format, making it highly reliable for activity detection tasks.

The classification metric evaluation shows that this model has an Area Under ROC Curve (AUC) value of 0.96. This indicates that the model has excellent ability to distinguish between classes, namely between daily activities, falls, and uncertain conditions. Precision 1.00 indicates that the model's predictions are highly accurate, without producing any false positives. Recall 1.00 indicates that the model is able to correctly recognize all actual data. F1 Score 1.00 is a balance between precision and recall, and this perfect score indicates excellent overall performance.

From the confusion matrix, it can be seen that the model can recognize daily activities (ADL) with high accuracy, with ADL classified correctly 100% of the time, without any misclassification as Fall or Uncertain. Fall data is recognized with 91.2% accuracy, but 8.8% is still misclassified as ADL. No samples were classified as Uncertain, so the model did not produce any unusual outputs in this case. The model demonstrated very high performance after quantization to the int8 format. This proves that the model does not experience significant performance degradation and is highly suitable for deployment on devices with limited resources.

Testing Results

After the fall detection device was successfully developed, it was tested to evaluate the system's performance in detecting falls and activities of daily living (ADL). In this test, two categories of activities were used, namely falls and ADL, with a total of 30 movement variations, consisting of 13 types of fall movements and 17 types of ADL activities. Each type of movement was performed five times per type of movement. The activities were performed by the user with the device attached to the waist, in accordance with the design of the fall detection system that had been created[14]. This kind of testing is also done by[15]. The results of the device testing for fall activities can be seen in **Table 4**.



Fig. 12: Code F01, F04 F05,F06



Fig. 13: Code F02



Fig. 14: Code F03,F07



Fig. 15: Code F08



Fig. 16: Code F09



Fig. 17: Code F10



Fig. 18: Code F11



Fig. 19: Code F12

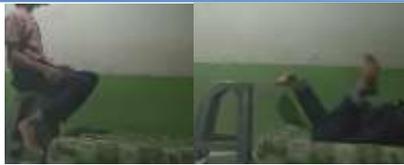


Fig. 20: Code F13



Fig. 21: Code F14

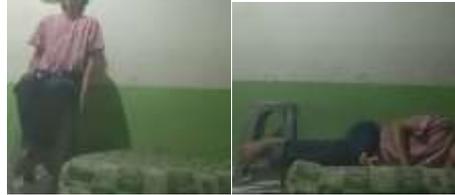


Fig. 22: Code F15

Table 4. Results of fall activity device testing

Code	Trials	Fall Notification		Code	Trials	Fall Notification	
		Yes	No			Yes	No
F01	5	5	0	F09	5	4	1
F02	5	5	0	F10	5	5	0
F03	5	5	0	F11	5	5	0
F04	5	5	0	F12	5	4	1
F05	5	5	0	F13	5	5	0
F06	5	5	0	F14	5	5	0
F07	5	4	1	F15	5	3	2
F08	5	5	0				

Based on the results of the Fall Activity Test in **Table 4** it was found that out of a total of 13 fall movements, almost all movements were detected successfully. Eleven out of 15 movements (F01–F06, F08, F10–F11, F13–F14) were detected perfectly (5/5) by the device. Three fall movements (F07, F09, F12) had one detection failure, resulting in only 4 or 5 fall notifications and one fall movement (F15) had two detection failures.

Next, testing was conducted for activities of daily living (ADL). The results of the device testing for falling activities can be seen in **Table 5**.



Fig. 23: Code D01, D02



Fig. 24: Code D03, D04



Fig. 25: Code D05, D06



Fig. 26: Code D07, D08



Fig. 27: Code D09, D10



Fig. 28: Code D11



Fig. 29: Code D12,D13



Fig. 30: Code D14



Fig. 31: Code D15



Fig. 32: Code D16



Fig. 33: Code D17



Fig. 34: Code D18

Table 5. Results of ADL device testing

Code	Trials	Fall Notification		Code	Trials	Fall Notification	
		Yes	No			Yes	No
D01	5	0	5	D10	5	0	5
D02	5	0	5	D11	5	0	5
D03	5	0	5	D12	5	1	4
D04	5	0	5	D13	5	1	4
D05	5	0	5	D14	5	1	4
D06	5	0	5	D15	5	0	5
D07	5	0	5	D16	5	0	5
D08	5	0	5	D17	5	0	5
D09	5	0	5	D18	5	0	5

Based on the results of activity testing for activities of daily living (ADL) consisting of 18 movements, each movement was also tested 5 times, resulting in a total of 90 trials. From all the experiments, there were two false positive notifications that appeared in movements D12 and D13, each occurring once. Thus, the system was able to distinguish normal activity from falling activity very well, although it was not completely free from false alarms.

From the results of the fall detection system testing conducted, the number of each performance evaluation parameter for classification is as shown in **Table 6** below.

Table 6. Results for each parameter

Parameter	Count
True Positive (TP)	71
False Positive (FP)	0
False Negative (FN)	4
True Negative (TN)	90

From the classification data, system performance analysis was conducted using three main evaluation metrics, namely accuracy, precision, sensitivity, and specificity.

$$\text{Accuracy} = \frac{70 + 88}{70 + 88 + 2 + 5} = 95.76\% \quad \text{Precision} = \frac{70}{70 + 2} = 97.22\%$$

$$\text{Sensitivity} = \frac{70}{70 + 5} = 93.33\% \quad \text{Specificity} = \frac{88}{88 + 2} = 97.78\%$$

The evaluation results show that the developed fall detection system performs very well. An accuracy value of 95.76% indicates that the system is able to classify activities accurately in general. A precision value of 97.22% shows that most of the fall notifications given are indeed fall activities, with only a few false alarms. The sensitivity value of 93.33% indicates that the system is very good at detecting fall activities, although there are still some cases that are not recognized as falls. Meanwhile, the specificity value of 97.78% shows that the system is also very good at recognizing ADL activities as non-falls.

4. Conclusion

The fall detection device was successfully created and functions properly. The feedforward neural network model demonstrated good performance with a model evaluation accuracy of 99.62% and an F1-score metric of 1.00, indicating that the model can classify both Fall and ADL classes equally well. Additionally, the model has been successfully deployed to a microcontroller, where testing results after deployment show an accuracy of 95.76%, precision of 97.22%, sensitivity of 93.33%, and specificity of 97.78% indicating that the device is highly reliable for detecting fall events and does not mistakenly classify normal activities as falls.

5. References

- [1] E. D. Noorratri, A. S. Mei Leni, and I. S. Kardi, "Deteksi Dini Resiko Jatuh Pada Lansia Di Posyandu Lansia Kentingan, Kecamatan Jebres, Surakarta," *GEMASSIKA J. Pengabd. Kpd. Masy.*, vol. 4, no. 2, p. 128, 2020, doi: 10.30787/gemassika.v4i2.636.
- [2] J. G. B. Talete, "Hubungan Dukungan Sosial Dengan Keaktifan Berorganisasi Pada Lansia di Jemaat Gmim Bait-Lahim Talete 1 Jofie Hilda Mandang Universitas Negeri Manado," vol. 9, no. 19, pp. 989–997, 2025.
- [3] S. R. B. A. Rudi, "Analisis Faktor Yang Mempengaruhi Risiko Jatuh Pada Lansia," *J. Ilm. Kesehat. Wawasan Kesehat.*, vol. Vol. 5, 2019.
- [4] H. Lilyanti, E. Indrawati, and A. Wamaulana, "Resiko Jatuh pada Lansia di Dusun Blendung Klari," *Indogenius*, vol. 1, no. 2, pp. 78–86, 2022, doi: 10.56359/igj.v1i2.67.
- [5] M. A. Yuzefo, F. Sabrian, and R. Novayelinda, "Hubungan Status Spritual Dengan Kualitas Hidup Pada lansia Di Kelurahan Tuah Karya," *Jom*, vol. 2, no. 2, pp. 1266–1274, 2016.
- [6] L. A. Sholekah, E. Soesanto, and S. Aisah, "Hubungan Faktor Fisiologis Pada Lansia Dengan Resiko Jatuh Di Dusun Wangil Desa Sambonganyar Kabupaten Blora," *J. Keperawatan dan Kesehat. Masy. Cendekia Utama*, vol. 11, no. 2, p. 174, 2022, doi: 10.31596/jcu.v11i2.1116.
- [7] M. Firmansyah, "Rancang Bangun Sistem Fall Detection Untuk Orang Lanjut Usia Berbasis Inertial Measurement Unit," *TEKTRIKA - J. Penelit. dan Pengemb. Telekomun. Kendali, Komputer, Elektr. dan Elektron.*, vol. 1, no. 2, pp. 133–137, 2019, doi: 10.25124/tektrika.v1i2.1745.
- [8] A. Darmawan and R. F. Hasani, "Rancang Bangun Sistem Monitoring Posisi dan Kondisi Jatuh Manusia Lanjut Usia (Manula) Berbasis Internet of Things (IoT)," *Spektral*, vol. 1, no. 1, pp. 10–16, 2020, doi: 10.32722/spektral.v1i1.3436.
- [9] A. Najmurokhman, Kusnandar, U. Komarudin, and A. Wibisono, "Development of Falling Notification System for Elderly Using MPU6050 Sensor and Short Message Service," *Proc. 2nd Int. Semin. Sci. Appl. Technol. (ISSAT 2021)*, vol. 207, no. Issat, pp. 345–351, 2021, doi:

- 10.2991/aer.k.211106.055.
- [10] R. Das, “Accelerated Safety: Revitalizing ADXL345 for Enhanced IoT-enabled Fall Detection,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 12, pp. 1117–1120, 2023, doi: 10.22214/ijraset.2023.57503.
- [11] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, “SisFall: A fall and movement dataset,” *Sensors (Switzerland)*, vol. 17, no. 1, 2017, doi: 10.3390/s17010198.
- [12] S. Campanella, A. Alnasef, L. Falaschetti, A. Belli, P. Pierleoni, and L. Palma, “A Novel Embedded Deep Learning Wearable Sensor for Fall Detection,” *IEEE Sens. J.*, vol. 24, no. 9, pp. 15219–15229, 2024, doi: 10.1109/JSEN.2024.3375603.
- [13] E. Schröter, D. Thanh Nghi, and A. Schneider, “Development of an Intelligent Walking Aid for Fall Detection,” *Curr. Dir. Biomed. Eng.*, vol. 9, no. 1, pp. 287–290, 2023, doi: 10.1515/cdbme-2023-1072.
- [14] I. Wayan Wiprayoga Wisesa, G. Mahardika, J. Terusan Ryacudu, W. Huwi, K. Jati Agung, and L. Selatan, “IoT-based Architecture for Automatic Detection of Fall Incident using Accelerometer Data,” *Indones. J. Artif. Intell. Data Min.*, vol. 3, no. 2, pp. 78–85, 2020.
- [15] S. Norhabibah, W. Andhyka, and D. Risqiwati, “Rancang Bangun Sistem Monitoring Deteksi Jatuh untuk Manula dengan Menggunakan Accelerometer,” *JOINCS (Journal Informatics, Network, Comput. Sci.)*, vol. 1, no. 1, p. 43, 2017, doi: 10.21070/joincs.v1i1.803.