applied sciences

MDPI

Article

Fall Detection Based on Recurrent Neural Networks and Accelerometer Data from Smartphones

Natalia Bartczak, Marta Glanowska, Karolina Kowalewicz, Maciej Kunin and Robert Susik *

Institute of Applied Computer Science, Lodz University of Technology, Żeromskiego 116, 90-924 Lodz, Poland
* Correspondence: rsusik@kis.p.lodz.pl

Abstract: An aging society increases the demand for solutions that enable quick reactions, such as calling for help in response to events that may threaten life or health. One of such events is a fall, which is a common cause (or consequence) of injuries among the elderly, that can lead to health problems or even death. Fall may be also a symptom of a serious health problem, such as a stroke or a heart attack. This study addresses the fall detection problem. We propose a fall detection solution based on accelerometer data from smartphone devices. The proposed model is based on a Recurrent Neural Network employing a Gated Recurrent Unit (GRU) layer. We compared the results with the state-of-the-art solutions available in the literature using the UniMiB SHAR dataset containing accelerometer data collected using smartphone devices. The dataset contains the validation dataset prepared for evaluation using the Leave-One-Subject-Out (LOSO-CV) and 5-Fold Cross-Validation (CV) strategies; consequently, we used them for evaluation. Our solution achieves the highest result for Leave-One-Subject-Out and a comparable result for the *k*-Fold Cross-Validation strategy, achieving 98.99% and 99.82% accuracy, respectively. We believe it has the potential for adoption in production devices, which could be helpful, for example, in nursing homes, improving the provision of assistance especially when combined into a multimodal system with other sensors. We also provide all the data and code used in our experiments publicly, allowing other researchers to reproduce our results.

Keywords: fall detection; accelerometer; cellphone; UniMiB SHAR; recurrent neural networks

check for updates

Academic Editors: Giorgio De Nunzio and Arkady Voloshin

Received: 13 March 2025
Revised: 30 May 2025
Accepted: 11 June 2025
Published: 14 June 2025

Citation: Bartczak, N.; Glanowska, M.; Kowalewicz, K.; Kunin, M.; Susik, R. Fall Detection Based on Recurrent Neural Networks and Accelerometer Data from Smartphones. *Appl. Sci.* **2025**, *15*, 6688. https://doi.org/10.3390/app15126688

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

1.1. Problem

The rapid aging of the global population has created a growing need for technologies that increase safety and improve the quality of life for older adults. Advances in artificial intelligence, combined with the widespread availability of devices such as accelerometers, have made it increasingly feasible to monitor human activity with high precision, enabling the prevention or mitigation of injuries that may pose serious risks to health or life. As the number of elderly individuals continues to rise [1], so does the demand for systems capable of preventing or minimizing the consequences of events that threaten well-being. Falls represent one of the most frequent and dangerous causes of injury in this population, often leading to fractures, long-term disability, or even death [2]. Furthermore, a fall may signal the onset of critical medical conditions such as strokes or cardiac events, which require immediate intervention. As a result, the development of accurate, reliable, and responsive fall detection systems has become an increasingly important area of research.

In this article, we address the problem of fall detection using accelerometer data. This area has attracted increasing interest in recent years, leading to the development of numerous methods and datasets related to human activity recognition. However, it is important to highlight that many of these studies focus predominantly on general activity classification, with fall detection receiving comparatively less attention. Consequently, the resulting models often suffer from limited accuracy or insufficient prediction speed for real-time applications. In response to these limitations, we propose a solution that achieves high accuracy and fast inference, making it suitable for practical deployment. In addition, we observed that authors often do not provide source codes, which hinders reproducibility and comparative evaluation. To promote transparency, we use the publicly available UniMiB-SHAR dataset [3] in our experiments, which contains accelerometer data collected from smartphones during various activities, including falls and activities of daily living (ADLs). We also share all the data and code used in our experiments publicly, enabling other researchers to reproduce our results and build upon our work.

1.2. Related Work

Existing fall detection systems leverage various technologies, including wearable devices, ambient sensors, and smartphones. Among these, smartphones (or rather their accelerometers and gyroscopes) are particularly promising due to their widespread availability. This has led to the exploration of various approaches for fall detection. Many traditional machine learning solutions, such as support vector machines (SVMs) [4] and *k*-Nearest Neighbors (*k*-NNs) [5], have been explored [6], but their performance is often limited by the need for handcrafted features and their inability to capture temporal dependencies in sequential data. Their research was based on the UR Fall Detection Dataset (URFD) [7], for which samples were recorded using an accelerometer hidden in two Microsoft Kinect cameras, which seems to be unpractical in real-life scenarios, in our opinion. In the course of the aforementioned studies, SVM gave better results. The authors of [8] addressed the fall detection problem using a comparable methodology. They constructed a dedicated dataset comprising six activities and 45 samples collected from three subjects. Several machine learning classifiers, including naive Bayes, random forest, and support vector machines (SVMs), were evaluated, with the SVM achieving the highest performance. Nevertheless, the dataset was limited in size and diversity.

In [3], the authors conducted a review of the most popular datasets on the performance of various activities available on the Internet. The postulate was made that signals should be collected using smartphones. They compared 13 datasets and found them insufficient to build an effective fall detection system. Only some of them contained information on both falls and activities of daily living (ADLs) that required movement. In addition, a few lacked details about study participants, such as gender, age, or height. The researchers decided to supplement the existing datasets with the missing cases by conducting controlled simulations of seven types of falls and nine types of movement-related activities. They collected data using an accelerometer inside a smartphone, placing it alternately in the participant's left and right pockets. The created UniMiB SHAR Dataset was shared publicly and has been widely used in subsequent research (including this paper). The highest accuracy score achieved in this study was 98.34%.

Later, T. Ivascu et al. [9] applied various machine learning techniques, including deep learning, for human activity recognition. The deep neural network implemented in their study outperformed traditional classifiers such as support vector machines (SVMs) and random forests (RFs) in terms of accuracy achieving 96.73% on the UniMiB SHAR dataset.

In 2018, Frédéric et al. [10] introduced an evaluation framework aimed at comparing different feature extraction methods. Their experiments, conducted on the Opportunity [11]

and UniMiB SHAR datasets, involved assessing multiple feature extraction approaches and comparing the performance of various machine learning algorithms, including Long Short-Term Memory (LSTM) networks. The solution achieved 74.66% on the UniMiB SHAR dataset.

Another LSTM-based work was presented by Boutellaa [12]. The author proposed the idea of building an autoencoder with LSTM layers achieving 98.17% accuracy on the UniMiB SHAR dataset. The model was trained using a combination of autoencoder as the feature extractor and SVM as the classifier.

In 2020, an interesting solution involving a layer-wise CNN (Convolutional Neural Network)-based model was presented [13]. In this paper, the local loss was applied to the Human Activity Recognition (HAR) problem achieving 98.82% accuracy.

Kanjilal and Uysal [14] used various classifier topologies to recognize human activities. The aim was to find out what affects the inaccurate learning of classification models by analyzing the architectures of deep neural networks. This approach gave the highest accuracy of 99.82% for the UniMiB SHAR dataset. Abidine et al. [15] proposed a method for fall detection based on dimensionality reduction using Kernel Discriminant Analysis (KDA) in combination with weighted support vector machines (WSVMs). This approach yielded an improvement of up to 5.6% in the F1-score compared to baseline models without feature reduction. The best performing model achieved an F1-score of 95.3% on the UniMiB-SHAR dataset.

Kaur and Sharma [16] conducted a comprehensive review of publicly available datasets related to fall detection. They provided an analysis of these datasets and subsequently evaluated the performance of five classical machine learning algorithms: *k*-Nearest Neighbors (*k*-NN), logistic regression, decision tree, naive Bayes classifier, and linear discriminant analysis. Their results indicated that *k*-NN was the most effective (about 95% F1-score) in distinguishing fall events from activities of daily living on a custom dataset. However, the authors did not provide results for the UniMiB-SHAR dataset specifically.

Mohammed et al. [17] addressed multiclass classification task using the UniMiB-SHAR dataset by introducing a novel architecture termed Multi-ResAtt. This model employs a multi-resolution attention mechanism to extract features from sensor data, which are then processed by a Recurrent Neural Network (RNN) for activity classification. They achieved 85.35% accuracy for the multiclass classification task. However, the authors did not provide the results for the binary classification task, which is the focus of our study.

In one of the most recent studies, Stampfler et al. [18] utilized the UniMiB-SHAR dataset to train an optimized ResNet-based architecture. Their model demonstrated high performance in the binary fall detection task (achieveing 99.87% accuracy).

Recent studies have explored deep learning-based approaches for fall detection using the UniMiB-SHAR dataset with remarkable success. Wang et al. [19] proposed a Patch-Transformer Network (PTN), a hybrid CNN-Transformer model, achieving an accuracy of 99.14% in binary fall detection. In 2024 [20] authors introduced the Parallel CNN-Transformer (PCNN-Transformer) architecture, which attained 98.68% accuracy on UniMiB-SHAR, along with similarly high performance across other benchmark datasets with 0.282 s average inference time. An interesting approach was presented in [21], where the authors combined information gathered from two sources: an accelerometer and WiFi signals. Unfortunately, the solution (source codes) was not shared publicly like in the other studies.

1.3. Aim

In this study, we propose two deep learning models based on Recurrent Neural Networks (RNNs) with Gated Recurrent Units (GRUs) for the task of fall detection. Unlike many existing studies that focus on general human activity recognition, our objective is

to develop a solution explicitly dedicated to fall detection, which can be applied in real-world scenarios to protect elderly individuals and prevent health-related complications. The proposed models are evaluated on the UniMiB-SHAR dataset using the Leave-One-Subject-Out Cross-Validation (LOSO-CV) and *k*-Fold Cross-Validation (CV). The results demonstrate that our approach achieves the highest accuracy among the methods reported in the literature for this dataset under the LOSO-CV setting. Moreover, the proposed models exhibit low inference times, making them particularly suitable for deployment in real-time monitoring systems.

The main contributions of this work are as follows:

(i) We introduce an RNN-based architecture specifically designed for fall detection, which achieves the highest accuracy than previously reported methods on the UniMiB-SHAR dataset, and publicly share all the data and code used in the experiments.
(ii) We demonstrate the effectiveness of the presented solution in terms of inference time, which is crucial for real-time applications.
(iii) We discuss potential future extensions, including the integration of multimodal sensor inputs and adaptation of the method to real-world deployment scenarios aimed at health monitoring and early intervention.

By addressing the challenges of fall detection, our study contributes to the development of practical and deployable systems that can improve safety and quality of life, particularly for vulnerable populations. The proposed solution achieves the highest accuracy and the lowest inference time, making it suitable for real-time applications.

2. Materials and Methods

2.1. Dataset

In our study, we employed the UniMiB-SHAR dataset [3], a publicly available and widely used benchmark for human activity and fall detection research. This dataset comprises accelerometer data collected from smartphones carried by participants while performing a variety of activities, including both falls and activities of daily living (ADLs). Each observation contains tri-axial acceleration readings measured along the X, Y, and Z axes with corresponding metadata such as the activity label, participant identifier, and trial number.

The dataset includes data from 30 people (of 24 women and 6 men), aged between 18 and 60 years. In total, it contains 7013 labeled samples. The original annotation defines 17 distinct activity classes, encompassing 8 types of falls (e.g., falling leftward, generic backward falls, falls with protective strategy, and syncope) and 9 types of routine activities (e.g., walking, running, jumping, sitting down, and going up or down the stairs).

The relatively diverse participant pool and rich annotation, makes UniMiB-SHAR a valuable resource for developing and evaluating machine learning models aimed specifically at fall detection and activity recognition. Its inclusion of multiple fall types provides an especially relevant testbed for algorithms intended to distinguish dangerous events from benign motions in real-world conditions.

In this paper, we focused on a binary problem to distinguish falls from ADLs. Figure 1 shows sample smartphones' accelerometer data for eight different types of falls and, additionally, a sitting-down activity. It can be noticed that there is only a slight difference between those activities, which is not obvious at first glance. The subtle variations in accelerometer readings caused by different user activities present a formidable obstacle to achieving robust and precise classification.

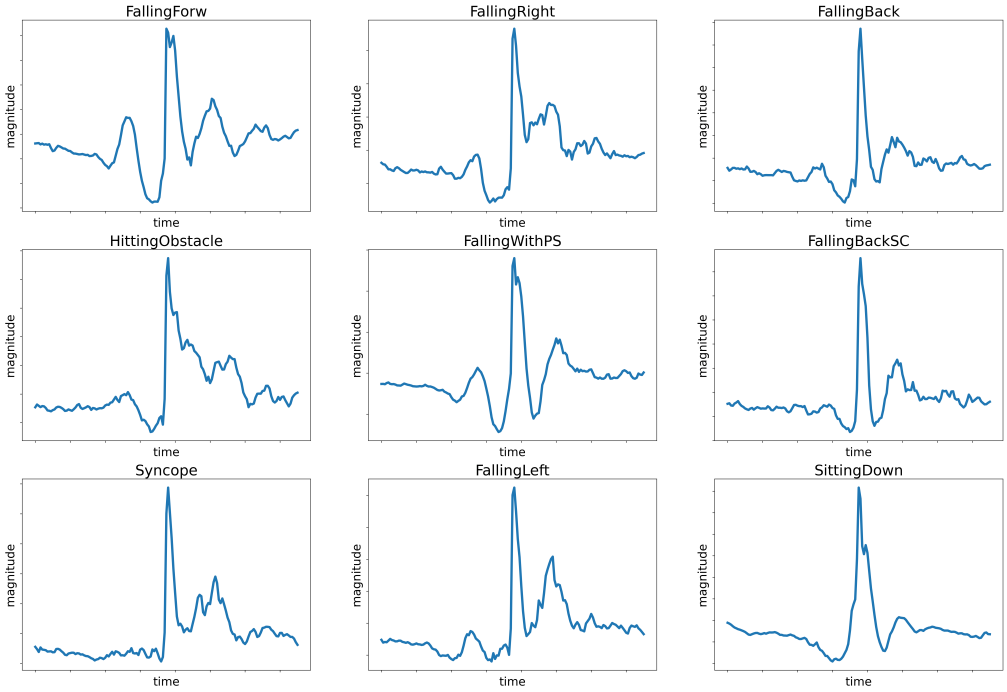


Figure 1. Sample smartphones' accelerometer data for eight different types of falls and sitting down as a reference.

2.2. Methodology

We used the same data to train and test the solution as in the reference [3] article to ensure that our results were comparable to the competing solution. Accordingly, two approaches were applied to divide the data. The first was the k-Fold Cross-Validation method with $k = 5$, and the second was the Leave-One-Subject-Out validation. After prior shuffling, we split the data into 5 and 30 folds, respectively. In the first method, 5 groups of equal size are created (independently of the subjects' identifiers). Leave-One-Subject-Out involves the establishment of samples about one subject as a test set and the remaining subjects as a training set, which provides a global approach to the problem. This way, we were able to reproduce the results from the reference article.

2.3. Metrics

To evaluate and compare the performance of our proposed models with existing solutions in the literature, we primarily used the accuracy metric (Equation (1)), as it is the most commonly reported measure in fall detection studies. Accuracy represents the overall proportion of correctly classified instances, including both falls and non falls.

However, since datasets in fall detection are often imbalanced, we also report three additional classification metrics: precision, recall, and the F1-score. These metrics provide a detailed understanding of model performance, particularly in distinguishing fall events from normal daily activities.

- Precision (Equation (2)) measures the proportion of correctly predicted fall events among all instances predicted as falls.
- Recall (Equation (3)) quantifies the proportion of actual fall events that were correctly identified.
- F1-score (Equation (4)) is the harmonic mean of precision and recall, offering a balanced metric especially useful in the presence of class imbalance.

The formulas for these metrics are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

where

- TP—true positives (correctly detected fall events);
- TN—true negatives (correctly detected non-fall events);
- FP—false positives (non-fall events incorrectly classified as falls);
- FN—false negatives (missed fall events).

These metrics provide a comprehensive evaluation of the model's effectiveness in real-world fall detection scenarios, where both sensitivity (recall) and specificity (precision) are essential.

2.4. Evaluation Strategy

To ensure the reliability of our results, we trained and evaluated the models using the k-Fold Cross-Validation (CV) strategy. In this approach, the dataset is split into k equally sized subsets (folds). That means, for each of the k iterations, one fold is reserved as the test set, while the remaining $k - 1$ folds are used for training. As a result, each sample in the dataset is used exactly once for validation and $k - 1$ times for training.

We evaluated our models using two widely adopted cross-validation schemes:

- k-Fold Cross-Validation with $k = 5$, which offers a good trade-off between computational efficiency and statistical robustness;
- Leave-One-Subject-Out Cross-Validation (LOSO-CV), corresponding to $k = 30$, where each subject is left out once for testing while the model is trained on the data of the remaining 29 participants.

This evaluation strategy was also chosen to ensure the compatibility of our results with those reported in other studies, particularly those based on the UniMiB-SHAR dataset, where LOSO-CV is commonly used as a benchmark protocol.

Each model was trained for 20 epochs using the Adam optimizer, with a learning rate of 0.0003 and a batch size of 16 observations. To assess performance, the final accuracy of a given model was calculated as the average of the accuracy scores obtained across all k folds.

2.5. Our Approach

We propose two deep-learning models based on Recurrent Neural Networks (RNNs) [22]. RNNs find application in signal processing due to the ability to model sequential data and capture temporal dependencies. This kind of network can effectively denoise and filter signals, making them more appropriate for further processing (i.e., classification task), and is an active subject of research [23–25]. The problems related to this type of network are exploding and vanishing gradients [26,27]. Thus, we used the Gated Recurrent Unit (GRU) mechanism [28] in our solutions to avoid such an issue.

Figure 2a illustrates the described neural network. The input to the model are 453 values of the accelerometer signal (151 values for each axis: X, Y, and Z), sampled at

50 Hz. The architecture consists of a GRU layer with an output of 512 and a hyperbolic tangent as an activation function. It is followed by a batch normalization layer and the transition to a dense layer with 128 neurons and activation function ReLU ([29]). In the next step, a dropout layer with a rejection rate of 30% and a dense layer with a single neuron with a sigmoidal activation function at the output.

The second proposed model is presented in Figure 2b. This model's architecture and its input shape are similar to the previous with the modification of the GRU layer to the bidirectional. The other layers were constructed in the same way as in the first model.

For both presented models, we used the same binary cross entropy loss function. We chose the Adam optimizer for training because of its adaptive learning rates, resulting in accelerated training.

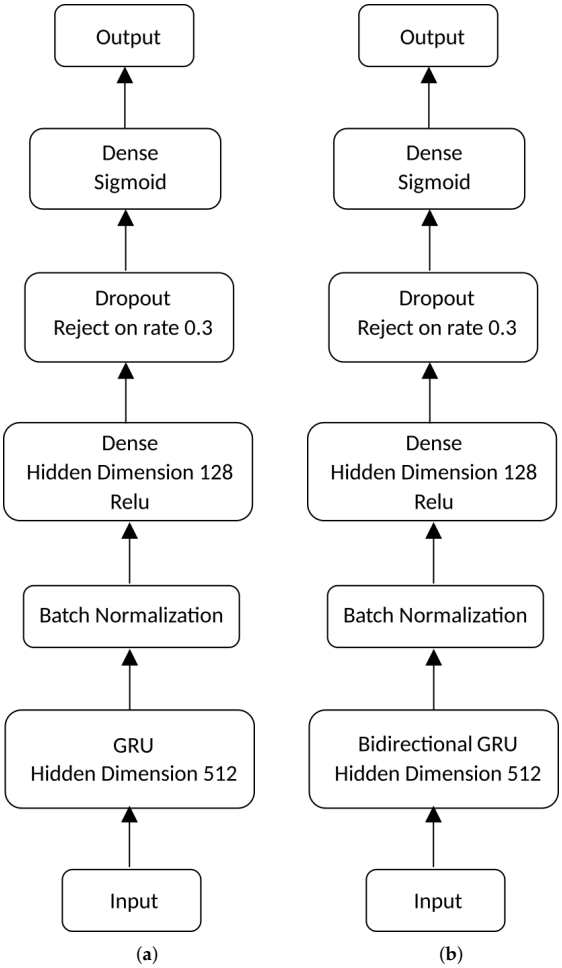


Figure 2. The architecture of proposed models (for optimized hyperparameters, see Table 1). (**a**) Recurrent Neural Network-based model (RNN). (**b**) Bidirectional Recurrent Neural Network-based model (BRNN).

Table 1. Hyperparameter search space and the best values found in the study.

Parameter	Range/Values	Step/Sampling	Best Value
`hidden_dim`	64–1024	64	768
`dense_units`	32–256	32	64
`learning_rate`	10^{-5}–10^{-3}	log	0.000347012
`dropout_rate`	0.1–0.5	0.1	0.3
l_1	10^{-6}–10^{-4}	log	1.7089×10^{-6}
l_2	10^{-6}–10^{-3}	log	6.3877×10^{-6}

2.6. Hyperparameter Optimization

Finally, we applied hyperparameter optimization to the BRNN model using the Keras Tuner framework (v. 1.4.7) [30]. Specifically, we employed the RandomSearch strategy, conducting 10 trials to explore the hyperparameter space defined in Table 1. This procedure allowed us to identify the optimal configuration of model hyperparameters which is presented in the last column. There might be still some room for improvement, but the process is time-consuming and requires a lot of computational resources.

2.7. Aparatus

The experiments were run on a machine equipped with an 11th Gen Intel® Core™ i7 processor clocked at 2.30 GHz with a 24 MB Smart Cache. The operating system was Windows 11 Home 64 64-bit. The computer had 32 GB of RAM clocked at 3200 MHz. The training was run on an NVIDIA GeForce RTX 3050 graphic card with 2560 CUDA cores, 4 GiB of RAM, a 128-bit memory bus, 1.55 GHz base frequency, and 1.76 GHz boost frequency). The proposed models were implemented in Python 3.9.16 and Tensorflow 2.10.1. All the source codes are shared publicly and available on the GitHub platform at https://github.com/rsusik/fall_detection_rnn (accessed on 12 March 2025).

3. Results

In this section, we first present the training and inference time analysis of the models. We begin with this analysis to provide a context for our decision to opt for the GRU architecture. Subsequently, we proceed to analyze the accuracy scores of the solutions available in the literature and compare their results to our approach.

Initially, we performed tests to measure the training time for an alternate approach with the LSTM layer in our models. We measured training and inference times for RNN- and BRNN-based models using GRU and LSTM layers to compare the computational efficiency of different recurrent architectures. The RNN model with a GRU layer required approximately 24 s per training epoch, while the bidirectional GRU variant (BRNN) increased this to 47 s. When replacing GRU with LSTM, training times increased notably, 33 s for a standard RNN and 63 s for the BRNN. These results confirm that GRU-based models are significantly more efficient during training, which is particularly beneficial for scenarios involving limited computational resources or frequent retraining.

In addition to training time, we also evaluated inference performance of the best performing model. The optimized BRNN model achieved a total inference time of 5.98 s on the entire test set (batch size = 32), corresponding to an average of 74 ms per batch and 2.31 ms per individual sample. When considering the case of processing a 1 s frame of accelerometer data (comprising 50 samples at a 50 Hz sampling rate), the average inference time per window was just 0.77 ms.

These results indicate that the model can produce predictions with an average inference time of 2.31 ms per sample, which gives 0.77 ms on average per 1 s of data. This is well within the constraints required for real-time applications. It demonstrates the potential for deploying the proposed solution on low latency platforms, such as mobile or wearable devices, for continuous health monitoring and fall detection.

Our goal was to compare these results with those reported in related works. However, we encountered a significant limitation as most publications do not specify the exact context in which their inference time measurements were obtained. For instance, in the study by Alqaness et al. [20], an average inference time of 282 ms is reported for the PCNN-Transformer model. Unfortunately, the authors do not clarify whether inference times corresponds to the time required for processing a single sample, a batch of samples, or the entire test set. This causes ambiguity and makes direct comparisons challenging and potentially

misleading. A similar issue arises in the work of Wang and Wu [19], where inference times are reported for models trained on input windows of different lengths of 1 s, 2 s, and 3 s. The corresponding inference times are 1.36 ms, 1.37 ms, and 1.43 ms, respectively. Although they report varying accuracy scores depending on the window size, it remains unclear how these inference times were measured, whether in a batched or unbatched setting, and whether additional preprocessing steps were included. Due to the lack of common process of inference time measurement across studies, we emphasize that our reported times were obtained in a controlled and repeatable environment, with details specified to facilitate reproducibility.

The optimized BRNN model achieved a high average accuracy of 99.82%, which means that nearly all samples were correctly classified. Additionally, the F1-score, that balances precision and recall, reached a very close value of 99.80%, reflecting the model's ability to maintain both a low false positive and false negative rates. The precision metric, with an average of 99.82%, confirms that most fall predictions were correct, while the recall of 99.78% suggests that almost all actual falls were successfully detected. These consistently high scores across all evaluation metrics demonstrate the stability of the proposed model in handling the fall detection. Given this, the model shows potential for deployment in real world applications, where reliability and accuracy are essential to ensure trustworthy alerts in critical situations on time.

Table 2 presents the results of related solutions available in the literature (including ours). Each record represents an algorithm proposed by the authors to solve the problem. The table includes information about the paper, publication year, validation method, the algorithm used in the solution, and the accuracy score. Unfortunately, not all authors share the source codes. Thus, we are not able to reproduce all the results. Additionally, some solutions use different approaches to validate the model. For instance, Boutellaa [12] uses a simple train/test split, where the test is a 20% subset of randomly selected records. On the other hand, Kanjilal et al. [14] performs random subsampling cross-validation. Assuming the approach of the original paper [3] we can divide the results into two categories: those that report the average accuracy score of *k*-Fold Cross-Validation and those that use Leave-One-Subject-Out validation. The article introducing the dataset [3] achieved the highest results using the 5-Fold Cross-Validation method, achieving a 98.57% accuracy score. Another approach using this validation method was proposed by Ivascu et al. [9]. They used RF, SVM, and DNN algorithms, of which the highest score of 96.73% was reached by DNN. Boutellaa [12], using a different method of data division and an autoencoder as a feature extractor, achieved a result of 98.17%. Kanjilal et al. [14] improved the results, splitting the data using random sub-sampling for Cross-Validation, regardless of the adopted network architecture, meaning the number of hidden layers added. The ResNet-based model by Stampfler et al. [18] achieved the highest result for 5-Fold Cross-Validation among competitors (0.05 percentage points more than our model). On the other hand, our model dominated the competitive solutions (0.51 percentage points) for the Leave-One-Subject-Out strategy.

Figure 3 presents the accuracy score for solutions that used the Leave-One-Subject-Out validation strategy. We can clearly see that our solution outperforms competitors in fall detection, achieving a 98.99% accuracy score.

However, the results differ when the models are evaluated using the *k*-Fold Cross-Validation strategy. Figure 4 presents the accuracy scores of the algorithms under this evaluation protocol. Notably, the method proposed by Stampfler et al. [18] achieves the highest accuracy for 5-Fold Cross-Validation, with a score that is nearly identical to ours (the difference amounts to only 0.05%). This marginal gap suggests that the potential for further improvement in terms of classification accuracy may be limited. Consequently,

future research efforts might be better directed toward optimizing other aspects of fall detection models, such as inference time or computational efficiency, particularly in the context of real-time applications.

Table 2. Accuracy of fall detection models reported in related studies and in our approach.

Authors	Year	Evaluation Method	Algorithm	Accuracy (%)
Micucci et al. [3]	2017	5-Fold Leave-One-Subject-Out	ANN	98.57 95.41
Ivascu et al. [9]	2017	5-Fold	DNN RF SVM	96.73 94.00 92.81
Boutellaa [12]	2019	Train/test by subjects 20% train as validation	LSTM-AE + SVM	98.17
Kanjilal et al. [14]	2021	Random sub-sampling for cross-validation	1D CNN RNN – LSTM ANN	99.82 99.77 99.52
Stampfler et al. [18]	2023	5-Fold Leave-One-Subject-Out	Optimized ResNet	99.87 98.48
Wang and Wu [19]	2023	80/20 train/test split	Patch-Transformer	99.14
Al-qaness et al. [20]	2024	70/30 train/test split 10-Fold	PCNN-Transformer	98.30 98.68
Our approach	2025	5-Fold Leave-One-Subject-Out	BRNN (Optimized) BRNN RNN BRNN RNN	99.82 99.70 99.69 98.99 98.66

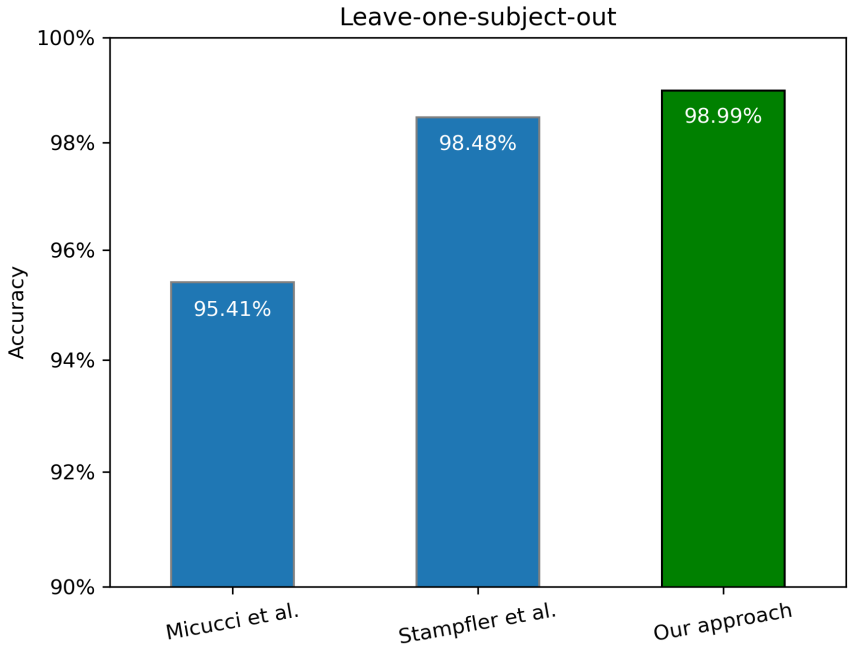


Figure 3. Fall classification accuracy scores for Leave-One-Subject-Out Cross-Validation [3,18].

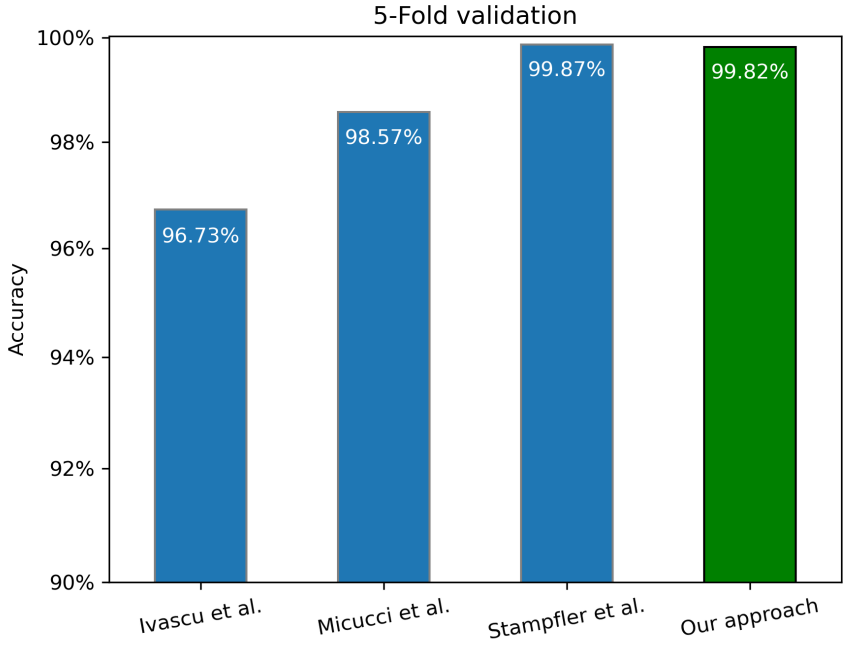


Figure 4. Fall classification accuracy scores for *k*-Fold Cross-Validation [3,9,18].

4. Discussion

The proposed RNN-based models, employing GRU and bidirectional GRU layers, achieved satisfactory results in the task of fall detection using the UniMiB-SHAR dataset. Specifically, the BRNN model reached an accuracy of 98.99% using the Leave-One-Subject-Out (LOSO) Cross-Validation protocol and 99.82% using 5-Fold Cross-Validation. These results are highly competitive when compared to existing methods reported in the literature. For instance, Al-qaness et al. [20] reported a maximum accuracy of 98.68% using the PCNN-Transformer model, while Wang and Wu [19] achieved 99.14% with their Patch-Transformer Network. Although Kanjilal et al. [14] reported slightly higher scores (up to 99.82%) using different architectures, the validation strategy was less standardized, making direct comparison difficult. In contrast, our models follow widely accepted evaluation protocols and achieve comparable or better performance.

The optimized BRNN model processes a single data sample in approximately 2.31 ms which results in an average processing time of 0.77 ms for data corresponding to 1 s. However, these results are difficult to compare directly with those reported by other authors. For example, Al-qaness et al. [20] reported an average inference time of 282 ms, though the precise measurement context (single sample, batch, or full set) was not specified. Similarly, Wang and Wu [19] reported times in the range of 1.36 ms to 1.43 ms for different window sizes (1–3 s), but without details on the computational setup or batching. Our measurements were conducted in a transparent and repeatable environment, ensuring the reliability of the reported latency. The ability to generate predictions in real time makes the proposed model particularly suitable for mobile or wearable deployment.

These findings have important implications for health monitoring applications, especially in elderly care. Falls are a leading cause of injury and hospitalization among older adults, and timely detection is critical for reducing health risks and improving response times. The high accuracy and real-time performance of our models demonstrate their potential for practical integration into assistive technologies. Unlike many existing studies that focus on general activity recognition, our work is tailored specifically to fall detection,

with the goal of contributing directly to systems aimed at enhancing safety and autonomy for older individuals.

Despite the promising results, this study has some limitations. The experiments were conducted exclusively on the UniMiB-SHAR dataset, which, although widely used, may not fully represent the variability found in real-world. Future research should include evaluations across multiple datasets (e.g., SisFall, MobiAct) to better assess the generalizability of the models. Additionally, fall detection alone may not provide sufficient context to assess a person's health status. Future systems could integrate additional data sources such as ECG, heart rate, or blood pressure to improve diagnostic accuracy. Recent studies have demonstrated the value of multimodal sensing for detecting critical events, and this direction appears promising for building more comprehensive health monitoring solutions.

5. Conclusions

In this study, we proposed a fall detection solution based on Recurrent Neural Networks, specifically utilizing bidirectional GRU layers. The developed models were rigorously evaluated using two commonly applied validation strategies: Leave-One-Subject-Out and 5-Fold Cross-Validation. The experimental results demonstrate that the proposed approach achieves better results for LOSO compared to existing methods reported in the literature attaining an accuracy of 98.99% and close to the best results for 5-Fold Cross-Validation achieving 99.82%.

While the UniMiB-SHAR dataset has become a widely adopted benchmark in fall detection research, it was originally intended to complement other representative datasets in the field. As such, one potential direction for future work is to evaluate the model in real-world scenarios, where the characteristics of actual falls may differ from the simulated falls present in available datasets.

Furthermore, although fall detection is a crucial step toward enhancing the safety of vulnerable individuals, particularly the elderly, it does not provide a complete picture of a person's health status. Falls may not always indicate a critical condition but can serve as early warnings of underlying medical issues. Therefore, future research could focus on the integration of additional physiological signals such as electrocardiogram (ECG), heart rate, or blood pressure to improve the reliability and clinical relevance of automated health monitoring systems.

Such multimodal solutions could be particularly valuable in care environments, such as nursing homes or assisted living facilities, where early detection and real-time alerting of potential health threats could significantly enhance the responsiveness and quality of care provided to residents.

Author Contributions: Conceptualization, R.S.; methodology, N.B., M.G., K.K., M.K. and R.S.; validation, N.B., M.G., K.K. and R.S.; writing, N.B., M.G. and K.K.; writing—review and editing, R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated during and/or analyzed during the current study are available in the UniMiB SHAR Dataset repository, http://www.sal.disco.unimib.it/technologies/unimib-shar/ (accessed on 12 March 2025). Source codes are available on the GitHub platform at https://github.com/rsusik/fall_detection_rnn (accessed on 12 March 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Population Division, Department of Economic and Social Affairs, United Nations. *World Population Ageing Report (ST/ESA/SER.A/390)*; United Nations: New York, NY, USA, 2015.
2. Tinetti, M.; Liu, W.; Claus, E. Predictors and prognosis of inability to get up after falls among elderly persons. *JAMA* **1993**, *269*, 65–70. [CrossRef] [PubMed]
3. Micucci, D.; Mobilio, M.; Napoletano, P. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Appl. Sci.* **2017**, *7*, 1101. [CrossRef]
4. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
5. Fix, E.; Hodges, J.L., Jr. Discriminatory analysis-nonparametric discrimination: Small sample performance. In *Technical Report*; California University Berkeley: Berkeley, CA, USA, 1952.
6. Qian, H.; Fan, X. Fall Detection for the Elderly by Accelerometers. *Int. J. Inf. Technol.* **2003**, *9*, 1–12.
7. Kwolek, B.; Kepski, M. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Comput. Methods Programs Biomed.* **2014**, *117*, 489–501. [CrossRef]
8. Luštrek, M.; Kaluža, B. Fall detection and activity recognition with machine learning. *Informatica* **2009**, *33*, 197–204.
9. Ivascu, T.; Cincar, K.; Dinis, A.; Negru, V. Activities of daily living and falls recognition and classification from the wearable sensors data. In Proceedings of the 2017 E-Health and Bioengineering Conference (EHB), Sinaia, Romania, 22–24 June 2017; pp. 627–630. [CrossRef]
10. Li, F.; Shirahama, K.; Nisar, M.A.; Köping, L.; Grzegorzek, M. Comparison of Feature Learning Methods for Human Activity Recognition Using Wearable Sensors. *Sensors* **2018**, *18*, 679. [CrossRef]
11. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
12. Boutellaa, E. Detecting Falls with Recurrent Autoencoders and Body Acceleration Data. In Proceedings of the 6th International Conference on Image and Signal Processing and their Applications (ISPA), Mostaganem, Algeria, 24–25 November 2019; pp. 1–5. [CrossRef]
13. Teng, Q.; Wang, K.; Zhang, L.; He, J. The layer-wise training convolutional neural networks using local loss for sensor-based human activity recognition. *IEEE Sens. J.* **2020**, *20*, 7265–7274. [CrossRef]
14. Kanjilal, R.; Uysal, I. The future of human activity recognition: Deep learning or feature engineering? *Neural Process. Lett.* **2021**, *53*, 561–579. [CrossRef]
15. Abidine, M.; Oussalah, M.; Fergani, B.; Lounis, H. Activity recognition on smartphones using an AKNN based support vectors. *Sens. Rev.* **2022**, *42*, 384–401. [CrossRef]
16. Kaur, R.; Sharma, R. Analyzing Wearables Dataset to Predict ADLs and Falls: A Pilot Study. *arXiv* **2022**, arXiv:2209.04785.
17. Al-qaness, M.A.A.; Dahou, A.; Elaziz, M.A.; Helmi, A.M. Multi-ResAtt: Multilevel Residual Network With Attention for Human Activity Recognition Using Wearable Sensors. *IEEE Trans. Ind. Inform.* **2023**, *19*, 144–152. [CrossRef]
18. Stampfler, T.; Elgendi, M.; Fletcher, R.R.; Menon, C. The use of deep learning for smartphone-based human activity recognition. *Front. Public Health* **2023**, *11*, 1086671. [CrossRef] [PubMed]
19. Wang, S.; Wu, J. Patch-Transformer Network: A Wearable-Sensor-Based Fall Detection Method. *Sensors* **2023**, *23*, 6360. [CrossRef]
20. Al-qaness, M.A.; Dahou, A.; Abd Elaziz, M.; Helmi, A.M. Human activity recognition and fall detection using convolutional neural network and transformer-based architecture. *Biomed. Signal Process. Control* **2024**, *95*, 106412. [CrossRef]
21. Wang, L.; Su, D.; Zhang, A.; Zhu, Y.; Jiang, W.; He, X.; Yang, P. Real-Time Fall Detection Using Smartphone Accelerometers and WiFi Channel State Information. *IEEE Sens. J.* 2025, *Early Access. [CrossRef]*
22. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
23. Yang, Y.; Sautière, G.; Ryu, J.J.; Cohen, T.S. Feedback Recurrent AutoEncoder. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 3347–3351.
24. Susik, R. Recurrent Autoencoder with Sequence-Aware Encoding. In *Computational Science—ICCS 2021*; Springer: Cham, Switzerland, 2021; pp. 47–57.
25. Eltouny, K.A.; Liang, X. Large-scale structural health monitoring using composite recurrent neural networks and grid environments. *Comput.-Aided Civ. Infrastruct. Eng.* **2023**, *38*, 271–287. [CrossRef]
26. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef]
27. Doya, K. Bifurcations in the learning of recurrent neural networks. In Proceedings of the 1992 IEEE International Symposium on Circuits and Systems, San Diego, CA, USA, 10–13 May 1992; Volume 6, pp. 2777–2780. [CrossRef]
28. Cho, K.; van Merrienboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv* **2014**, arXiv:1409.1259.

29. Fukushima, K. Cognitron: A self-organizing multilayered neural network. *Biol. Cybern.* **1975**, *20*, 121–136. [CrossRef] [PubMed]
30. KerasTuner. 2019. Available online: https://github.com/keras-team/keras-tuner (accessed on 12 March 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.