Time Series data augmentation and dropout roles in Deep Learning applied to Fall Detection

Enol García González, José R. Villar, and Enrique de la Cal

Computer Science Department, University of Oviedo, Oviedo, Spain
`enolgargon@gmail.es`, {`villarjose,delacal`}`@uniovi.es`

Abstract. Fall Detection is one of the most interesting and challenging research topics in the world today because of its implications in society and also because the complexity of processing Time Series (TS). Plenty of research has been published in the literature, several of them introducing Deep Learning (DL) Neural Network (NN) as the modelling element. In this study we analyse one of these contributions and address several enhancement using TS data augmentation and dropout. Moreover, the possibility of reducing the NN to make it lighter has been studied. The NN has been implemented using Keras in Python and the experimentation includes an staged fall publicly available data set. Results show the TS data augmentation together with dropout helped in learning a more robust and precise model. Future work includes introducing different types of cross-validation as well as introducing other types of DL models more suitable for TS.

Keywords: Fall Detection, Neural Network, Time Series, Accelerometer, Wearables

1 Introduction and related work

The Fall Detection (FD) represents a challenge that, if overcome, could significantly improve the quality of life of people living alone, especially the elderly[1]. During the last few years, different approaches have been proposed to solve the problem of FD, but most methods face a major challenge when processing time series in which there is no window covering the whole time and data are obtained progressively. This study focuses on FD using wearable devices that includes an tri-axial accelerometer (3DACC) placed on a wrist; as shown in [2], this solution might be more usable for elder people.

The literature concerning the focused specific problem includes a wide variety of solutions. Machine learning (ML) is the main way to address the problems of fall detection and classification. Some examples of these methods are those presented In [3, 4] feature extraction stage and Support Vector Machines are used in order to classify the TS windows. Thresholds could be used in FD [5–7] using instances that have been previously labeled as a function of acceleration predefined magnitude values. Thresholds could be used in FD in order to establish rules that drive the final decision [5, 8, 9]. [10] shows a comparison of these type of methods. In [11] an user centred solution is proposed, where a single model is build for a concrete individual, this model identifies the normal activities and every thing outside the normal activity could be a possible fall. Besides, a solution based on clustering the detected peaks and building specific models for each cluster was proposed in [12]. In [13], the authors show how all these studies can be useful in real life by improving the quality of people's lives through the advancement of technology with the diffusion and ease of access to wearables. However, the research published in [14] presents aspects such as power consumption and real-time data processing that make it necessary to work in lighter FD systems.

One of the main characteristics of all the previous approaches is that all of them developed models with a reduced computational cost. However, with the raise of Deep Learning (DL), several studies have proposed using DL as a FD service [15]. One of these is [16], where an open source dataset and LSTM deep learning model are used in order to detect falls and daily activities with a high success rate. Using DL the input change dramatically: instead of detecting peaks and producing a feature extraction from the TS, with DL a complete TS window is feed to the model without further pre-processing. Alternatively, a Convolutional network (convnet) solution is proposed in [17], where 4 blocks of convnet models are sequenced to produce the final TS label. Interestingly, the authors obtained a good model but some of the normal enhancements in DL were not considered.

This study focuses on extending the work presented in [17] to introduce two well known DL enhancements, such as data augmentation and dropout to avoid overfitting. Furthermore, several different configurations have been tested in order to reduce the dimension of the convnet. The main novelty of this study, however, is the designed TS data augmentation. The experimentation will compare all the solutions and shows the benefits of the designed TS data augmentation.

The structure of this manuscript is as follows. The next section describes the convnet proposed in [17] as it represents the basis of this research. Moreover, the section also give details of the TS data augmentation and the dropout introduced to the model. Finally, the data set and the experimentation set up are described at the end of this section. Section 3 includes the obtained results and the comparison of the different options analyzed in this research, together with a discussion on the results. Finally, the conclusions are drawn.

2 Material and methods

This section will be responsible for defining the motivation for the project, as well as the starting NN for the research. It also includes a description of the modifications that were made to the original NN in order to improve it and the data and experiments carried out.

2.1 Neural Network models used in this study

This study is based on the network proposed in [17]. This NN is built by 4 levels where we found a Convolution layer, a Normalization layer, a ReLU layer and a Max Pooling layer in each level. Furthermore, every Convolution and max Pooling Layer has a filter size of 1x5. The first Convolution layer has 16 filters, the second 32, the third 64 and the fourth 128. Finally, there is a classification dense layer with Softmax activation which gives the output of the NN. Fig 1 depicts the structure of the NN. The authors studied this NN with the UMAFall data set mixing all the TS in a single bag and using 10 fold cross validation. From now on, this model is referred as CN_{CAS}.

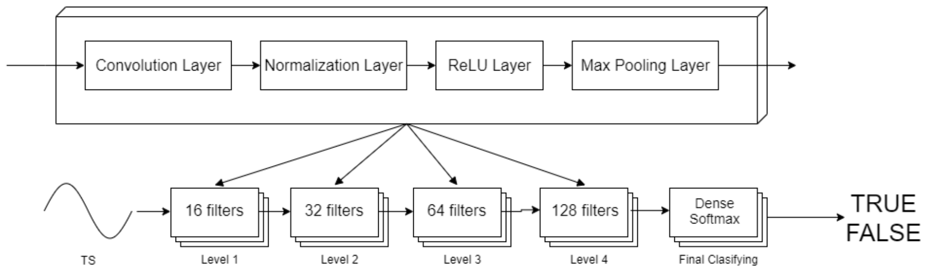

Fig. 1. Structure of the NN proposed in [17].

This NN model has been revisited, proposing different alternative configurations. The NN was considered to be too heavy and over-dimensioned for the size of the problem to be solved, therefore, networks were considered in which each level contained the same layers, but reducing the number of levels, in particular 2 and 3 levels were considered. These alternatives have been tested but their results were poorer than those from the CN_{CAS}, therefore their results have not been included in this research.

2.2 Enhancements in the network learning

Two main improvements have been included: introducing data augmentation in the training data feed and several over-fitting avoiding drop-out layers.

Data Augmentation. The first modification that was made to CN_{CAS} was to apply data augmentation to achieve a much wider and more varied set of training data. In this way, it is possible to eliminate the over-fitting that appears when all the falls are located at the same time in the data window or because they are too similar in magnitude. The NN that adds data augmentation on the CN_{CAS} NN will henceforth be referred to as CN_{CAS+DA}. The augmentation was done in two ways:

– A random number is generated to increase or decrease the difference between two consecutive values of the time series.
– In addition, the moment at which the fall within the time series occurs was modified. For this, the series was observed and it was seen that the magnitude at the beginning and at the end was identical, since it starts at rest and ends at rest, which allows moving the time series in time using rotation, that is, all the data are delayed a fixed number of milliseconds and those that exceed the end of the series are placed at the beginning of it.

An example of this data augmentation process can be seen in Fig 2.

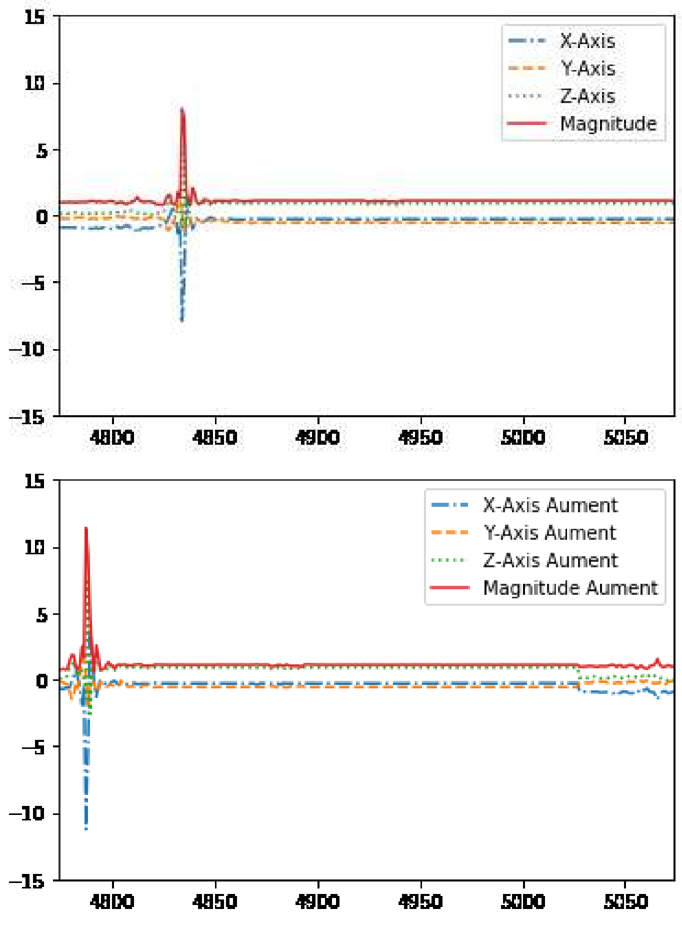

Fig. 2. Comparison of data before and after the DA process. The X-axis shows the sample's index, the Y-axis is multiple of $G = 9.8m/s^2$. The scale and shift of the multivariate TS is clearly shown.

Drop-out. A modification we made to the CN_{CAS+DA} NN to avoid over-adjustment during training was to add a drop-out layer between each level. We denote the NN that makes used of drop-out together with data augmentation as $\text{CN}_{CAS+DA+DO}$. Interestingly, the NN including drop-out only has also been evaluated but its results were not competitive.

2.3 Data set and cross validation

In order to compare this results with the [17], we use the staged falls data set provided in [18]. In this data set, up to 19 participants performed several human activities of daily living plus staged falls. Three different types of fall were staged: forward, lateral and backwards fall. The data was gathered using inertial devices, including both 3DACC, magnetometer and gyroscope, placing the sensors on different body locations. In this research we consider only the TS from the 3DACC sensor placed on a wrist.

Each participant recorded several runs of each activity or staged fall, each run producing a TS including the acceleration components for each axis. All the TS have been introduced in a bag of TS with their corresponding label (either FALL or NOT_FALL), 20% of the TS are preserved for validation, while the remaining samples are kept for training and testing. A sliding window of size 650 miliseconds with a shift of 1 sample is used to evaluate each interval within a TS.

We use 10 fold cross validation for the training and testing stage. In this cross validation configuration, TS belonging to any of the participants can be included in the train and test, there is no distinction by participant. We are going to compare the different options explained before. For each model and fold the Accuracy, Kappa factor, Sensitivity and Specificity are determined.

3 Results and discussion

The obtained results are shown in several graphs and tables as detailed next:

- Table 1 shows the average of the different metrics and each network configuration CN_{CAS}, CN_{CAS+DA} and $\text{CN}_{CAS+DA+DO}$.
- Fig. 3 shows the box plots obtained for each network configuration.

Table 1. Average of the metrics obtained during the experiments

Experiment	Accuracy	Specificity	Sensitivity	Kappa
CN_{CAS}	0.6967	0.7091	0.7091	0.3947
CN_{CAS+DA}	0.8915	0.8870	0.8870	0.7829
$\text{CN}_{CAS+DA+DO}$	0.9125	0.9142	0.9142	0.7712

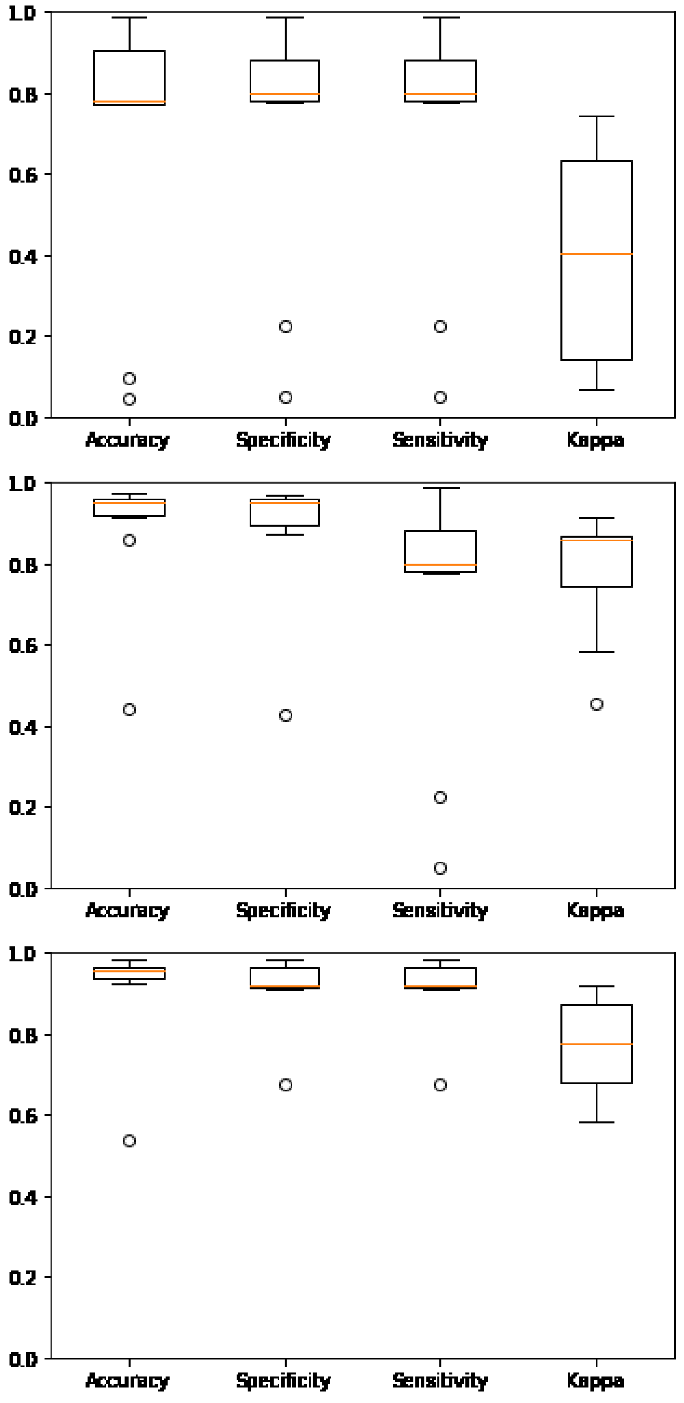

Fig. 3. Box plots obtained for each of the configurations. A box plot for each metric is included in each graph. Top, center and bottom correspond to CN$_{CAS}$, CN$_{CAS+DA}$ and CN$_{CAS+DA+DO}$.

The results of the experiments shows that both CN$_{CAS+DA}$ and CN$_{CAS+DA+DO}$ obtaine better metrics than previous the proposed in CN$_{CAS}$; the most remarkable improvement happening with the CN$_{CAS+DA}$ network. From this improvement we can see that CN$_{CAS}$ NN took learned the peaks in an specific position within a TS window; when the peak came from an modified staged fall, the CNN$_{CAS}$ was not able to identify the fall. By training the same network with a much larger data set, the performance with the test set has been greatly improved. Furthermore, if we look at the box plots of the networks we can see how the dispersion of metrics is reduced, since with the CN$_{CAS}$ network both very good and very bad networks were obtained. However, with the CN$_{CAS+DA}$ network the values taken from the metrics are more concentrated.

From the data reflected in the table, it seems that the CN$_{CAS+DA+DO}$ network offers an improvement over the CN$_{CAS+DA}$ network, but this improvement is obtained at the cost of sacrificing the concentration of results discussed above. While the CN$_{CAS+DA}$ network has the results of metrics very well concentrated, the new CN$_{CAS+DA+DO}$ network has more than one occasion in which it shows a malfunction. However, despite this dispersion, the results can be considered better.

Finally, it is worth mentioning that the Kappa and Accuracy results show how the balance in the data set is in compromise. Certainly, the data set has a big difference in the number of TS labeled as Fall or Not Fall. For the scope of this study we have not cope with this issue because we wanted to compare with the original method. However, future research includes a DA that copes with this particular issue, balancing the number of TS for each label.

4 Conclusion

In this study, a proposal for FD using DL NN has been refined with several elements: i) a 3DACC located on a wrist, ii) using TS data augmentation and iii) introducing dropout; these two latter to avoid the overfitting. A publicly available staged fall data set (UMA FALL) was used in the experimentation to evaluate and compare the options. With all the results obtained and compared in the previous topic it can be concluded that the CN$_{CAS+DA}$ and CN$_{CAS+DA+DO}$ networks are options that significantly improve the performance of the initial network.

Future work includes analysing several different staged fall data sets, and designing other types of NN, such as LTSM and recurrent networks or CONV1D NN. Moreover, more interesting TS data augmentation designs can be introduced in order to get a good variation of the signals.

ACKNOWLEDGMENT

This research has been funded by the Spanish Ministry of Science and Innovation under project MINECO-TIN2017-84804-R and by the Grant FCGRUPIN-IDI/2018/000226 project from the Asturias Regional Government.

References

1. Jahanjoo, A., Naderan, M., Rashti, M.J.: Detection and multi-class classification of falling in elderly people by deep belief network algorithms. Ambient Intelligence and Humanized Computing (2020)
2. Khojasteh, S.B., Villar, J.R., Chira, C., Suárez, V.M.G., de la Cal, E.A.: Improving fall detection using an on-wrist wearable accelerometer. Sensors 18 **5** (2018) 1350
3. Zhang, T., Wang, J., Xu, L., Liu, P.: Fall detection by wearable sensor and one-class svm algorithm. In Huang DS., Li K., I.G., ed.: Intelligent Computing in Signal Processing and Pattern Recognition. Volume 345 of Lecture Notes in Control and Information Systems. Springer Berlin Heidelberg (2006) 858–863
4. Wu, F., Zhao, H., Zhao, Y., Zhong, H.: Development of a wearable-sensor-based fall detection system. International Journal of Telemedicine and Applications **2015** (2015) 11
5. Bourke, A., O'Brien, J., Lyons, G.: Evaluation of a threshold-based triaxial accelerometer fall detection algorithm. Gait and Posture **26** (2007) 194–199
6. Fang, Y.C., Dzeng, R.J.: A smartphone-based detection of fall portents for construction workers. Procedia Eng. **85** (2014) 147–156
7. Fang, Y.C., Dzeng, R.J.: Accelerometer-based fall-portent detection algorithm for construction tiling operation. Autom. Constr. **84** (2017) 214–230
8. Huynh, Q.T., Nguyen, U.D., Irazabal, L.B., Ghassemian, N., Tran, B.Q.: Optimization of an acc. and gyro.-based fall det. algorithm. Journal of Sensors (2015)
9. Kangas, M., Konttila, A., Lindgren, P., Winblad, I., Jämsaä, T.: Comparison of low-complexity fall detection algorithms for body attached accelerometers. Gait and Posture **28** (2008) 285–291
10. Hakim, A., Huq, M.S., Shanta, S., Ibrahim, B.: Smartphone based data mining for fall detection: Analysis and design. Procedia Computer Science **105** (2017) 46–51
11. Villar, J.R., de la Cal, E.A., Fáñez, M., Suárez, V.M.G., Sedano, J.: User-centered fall detection using supervised, on-line learning and transfer learning. Progress in AI 8 **4** (2019) 453–474
12. Fáñez, M., Villar, J.R., de la Cal, E.A., Suárez, V.M.G., Sedano, J.: Feature clustering to improve fall detection: A preliminary study. SOCO 2019 (2019) 219–228
13. Godfrey, A.: Wearables for independent living in older adults: Gait and falls. Maturitas **100** (2017) 16–26
14. Igual, R., Medrano, C., Plaza, I.: Challenges, issues and trends in fall detection systems. BioMedical Engineering OnLine **12** (2013) 66
15. Casilari-P'erez, E., Lagos, F.G.: A comprehensive study on the use of artificial neural networks in wearable fall detection systems. Expert Systems With Applications **138** (2019)
16. XiaodanWu, Cheng, L., Chu, C., J. Kim, J.: Using deep learning and smartphone for automatic detection of fall and daily activities. Lecture Notes in Computer Science **11924** (2019) 61–74
17. Casilari, E., Lora-Rivera, R., García-Lagos, F.: A wearable fall detection system using deep learning. Advances and Trends in Artificial Intelligence (2019) 445–456
18. Casilari, E.: Umafall: A multisensor dataset for the research on automatic fall detection. Procedia Computer Science **110** (2017) 32–39