information

MDPI

*Communication*

# Fall Detection with CNN-Casual LSTM Network

**Jiang Wu, Jiale Wang *, Ao Zhan and Chengyu Wu**

Department of Communication Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China; wujiang@zstu.edu.cn (J.W.); zhanao1983@zstu.edu.cn (A.Z.); jerry916@zstu.edu.cn (C.W.)
* Correspondence: 201930605048@mails.zstu.edu.cn

**Abstract:** Falls are one of the main causes of elderly injuries. If the faller can be found in time, further injury can be effectively avoided. In order to protect personal privacy and improve the accuracy of fall detection, this paper proposes a fall detection algorithm using the CNN-Casual LSTM network based on three-axis acceleration and three-axis rotation angular velocity sensors. The neural network in this system includes an encoding layer, a decoding layer, and a ResNet18 classifier. Furthermore, the encoding layer includes three layers of CNN and three layers of Casual LSTM. The decoding layer includes three layers of deconvolution and three layers of Casual LSTM. The decoding layer maps spatio-temporal information to a hidden variable output that is more conducive relative to the work of the classification network, which is classified by ResNet18. Moreover, we used the public data set SisFall to evaluate the performance of the algorithm. The results of the experiments show that the algorithm has high accuracy up to 99.79%.

**Keywords:** fall detection; deep learning; spatial feature extraction; CNN encoder; CNN decoder

check for updates

**Citation:** Wu, J.; Wang, J.; Zhan, A.; Wu, C. Fall Detection with CNN-Casual LSTM Network. *Information* **2021**, *12*, 403. https://doi.org/10.3390/info12100403

Academic Editors: Neil Gordon and Han Reichgelt

Received: 14 August 2021
Accepted: 27 September 2021
Published: 29 September 2021

## 1. Introduction

Falls are one of the main causes of elderly injuries [1]. According to the World Health Organization (WHO), about 30% of people over 65 fall every year, causing more than 300,000 deaths [2]. The injuries to the elderly caused by falls not only cause harm physically but also mentally [3]. Furthermore, medical analyses have demonstrated that the injuries caused by falls are highly dependent on the response and rescue time [4]. Therefore, for the elderly, it is of great significance to protect their health by calling the police and the hospital in time when they fall.

In order to eliminate the risk of injury caused by falls for the elderly, researchers have developed different protection measures from Fall Detection to Fall Prevention [5]. The main goal of fall detection systems is to distinguish between fall events and activities of daily living (ADLs) [6]. Some ADLs, such as moving from standing or sitting position to lying position, are similar to falls. Therefore, it is tough to develop a method that identifies them accurately. In addition to the accuracy and timeliness of the fall detection algorithm, the most important issue is user convenience and privacy.

In this research, from the perspective of versatility and privacy, a fall detection method is proposed based on a three-axis acceleration and three-axis rotation angular velocity sensor that preserves privacy. The algorithm is based on a CNN-Casual LSTM, which includes an encoding layer, a decoding layer, and a ResNet18 classifier [7].

The organization of this paper is as follows. In Section 2, we discuss the existing methods for fall detection using wearable devices, with a focus on deep learning research. In Section 3, we introduce the details of the collected data set, data preprocessing steps, and CNN-Casual LSTM. The evaluation settings and results are presented and discussed in Section 4. The conclusion of this article is given in Section 5.

## 2. Related Work

Nowadays, existing fall detection technology can be roughly divided into three categories [4]: vision-based sensors [8,9], ambient sensors [10,11], and wearable sensors [12,13].

Vision-based sensors obtain motion information by monitoring equipment and extracting human body image inclination [14] or human bone annotations from the obtained video or picture information [15] to detect whether a fall has occurred. Subhash Chand Agrawal [16] used the improved GMM to perform background subtraction to find the foreground object and judges the fall by calculating the distance between the top and the middle center of the rectangle covering the human body. The user does not need to wear extra equipment. However, it is easily blocked and invades the privacy of the subjects. In order to solve this difficulty, Xiangbo Kong [17] used a depth camera to obtain the skeleton images of a person who is standing or falling down, and he used FFT to encrypt images and to detect the fall. Ambient sensors often detect falls by collecting infrared [18], radar [19], or other signals from the scene sensor. Tao Yang [20] used radars placed in three fixed positions to measure data, and he used time-frequency distribution and range-time intensity as input data for feature extraction to detect falls. Although it does not cause privacy invasion issues, it comes with a high cost. It is vulnerable to noise, and the detection range is relatively limited. Wearable sensors rely on various low-cost sensors to detect falls [21]. Its detection capabilities rely on real-time wear of the sensor, but the elderly may not be able to wear them in some cases, such as taking a bath. In addition, personal wear may cause discomfort for some elderly people.

In recent years, due to the low cost of sensors, wearable sensors have attracted more and more attention. The most commonly used parts of wearable sensors are the calf, spine, head, pelvis, and feet [22] to obtain the three-axis acceleration at different positions and the three-axis rotation angular velocity in a gyroscope. Kazi Md. Shahiduzzaman [23] used smart helmets to integrate wearable cameras, accelerometers, and gyroscope sensors and processed multi-sensor collaboration data on the edge. Seyed Amirhossein Mousavi [24] proposed a method of using smartphones and acceleration signals to detect falls by using smartphone sensors and reporting the person's position, with an accuracy rate of 96.33%. Kimaya Desai [12] performed human fall detection by deploying a simple 32-bit microcontroller on the wearable belt. Threshold methods and machine classification algorithms are widely used in wearable devices for fall detection [25]. Threshold methods are roughly divided into static threshold methods [26] and dynamic threshold methods [27]. When the threshold is passed (either upwardly or downwardly), it is judged as a fall. Machine learning methods are mainly divided into traditional pattern recognition and deep learning-based classification and recognition [5]. Traditional recognition algorithms (such as support vector machine [26], K-nearest neighbor algorithm [28], etc.) all rely on manual feature extraction for recognition. Therefore, higher requirements for fall detection are needed for researchers. First of all, it is necessary to find the parts of the human body involved in the process of falling. Second, it is essential to consider how these features are distinguished from ADLs such as sitting and jumping, and the process of feature extraction will be greatly delayed. Classification and recognition based on deep learning are currently more commonly used in fall detection algorithms, which can automatically extract feature information. Due to this advantage, deep learning methods have become more and more popular in the research community. They have been used in numerous areas in which they have played a role equivalent to human experts. Generally, the steps involved in deep learning methods using wearable sensor data are to preprocess the acquired signals, extract features from signal segments, and train a model that uses these features as input [29]. Therefore, the existing research in the field of wearable sensor data fall risk assessment mainly focuses on engineering optimizing features. The extracted features are used as input to different deep learning algorithms to predict the occurrence of falls. Mirto Musci [30] used a fall detection algorithm based on LSTM, taking a long-time sequence as input and automatically extracting temporal information. He Jian [31] utilized the FD-CNN network to preprocess the input data into a picture format, where the three-axis acceleration is normalized and mapped to the RGB channel. Four hundred three-axis data windows can be regarded as 20 × 20 pixels and automatically extract spatial feature information.

## 3. Materials and Methods

As shown in Figure 1, the framework proposed in this study comprises three main steps: motion and data acquisition, data pre-processing, and CNN-Casual LSTM algorithm-based feature extraction and classification. In detail, the framework consists of two stages: Stage 1—preprocessing the acquired data into a form suitable to apply CNN-Casual LSTM; Stage 2—automatic feature extraction and learning for the changes of the data when a fall event occurs using the CNN-Casual LSTM algorithm and using the trained CNN-Casual LSTM model to finally determine whether a fall occurs by using image recognition and classification.

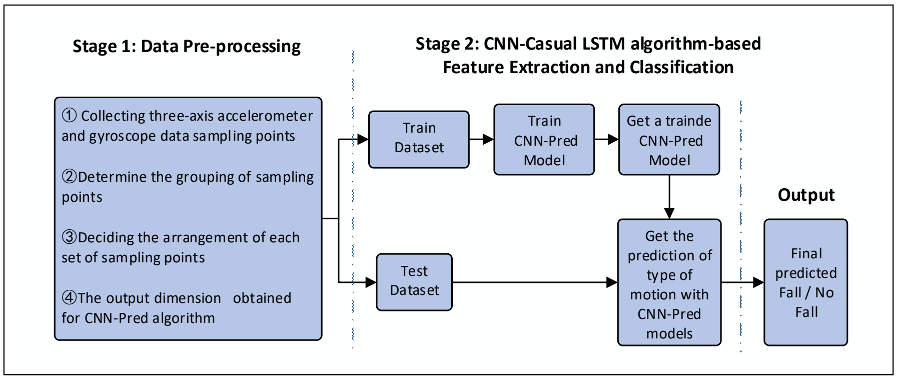

Stage 1: Data Pre-processing
Stage 2: CNN-Casual LSTM algorithm-based Feature Extraction and Classification
① Collecting three-axis accelerometer and gyroscope data sampling points
②Determine the grouping of sampling points
③Deciding the arrangement of each set of sampling points
④The output dimension obtained for CNN-Pred algorithm
Train Dataset
Train CNN-Pred Model
Get a trainde CNN-Pred Model
Test Dataset
Get the prediction of type of motion with CNN-Pred models
Output
Final predicted Fall / No Fall

**Figure 1.** The proposed approach for fall detection by using a convolutional neural network.

### 3.1. Motion and Data Acquisition

We used the public data set SisFall [32] to test the accuracy and latency of the model. This data set uses a wearable embedded device to collect three-axis acceleration and three-axis angular velocity during volunteer activities. The device is worn at the waist. The data are provided by a total of 38 volunteers: (1) 23 young people and a 60 year old judo athlete provided 19 types of non-fall and 15 types of fall data; (2) 15 types of non-fall data provided by 14 healthy people over 62 years old.

The SisFall data set collects three-axis acceleration and three-axis rotation angular velocity information by fixing the embedded device on the volunteers' waists.

### 3.2. Data Pre-Processing

According to statistics, human falling time is generally less than 2 s [31]. The time duration of a received three-axis accelerometer and three-axis gyroscope data point received in Sisfall is 12 s (sampling frequency is 200 Hz and 2400 sampling points in total); thus, the data were divided into six pieces associated with time information. Each X (2 s piece) contains 400 sampling points:

$$\mathrm{X_t} = [\mathrm{X_{t\_accx}}, \mathrm{X_{t\_accy}}, \mathrm{X_{t\_accz}}, \mathrm{X_{t\_gyrox}}, \mathrm{X_{t\_gyroy}}, \mathrm{X_{t\_gyroz}}] \tag{1}$$

where $\mathrm{X_t}$ is the data point at the time t; $\mathrm{X_{t\_accx}}$, $\mathrm{X_{t\_accy}}$, and $\mathrm{X_{t\_accz}}$, respectively, represent the three-axis acceleration; and $\mathrm{X_{t\_gyrox}}$, $\mathrm{X_{t\_gyroy}}$, and $\mathrm{X_{t\_gyroz}}$ represent the three-axis rotation angular velocity.

We rearrange the feature values based on measurement types and axes. For each group, we reshape the 400 sampling points to 20 × 20 to obtain the spatial relationship between the following values:

$$Y = [F_{accx}, F_{accy}, F_{accz}, F_{gyrox}, F_{gyroy}, F_{gyroz}], \tag{2}$$

where $F_{accx}$, $F_{accy}$, and $F_{accz}$ are the feature vectors of three-axis acceleration, respectively; $F_{gyrox}$, $F_{gyroy}$, and $F_{gyroz}$ are the feature vectors of three-axis rotation angular velocity, respectively; and Y is the aggregated output.

Therefore, after processing, the output has dimensions of 6 × 6 × 20 × 20 and is used as the input of the subsequent network.

### *3.3. CNN-Casual LSTM Algorithm-based Feature Extraction and Classification*

The architecture of the neural network in this paper is shown in Figure 2. The network is divided into coding layer, decoding layer, and classification layer. The encoding layer includes three layers of CNN, three layers of Causal LSTM, and a single layer of GHU. The encoding layer finally converts the data dimension from [6,6,20,20] to [6,10,10,128] through a series of transformations. The decoding layer includes three layers of deconvolution and three layers of Casual LSTM. The decoding layer restores the data through deconvolution and transforms the data dimension into [1,6,20,20]. The decoding layer outputs a hidden variable that is classified by ResNet18. The specific introduction of each layer will be provided below.

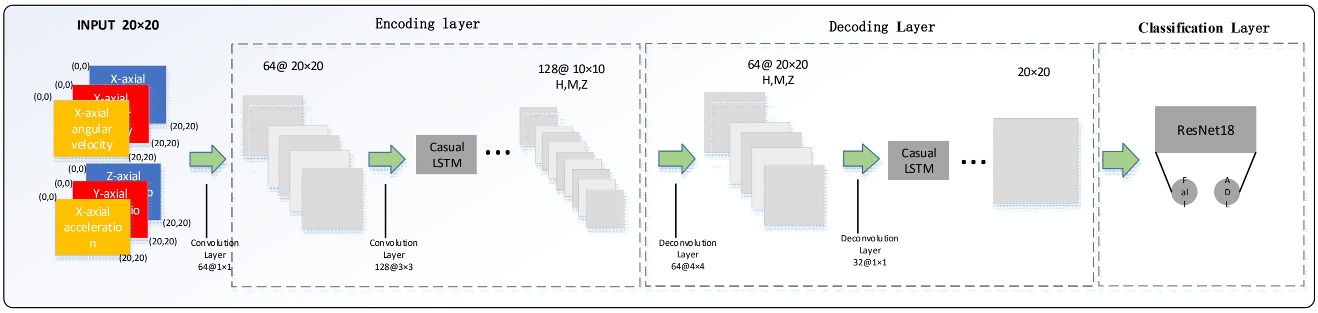


**Figure 2.** Architecture of neural network implemented for fall detection with classification of the data.

#### Causal LSTM

Causal LSTM is a gated recurrent neural network (Gated RNN) proposed by Yunbo Wang [33] from the Tsinghua team in 2018. Causal LSTM includes a three-layer structure. In the case of the same number of levels (about 8000 samples), more non-linear operations are added to extract the features, and the dual memories are linked in a cascaded manner. Therefore, compared with traditional LSTM, Causal LSTM obtains stronger spatial correlation and short-term dynamic modeling abilities, which is more conducive for capturing short-term dynamic changes and emergencies.

The structure of Causal LSTM is shown in Figure 3. Compared with LSTM, all gates of Causal LSTM are jointly determined by X, H, and C. The “input gate” is the information added to the cell. The “forget gate” determines the information to be discarded. The “output gate” determines the final output.

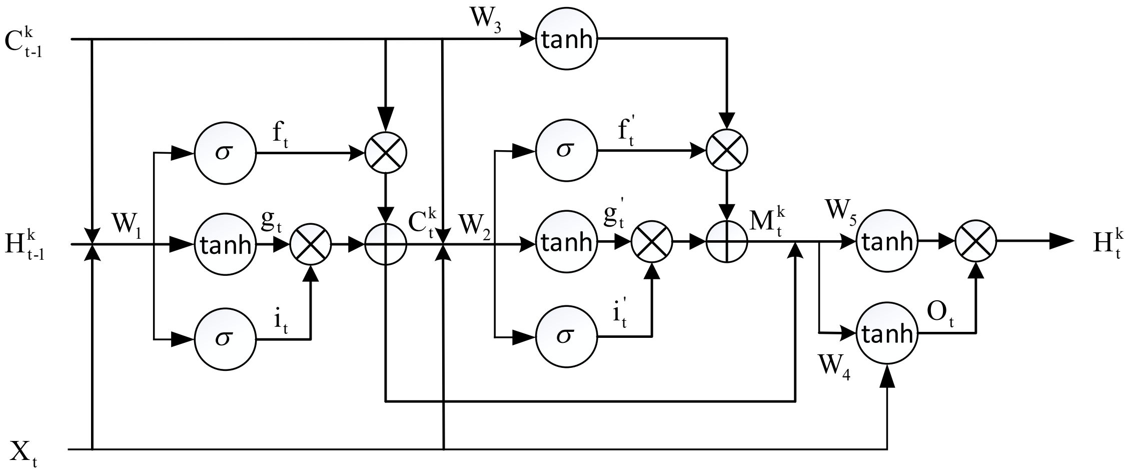


**Figure 3.** Causal LSTM in which spatio-temporal memories are connected in cascade through gated structures.

Causal LSTM has a three-layer structure. The output of the first layer is $C_{\mathrm{t}}^{\mathrm{k}}$, and it is determined by the input $X_{\mathrm{t}}$, the output response $H_{\mathrm{t}-1}^{\mathrm{k}}$, and $C_{\mathrm{t}-1}^{\mathrm{k}}$. K is the number of hidden layers, and C is the temporal state including temporal dimension information:

$$\begin{pmatrix} \mathrm{g_t} \\ \mathrm{i_t} \\ \mathrm{f_t} \end{pmatrix} = \begin{pmatrix} \tan \mathrm{h} \\ \sigma \\ \sigma \end{pmatrix} \mathrm{W_1} * [\mathrm{X_t}, \mathrm{H_{t-1}^k}, \mathrm{C_{t-1}^k} \Big] \tag{3}$$

$$C_{\mathrm{t}}^{k} = f_t \otimes C_{\mathrm{t}-1}^{k} + i_t \otimes g_t, \tag{4}$$

where * is convolution operation, $\sigma(\cdot)$ is a Sigmoid function: $\sigma(x) = 1/(1+e^{-x})$, $W_1$ is a convolutional filter, $\otimes$ calculates the Hadamard product between vectors, $\mathrm{f_t}$ is a forget gate, $\mathrm{i_t}$ is an input gate, and $\mathrm{g_t}$ is an intermediate long-term memory state.

The output of the second layer structure is $M_{\mathrm{t}}^{k}$, determined by $X_t$, $C_t^k$, and the previous layer $M_t^{k-1}$. M determines the spatial state of the cell and contains spatial dimension information:

$$\begin{pmatrix} g_t' \\ i_t' \\ f_t' \end{pmatrix} = \begin{pmatrix} \tanh \\ \sigma \\ \sigma \end{pmatrix} W_2 * \left[ X_t, C_t^k, M_t^{k-1} \right] \tag{5}$$

$$M_{\mathrm{t}}^{k} = f_t' \otimes \tanh\left( W_3 * M_t^{k-1} \right) + i_t' \otimes g_t', \tag{6}$$

where $f_t'$ is a forget gate; $i_t'$ is an input gate; and $g_t'$ is an intermediate long-term memory state.

The output of the third layer structure is the output of the cell and includes time and space state information:

$$o_{\mathrm{t}} = \tanh\left( W_4 * \left[ X_t, C_t^k, M_t^k \right] \right) \tag{7}$$

$$H_{\mathrm{t}}^{k} = o_t \otimes \tanh\left( W_5 * \left[ C_t^k, M_t^k \right] \right), \tag{8}$$

where the output is $H_{\mathrm{t}}^{k}$, and it is determined by the time state $C_t^k$, space state $M_t^k$, and input $X_t$ at time t; and $o_t$ is an intermediate state.

Causal LSTM has a long-term problem with gradients in back-propagation. In particular, due to the long transition, time memory information may be forgotten, especially when it is processing information with periodic motions or frequent occlusions.

We need an information highway to learn skip-frame relations. Gradient Highway Unit (GHU) is a "high-speed channel" in neural networks that can effectively transmit gradients in very deep networks and then prevent long-term gradient dispersion.

The structure of GHU is shown in Figure 4. The input is the output of the current lower layer $X_{\mathrm{t}}$ and the input of the GHU at the previous moment $Z_{\mathrm{t}-1}$. The input of the current and the previous moments is connected so that the propagation distance is shortened.

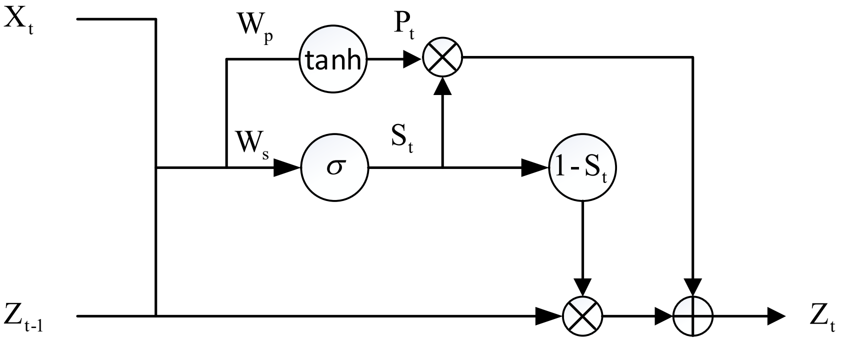


**Figure 4.** In GHU, the gradient highway connects the current time step directly with previous inputs. $\sigma(\cdot)$ is a Sigmoid function.

### *3.4. Proposed CNN-Casual LSTM Network*

This paper designs a fall detection model based on the CNN-Casual LSTM Network network. It mainly consists of three parts:

#### 3.4.1. Encoding Layer

As shown in Figure 5, this layer has a total of 6 moments of data input and a 3-layer network structure, including CNNs, Causal LSTMs, and a GHU high-speed channel. The first layer uses CNN to sample the data at each moment in 1 × 1. Then, through the Causal LSTM layer, a series of linear and nonlinear transformations are performed on the sampling results according to Equations (3)–(8). The latter two layers use CNN to perform 3 × 3 downsampling to extract feature information from the input and spatial state information at the previous moment and extract spatio-temporal information from reduced-dimensional information. The three-tier cascade structure ensures that the model can learn enough spatio-temporal features. The weight matrices inside the network are optimized iteratively during the training phase and learn spatio-temporal characteristics. The GHU network is inserted between the first and the second layers and directly transmits the information obtained from the first layer to the next moment, which can effectively prevent gradient dispersion caused by the deep network.

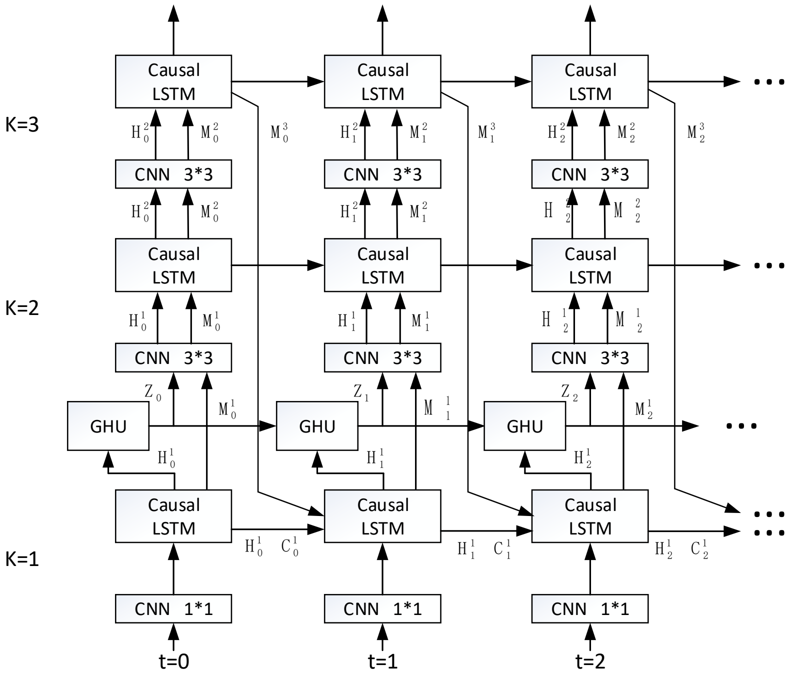


**Figure 5.** Encoding layer structure.

#### 3.4.2. Decoding Layer

The layer is also in the form of a three-layer cascade in Figure 6, including deconvolutional layers, Causal LSTMs, and a GHU high-speed channel in the reverse order of those in the encoding layer.

$$H_6^1, C_6^1, M_6^1 = \mathrm{CasualLSTM}\left(H_5^3, H_5^3, C_5^3, M_5^3\right) \tag{9}$$

$$H_6^2, C_6^2, M_6^2 = \mathrm{CasualLSTM}\left(H_6^1, H_5^2, C_5^2, M_6^1\right) \tag{10}$$

$$Z_6 = \mathrm{GHU}\left(H_5^2, Z_5\right) \tag{11}$$

$$H_6^3, C_6^3, M_6^3 = \mathrm{CasualLSTM}\left(Z_6, H_5^1, C_5^1, M_6^2\right) \tag{12}$$

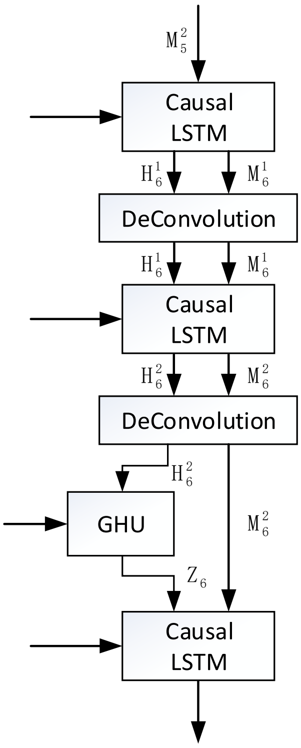


**Figure 6.** Decoding layer structure.

According to Equations (9)–(12), the spatiotemporal information obtained by the coding layer is processed and restored by the Causal LSTM layer and the deconvolution layer twice. It is then processed through a layer of GHU in order to directly obtain the information of the previous layer and finally through a layer of Causal LSTM. The three-layer cascade structure ensures that the spatiotemporal information is mapped to the hidden variable output that is more conducive to the work of the classification network.

### 3.4.3. Classification Layer

This layer classifies the output of the decoding layer (falling or non-falling). This article uses the ResNet18 basic network that comes with PyTorch to perform the classification task. The network contains 17 convolutional layers and one fully connected layer.

## 4. Experiments

In order to ensure the integrity of the information, we use a 12 s time window to extract data for SisFall. After processing the data set, 1798 cases of falls and 6146 cases of non-falls are obtained.

The hardware of this experiment includes an Intel Core i7-9700 processor and an NVIDIA GeForce GTX 1660 graphics card. The software environment is Python 3.7.0 with PyTorch 1.2 and CUDA 10.2.

### *4.1. Ablation Experiments*

The experimental data set is divided into 80% training set and 20% test set. The ablation experiment in this article is divided into two parts. First of all, similar to the one shown in Figure 5, we have conducted many experiments through the network structure of k = 2~4 layers to verify the rationality of the network structure proposed in this paper. The structure of k = 2 is similar to the structure in Figure 5. Compared with the network structure of K = 3, it only has a CNN convolutional layer and a Casual LSTM cell structure in the coding layer. On the contrary, the structure of k = 4 adds a layer of structure on the basis of this article, and the layer structure includes one more deconvolution on the original basis in order to achieve the same number of channels. In addition, the number of decoding layers corresponds to the number of coding layers. Second, we use the ST-LSTM [34] unit proposed by Ashesh Jain as an alternative to the Casual LSTM unit. We change all the places where the Casual LSTM unit should be set to ST-LSTM while other parameters remain unchanged in order to explore the effectiveness of the Casual LSTM unit.

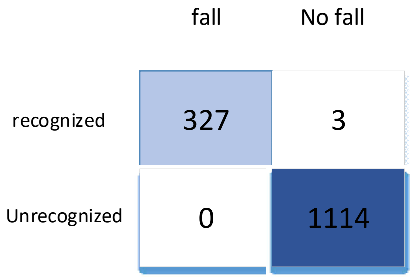
The experimental results in Table 1 show that, compared to ST-LSTM, although it can also send a reminder for every fall behavior, the performance of Casual-LSTM has been improved, which can better reduce the false alarm rate. In addition, the two-layer Casual-LSTM's ability to extract spatial information is obviously inferior to the three-layer structure. The training process converges slowly, and the sensitivity of the experimental results decreases (judging that the fall data does not fall), which can result in dangerous results. The four-layer Casual-LSTM can also achieve the same excellent results, but we finally chose the three-layer LSTM as the structure of this article after considering its better operating efficiency.

**Table 1.** Classification accuracy, sensitivity, and specificity of ablation experiments on the SisFall data set.

| MODEL | ACC/% | SEN/% | SPE/% |
|---|---|---|---|
| CNN-CASUAL LSTM | 99.79 | 100 | 99.73 |
| CNN-ST LSTM | 99.17 | 100 | 98.92 |
| TWO-LAYER | 99.65 | 99.39 | 99.73 |
| FOUR-LAYER | 99.58 | 100 | 99.46 |

### 4.2. Experimental Results

Figure 7 shows the performance of the CNN-Casual LSTM network on the SisFall test set in the form of a confusion matrix. Of the 1444 test examples, only 3 predictions are wrong (three false positives and zero false negatives). It shows that the network proposed in this article is very sensitive to falls and has high recognition accuracy. After reviewing the three cases of data, we found that these data all come from young people and include actions such as squatting or squatting after taking off, which shows that our network cannot accurately identify such behaviors.



| | fall | No fall |
|---|---|---|
| recognized | 327 | 3 |
| Unrecognized | 0 | 1114 |

**Figure 7.** The confusion matrix on SisFall.

Table 2 provides the performance of algorithms with three indicators. We selected RNN and LSTM [35] and a convolutional neural network (FD-CNN [31]) as baselines.

**Table 2.** Classification accuracy, sensitivity, and specificity of comparative experiments on SisFall data set.

| MODEL | ACC/% | SEN/% | SPE/% |
|---|---|---|---|
| RNN | 95.23 | 91.24 | 97.06 |
| LSTM | 99.58 | 99.27 | 99.73 |
| FD-CNN | 97.47 | 97.44 | 99.63 |
| CNN-CASUAL LSTM | 99.79 | 100.00 | 99.73 |

According to the data in Table 2, the ACC of this model is 99.79%; the SEN is 100%; and the SPE is 99.73%. It can be observed that the network proposed in this paper is superior to other methods in three metrics.

Among them, the ACC of LSTM [35] is 99.58%, and the SEN is 99.27%, both of which are slightly lower than this article. We think this may be due to the network structure of LSTM. LSTM tends to pay more attention to adjacent information when dealing with long sequences. It is very prone to gradient dispersion. Therefore, if the data sequence is long, the data utilization rate is insufficient. In this paper, by processing of the data set, the 12S long sequence data are processed into 6 groups of 2S data. For the 400 data volume of each group of data, we will first extract 20 data points into one point, and then the processed 20 consecutive points are processed. By arranging these data in space and then using convolution to extract features, the process of extracting features does not only rely on long or short time intervals, long time interval features such as the overall action of the falling process, and short time intervals such as falling that invert the details of the action in progress. It takes into account the receptive fields of different time scales while ensuring that there is no gradient dispersion phenomenon.

In addition, FD-CNN [31] treats four hundred three-axis data as 20 × 20 × 3 pixels as input. It displays the original coordinate axis information as spatial information, and each group of data is 2S. However, a fall is a continuous process that may include leaning forward, bending the lower limbs, and rebounding from contact with the ground. This means that if a fall occurs at the end of a set of data and the beginning of the next set, part of the information that contains the fall process may be missing, and the lack of time information in this part may cause it to be misjudged. This may be the reason why its ACC is 97.47%. The spatiotemporal network model proposed in this paper extracts the spatial features of each input through the coding layer, while retaining the influence of time information through the GHU fast channel, which can make full use of spatial and temporal information, which means that it has a high recognition rate for fall behavior. For every fall occurrence, it can trigger an alarm, which improves the safety of the elderly.

## 5. Conclusions

In this work, we study a network model that relies on sensor device input, compared it to other networks, and observed that the CNN-Casual LSTM network can extract temporal and spatial information better, thereby improving the accuracy of fall detection. The detection accuracy of the algorithm in this paper is 99.79% (0.21% false-positive rate and 0% false-negative rate). The detection performance is better than other methods, and it can detect every fall occurrence, which further proves the high robustness and stability of this method. However, this article still has many directions worthy of improvement. First, limited by the data of the public data set, the data set of the network of this article is very imbalanced, and the fall data only accounts for about 27% of all data. Secondly, the network structure proposed in this article is too redundant. In order to process the 12S data set of the public data set, six sets of 2S data sets are adapted, which renders the training process of the entire network very long. In addition, we also have many aspects of work that can continue to be studied in depth. Firstly, the application scenario of this article is a nursing home. However, considering the risk of falling, the elderly data in the public data set accounts for a relatively small proportion. In the future work, if allowed, we can collect more fall data of the elderly and train the network structure of this article again. Secondly, the network structure of the text is relatively complicated, and it is necessary to explore more convenient channels to transmit information in the future.

**Author Contributions:** Conceptualization, J.W. (Jiang Wu); methodology, J.W. (Jiang Wu); software, J.W. (Jiang Wu) and J.W. (Jiale Wang); validation, C.W.; investigation, A.Z.; data curation, J.W. (Jiale Wang); writing—original draft preparation, J.W. (Jiale Wang); writing—review and editing, A.Z.; supervision, C.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No.LGF19F010008.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Nari, M.I.; Suprapto, S.S.; Kusumah, I.H.; Adiprawita, W. A simple design of wearable device for fall detection with accelerometer and gyroscope. In Proceedings of the 2016 International Symposium on Electronics and Smart Devices (ISESD), Bandung, Indonesia, 29–30 November 2016.
2. World report on ageing and health. *Indian J. Med Res.* **2017**, *145*, 150–151. [CrossRef]
3. National Bureau of Statistics of the People's Republic of China. *The Sixth National Population Census of the People's Republic of China*; China National Bureau of Statistics: Beijing, China, 2010.
4. Mubashir, M.; Shao, L.; Seed, L. A survey on fall detection: Principles and approaches. *Neurocomputing* **2013**, *100*, 144–152. [CrossRef]
5. Chaccour, K.; Darazi, R.; El Hassani, A.H.; Andres, E. From fall detection to fall prevention: A generic classification of fall-related systems. *IEEE Sens. J.* **2016**, *17*, 812–822. [CrossRef]
6. Igual, R.; Medrano, C.; Plaza, I. Challenges, issues and trends in fall detection systems. *BioMed. Eng. OnLine* **2013**, *12*, 66. [CrossRef]
7. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
8. Waheed, S.A.; Khader, P. A Novel Approach for Smart and Cost Effective IoT Based Elderly Fall Detection System Using Pi Camera. In Proceedings of the 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, India, 14–16 December 2017.
9. Huang, Z.; Liu, Y.; Fang, Y.; Horn, B. Video-based Fall Detection for Seniors with Human Pose Estimation. In Proceedings of the 2018 4th International Conference on Universal Village (UV), Boston, MA, USA, 21–24 October 2018.
10. Ogawa, Y.; Naito, K. Fall detection scheme based on temperature distribution with IR array sensor. In Proceedings of the 2020 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 4–6 January 2020.
11. Sun, Y.; Hang, R.; Li, Z.; Jin, M.; Xu, K. Privacy-Preserving Fall Detection with Deep Learning on mmWave Radar Signal. In Proceedings of the 2019 IEEE Visual Communications and Image Processing (VCIP), Sydney, Australia, 1–4 December 2019.
12. Desai, K.; Mane, P.; Dsilva, M.; Zare, A.; Ambawade, D. A Novel Machine Learning Based Wearable Belt For Fall Detection. In Proceedings of the 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 2–4 October 2020.
13. Astriani, M.S.; Heryadi, Y.; Kusuma, G.P.; Abdurachman, E. Long Short-Term Memory for Human Fall Detection Based Gamification on Unconstraint Smartphone Position. In Proceedings of the 2019 International Congress on Applied Information Technology (AIT), Yogyakarta, Indonesia, 4–6 November 2019.
14. Bosch-Jorge, M.; Sánchez-Salmerón, A.; Valera, Á.; Ricolfe-Viala, C. Fall detection based on the gravity vector using a wide-angle camera. *Expert Syst. Appl.* **2014**, *41*, 7980–7986. [CrossRef]
15. Yao, L.; Yang, W.; Huang, W. A fall detection method based on a joint motion map using double convolutional neural networks. *Multimed. Tools Appl.* **2020**, *1*, 1–18. [CrossRef]
16. Agrawal, S.C.; Tripathi, R.K.; Jalal, A.S. Human-fall detection from an indoor video surveillance. In Proceedings of the 2017 8th International Conference on Computing, Delhi, India, 3–5 July 2017.
17. Kong, X.; Meng, Z.; Lin, M.; Tomiyama, H. A Privacy Protected Fall Detection IoT System for Elderly Persons Using Depth Camera. In Proceedings of the 2018 International Conference on Advanced Mechatronic Systems (ICAMechS), Zhengzhou, China, 30 August–2 September 2018.
18. Tzeng, H.W.; Chen, M.Y.; Chen, J.Y. Design of fall detection system with floor pressure and infrared image. In Proceedings of the 2010 International Conference on System Science and Engineering, Taipei, Taiwan, 1–3 July 2010.
19. Sadreazami, H.; Bolic, M.; Rajan, S. Fall Detection using Standoff Radar-based Sensing and Deep Convolutional Neural Network. *IEEE Trans. Circuits Syst. II: Express Briefs* **2019**, *67*, 197–201. [CrossRef]
20. Yang, T.; Cao, J.; Guo, Y. Placement selection of millimeter wave FMCW radar for indoor fall detection. In Proceedings of the 2018 IEEE MTT-S International Wireless Symposium (IWS), Chengdu, China, 6–10 May 2018; pp. 1–3.
21. Gao, T.; Yang, J.; Huang, K.; Hu, Q.; Zhao, F. Research and Implementation of Two-Layer Fall Detection Algorithm. In Proceedings of the 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, China, 10–12 November 2018.
22. Mannini, A.; Trojaniello, D.; Cereatti, A.; Sabatini, A.M. A Machine Learning Framework for Gait Classification Using Inertial Sensors: Application to Elderly, Post-Stroke and Huntington's Disease Patients. *Sensors* **2016**, *16*, 134. [CrossRef] [PubMed]

23. Shahiduzzaman, K.M.; Hei, X.; Guo, C.; Cheng, W. Enhancing fall detection for elderly with smart helmet in a cloud-network-edge architecture. In Proceedings of the 2019 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Yilan, Taiwan, 20–22 May 2019; pp. 1–2.
24. Mousavi, S.A.; Heidari, F.; Tahami, E.; Azarnoosh, M. Fall detection system via smart phone and send people location. In Proceedings of the 2020 28th European Signal Processing Conference (EUSIPCO), Amsterdam, Netherlands, 18–21 January 2021.
25. Singh, K.; Rajput, A.; Sharma, S. Human Fall Detection Using Machine Learning Methods: A Survey. *Int. J. Math. Eng. Manag. Sci.* **2019**, *5*, 161–180. [CrossRef]
26. Wang, R.D.; Zhang, Y.L.; Dong, L.P.; Lu, J.W.; He, X. Fall detection algorithm for the elderly based on human characteristic matrix and SVM. In Proceedings of the 2015 15th International Conference on Control, Automation and Systems (ICCAS), Busan, Korea, 13–16 October 2015.
27. Otanasap, N. Pre-Impact Fall Detection Based on Wearable Device Using Dynamic Threshold Model. In Proceedings of the 2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Guangzhou, China, 16–18 December 2016.
28. Özdemir, A.T.; Barshan, B. Detecting falls with wearable sensors using machine learning techniques. *Sensors* **2014**, *14*, 10691–10708. [CrossRef]
29. Kiprijanovska, I.; Gjoreski, H.; Gams, M. Detection of Gait Abnormalities for Fall Risk Assessment Using Wrist-Worn Inertial Sensors and Deep Learning. *Sensors* **2020**, *20*, 5373. [CrossRef]
30. Musci, M.; Martini, D.D.; Blago, N.; Facchinetti, T.; Piastra, M. Online Fall Detection using Recurrent Neural Networks on Smart Wearable Devices. *IEEE Trans. Emerg. Top. Comput.* **2020**, *9*, 1276–1289. [CrossRef]
31. He, J.; Zhang, Z.; Wang, X.; Yang, S. A low power fall sensing technology based on FD-CNN. *IEEE Sens. J.* **2019**, *19*, 5110–5118. [CrossRef]
32. Sucerquia, A.; Lopez, J.D.; Vargas-Bonilla, J.F. SisFall: A Fall and Movement Dataset. *Sensors* **2016**, *17*, 198. [CrossRef] [PubMed]
33. Wang, Y.; Gao, Z.; Long, M.; Wang, J.; Yu, P.S. *PredRNN++: Towards A Resolution of the Deep-in-Time Dilemma in Spatiotemporal Predictive Learning*; PMLR: Beijing, China, 2018.
34. Jain, A.; Zamir, A.R.; Savarese, S.; Saxena, A. Structural-RNN: Deep Learning on Spatio-Temporal Graphs. In Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
35. Hsieh, S.T.; Lin, C.L. Fall Detection Algorithm Based on MPU6050 and Long-Term Short-Term Memory network. In Proceedings of the 2020 International Automatic Control Conference (CACS), Hsinchu, Taiwan, 4–7 November 2020.