CapsFall: Fall Detection Using Ultra-Wideband Radar and Capsule Network

Hamidreza Sadreazami †, *Member, IEEE*, Miodrag Bolic †, *Senior Member, IEEE*,
and Sreeraman Rajan ‡, *Senior Member, IEEE*
†School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada
‡Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada

***Abstract*—Radar technology for at home health-care has many advantages such as safety, reliability, privacy-preserving and contact-less sensing nature. Detecting falls using radar has recently gained attention in smart health care. In this paper, CapsFall, a new method for fall detection using an ultra-wideband radar that leverages the recent deep learning advances is proposed. To this end, a radar time series is derived from the radar back-scattered matrix and its time-frequency representation is obtained and used as input to the the capsule network for automatic feature learning. In contrast to other existing methods, the proposed CapsFall method relies on multi-level feature learning from radar time-frequency representations. In particular, the proposed method utilizes a capsule network for automating feature learning and enhancing model discriminability. Experiments are conducted using a set of radar signals collected from ten subjects when performing various activities in a room environment. The performance of the proposed CapsFall method is evaluated in terms of classification metrics and compared with those of the other existing methods based on convolutional neural network, multi-layer perceptron, decision tree and support vector machine. The results show that the proposed CapsFall method outperforms the other methods in terms of accuracy, precision, sensitivity and specificity values.**

***Index Terms*—Biomedical signal processing, smart home care, fall detection, classification.**

I. Introduction

DEVISING technologies for fall detection and prevention is crucial in elder care systems, since sudden falls are considered the leading cause of injury and accidental death for seniors [1]. Fall is considered to be an uncontrolled, unintentional and sudden change of posture [2]. So far, several methods based on wearable devices, smart-phone sensors and video cameras have been developed for detecting falls [3], [4]. Radar-based fall detection has recently gained much attention [4], [5] due to the fact that this technology is privacy-friendly and non-contact unlike the techniques based on video-cameras and wearable sensors [3].

A wavelet-based approach was devised in [6] for fall detection purpose using Doppler radar. In [7], a fall detection scheme was presented by exploiting time-frequency characteristics of radar Doppler signatures. In [8], frequency domain features such as cepstrum coefficients were extracted from the radar signal in a fall detection approach. In [9], [10], several frequency domain features as well as the range information were used to develop a fall detection technique. In [11] and [12], time-frequency domain analysis was presented for human posture classification and fall detection by extracting a number of features. All these features are manually engineered and in most of the cases require an expert's knowledge.

Few attempts have so far been made to design an automated feature extraction method from radar data. In [13], a radar signal recognition was proposed using a deep restricted Boltzmann machine. In [14] and [15], deep neural network approaches were presented to reduce dimensionality of the extracted features from the radar signals based on the stack auto-encoder. A gait-based human identification was presented in [16] by using an acoustic sensor and a deep neural network. In [17], the use of transfer learning was investigated in classifying activities using data collected from wearable sensors. A pre-trained model was used for feature extraction from radar data followed by linear or nonlinear support vector machine for classification. However, devising an automatic radar-based fall detection with minimum preprocessing required is still challenging and has not been proposed in the literature especially using an automatic feature extraction from the radar time-frequency representations. This can be accomplished via deep learning which provides a tool for automatic feature extraction without the need for manual feature engineering and domain knowledge of the data.

In view of this and to automate feature extraction from radar data, in this paper, a new fall detection method, called CapsFall, is proposed by incorporating time-frequency representations derived from an ultra-wideband radar signals and leveraging the recent advances in deep neural networks. In particular, the proposed method is realized by adopting capsule network for extracting multi-level features from radar spectrograms. To this end, a radar time series is first obtained by summing up over all the range bins in the radar scattering matrix. A time-frequency representation of the radar time series is obtained and used as input to the capsule network for feature learning. Experiments are conducted at Health Devices Research Group Lab at the University of Ottawa to assess the performance of the proposed method and conduct performance comparison with that of other existing methods. For the purpose of comparison, other methods based on convolutional neural network, multi-layer perceptron, decision trees, naive Bayes and support vector machines are developed and the same time-frequency representations used with capsule networks are considered as inputs.

E-mail: {hsadreaz@uottawa.ca, mbolic@eecs.uottawa.ca, sreeraman.rajan@carleton.ca}

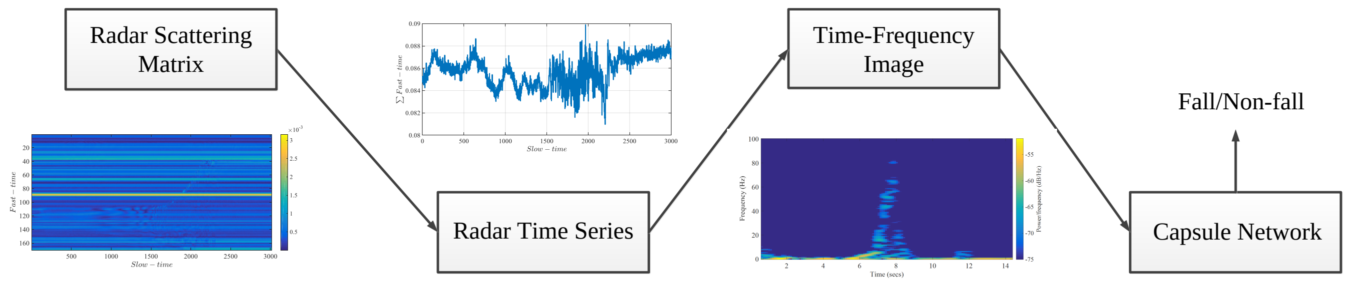


Fig. 1. CapsFall block diagram for radar-based fall detection.

TABLE I
NUMBER OF FALL AND NON-FALL ACTIVITIES PERFORMED IN OUR EXPERIMENTS BY DIFFERENT SUBJECTS.

Class	Number of Experiments
Fall	1870
Non-fall	1490

The paper is organized as follows. In Section II, the experimental setup and radar used for data acquisition is described. In Section III, the proposed CapsFall method is presented. Experimental results are included in Section IV. Finally, Section V concludes the paper.

II. EXPERIMENTAL SETUP AND MEASUREMENT

Xethru X4M03 kit [18] was used in our experiments for collecting data. The ultra-wideband radar operates in $5.9-10.3$ GHz, providing high spatial resolution. The radar is placed 1.5 m above the floor level. The radar system transmits a radio frequency electromagnetic signal and receives reflections from the subject under study and other objects in the room, i.e., clutters. In particular, in our experiments, each scan is repeated for 15 seconds and digitized at a rate of 200 samples/second (slow-time). The range of the radar used in this study is 10 m with a range resolution of 5.35 cm, resulting in 189 range bins. The radar time series signal is obtained via summing up the signals in these range bins.

A number of data samples were collected according to the pre-approved protocol by the University of Ottawa Research Ethics Board. 10 healthy male subjects aging from 20 to 35 participated in data collection process, performing several activities such as standing or walking toward radar, with or without stumbling and body movements, falling down at different distances and angles to the radar line of sight considering the transitioning from other activities, lying down with or without movements and side rolling, standing up from lying position in different distances and angles to the radar line of sight. The number of fall and non-fall activities performed in our experiments are given in Table I. The tests were performed in different locations throughout the room so that the proposed algorithm can be made invariant to location and relative angle.

III. CAPSFALL

In this section, the proposed CapsFall method is presented based on automatic feature extraction and classification using capsule network. Fig. 1 shows block diagram of the proposed CapsFall method.

A. Time-Frequency Analysis

In the experiments, the radar scattering matrix $\mathbf{X} = [x_{i,j}] \in \mathbb{R}^{m \times n}$, i.e., the received radar signal, is recorded, where n columns represent the spatial samples from different ranges (fast-time) and m rows correspond to observations recorded at different time intervals (slow-time). From the scattering matrix, columns are summed up resulting in a radar time series $\mathbf{x}$. To analyze the radar time series signals, a joint time-frequency representation is obtained by applying the short-time Fourier transform (STFT), given by $STFT[o,k] = \sum_{r=-\infty}^{\infty} x[r]W[r-o]\exp\left(-j2\pi rk/N\right)$, where $W[.]$ is a finite length sliding window function, o is the time index, $k = 0, 1, ..., N-1$ is the frequency index and N is the number of frequency points. The squared magnitude of STFT yields the spectrogram, i.e., $SP(o,k) = |STFT[o,k]|^2$. The image intensities indicate the energy corresponding to the micro motion signature at each time instant [11]. In the experiments, the STFT is applied using a Hamming window of size 256 samples with 80% overlap between two adjacent windows [12]. Fig. 2 depicts examples of radar time series as well as their corresponding spectrograms for falling down and standing transitions, respectively. It is seen from this figure that the energy content of these activities are distinguishable in their time-frequency representations.

The resulting spectrogram are used as input to a capsule network for automatic feature learning and classification.

B. Automatic Feature Learning

A common approach for a fall detection approach is to extract a set of features from the radar data [6], [7], [9]–[11]. The performance of any detection method is known to be highly reliant on the type of features extracted. Deciding on relevant features can be a tedious task. In addition, these features exhibit large variations due to size, habits, and health conditions of the monitored individuals. A possible solution to overcome these issues is to devise an automated approach

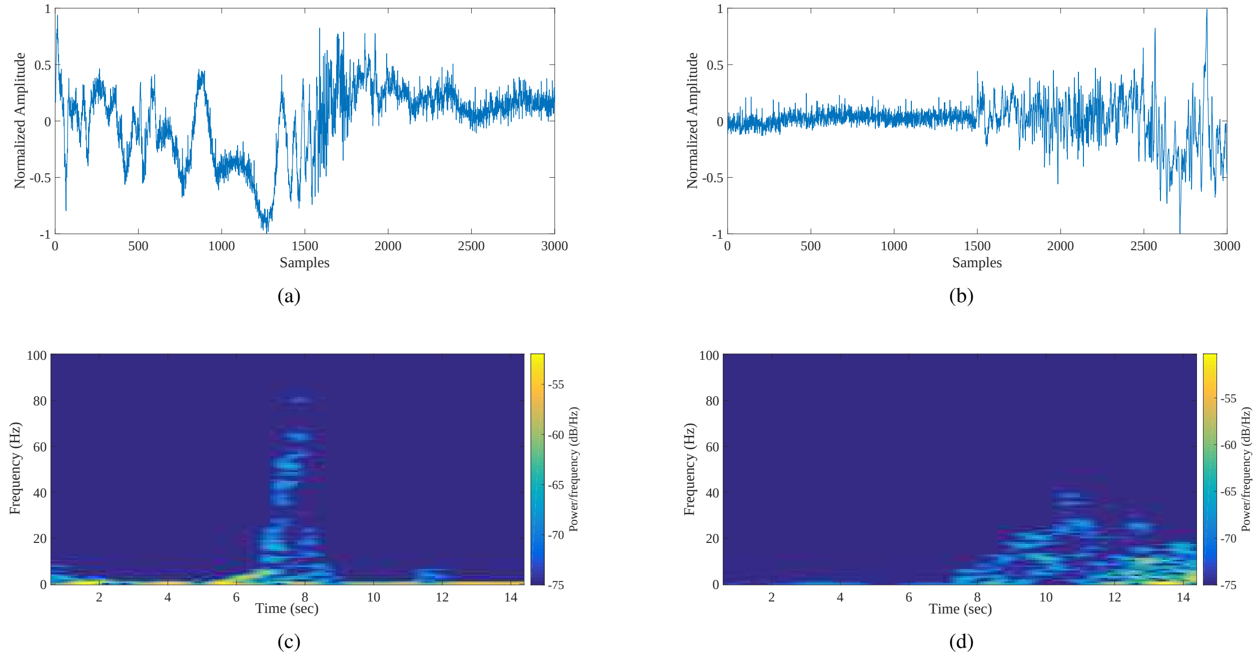


Fig. 2. Radar time series signal associated with (a) Fall after walking toward radar, (b) Stand up from a lying position and (c)-(d) their corresponding time-frequency representations.

that captures the intricate properties of the human motion signatures. In recent years, deep learning [19] has taken feature engineering to the next level by automating feature learning and avoiding strategic-thinking to invent feature abstractions. Deep learning approaches can work independently of or in tandem with other feature selection methods. Leveraging the recent deep learning advances, in the proposed CapsFall method, capsule network [20] is applied to the time-frequency representations derived from the radar time series for automatic feature learning and classification.

C. Capsule Network

The capsule network is selected over convolutional neural network (CNN), since it provides a better hierarchical image representation by passing an image through multiple layers of the network without using any pooling layer which results in the amount of detail to be processed at different layers to be unchanged. In addition, capsule network is known to better deal with small amounts of labeled data and class-imbalance than CNNs [20]. The capsule network is formed of two layers: a primary capsule layer, capturing low-level features, followed by a secondary capsule, capable of predicting the presence of specific patterns in the image.

It is noted that in the capsule network, the weight optimization is not through the common backpropagation approach but through a routing-by-agreement algorithm as proposed in [20]. The routing-by-agreement algorithm establishes a connection between lower and higher-level feature maps in such a way that a primary capsule (child) sends its input to the secondary capsule (parent) that better agrees with its input. In other words, after a number of iterations, each parents' outputs may converge to predictions of some children and diverge from those of others. Parent capsules s_j are being fed by the sum of predictions from all the children capsules as

$$\mathbf{s}_j = \sum c_{ij}\hat{\mathbf{u}}_{j|i}, \tag{1}$$

where $\hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij}\mathbf{u}_i$ is the prediction vector by a capsule i (child) for the capsule j (parent), which predicts the output of the parent capsule, and c_{ij} is a coupling coefficient given by a routing softmax as

$$c_{ij} = \frac{\exp{(b_{ij})}}{\sum_k \exp{(b_{ik})}}, \tag{2}$$

where b_{ij} is the prior log probability for the routing, i.e., the prior probability that capsule i should connect to capsule j in a higher level. The weights $\mathbf{W}_{ij}$ allow learning feature relationships instead of detecting independent features by filtering at different scale portions of the image.

Unlike CNNs, instead of passing the scalar outputs of the convolution filters through the rectified linear unit activation function, i.e., $f(x) = \max(0, x)$, capsule network builds tensors by grouping multiple feature channels, i.e. the output of each neuron is expanded from a scalar to a vector. The non-linearity used in capsule network is a squashing function and is given by

$$\mathbf{z}_j = \frac{\|\mathbf{s}_j\|^2}{1 + \|\mathbf{s}_j\|^2}\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|} \tag{3}$$

The magnitude of the resulting vector is regarded as the probability of the presence of a capsule's entity, i.e., patterns in time-

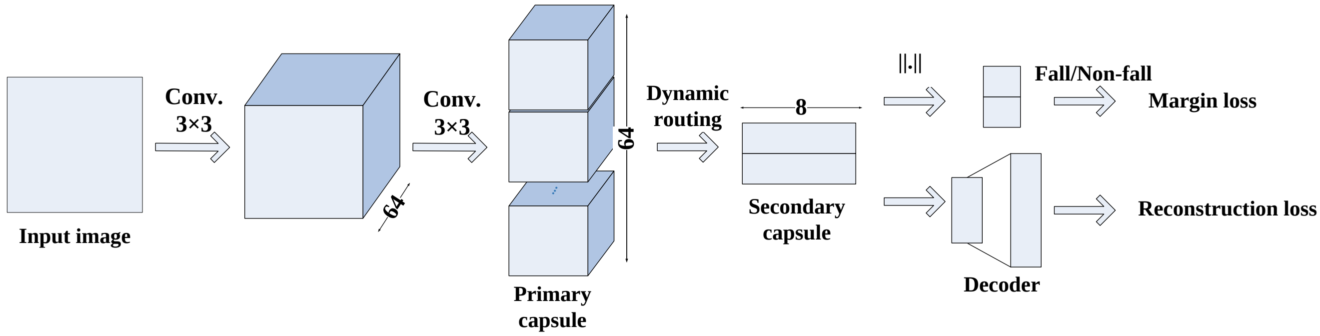


Fig. 3. Block diagram of the capsule network used in the proposed CapsFall method.

frequency representations. The prior log probabilities b_{ij} are then updated for a number of iterations by $b_{ij} = b_{ij} + \mathbf{z}_j \hat{\mathbf{u}}_{j|i}$. For the output layer, unlike CNNs which use a softmax activation and cross-entropy loss function, the capsule network computes a margin loss for the secondary capsule layers as

$$L_r = y_r \max\left(0, m^+ - \|\mathbf{z}_r\|\right)^2 + 0.5\,(1 - y_r)\max\left(0, \|\mathbf{z}_r\| - m^-\right)^2 \tag{4}$$

where y_r is the label for class r. It is noted that class r is predicted if $\|\mathbf{z}_r\| > m^+$, and is not predicted if $\|\mathbf{z}_r\| < m^-$. The parameters m^+ and m^- were experimentally found and set to 0.9 and 0.1, respectively [20]. The margin loss forces the class instances to be close to each other.

In addition, capsule network uses a decoder for regularization which is comprised of fully-connected layers. The loss value of this decoder is computed as

$$L = \left\|\mathbf{S} - \hat{\mathbf{S}}\right\|^2, \tag{5}$$

where $\mathbf{S}$ and $\hat{\mathbf{S}}$ are the input image and its reconstructed version. The total loss value of the capsule network is obtained as the weighted average of the margin loss and reconstruction loss. The sigmoid activation is used in the output layer, where the binary cross-entropy loss function is used to measure the error. The error is then minimized through backpropagation using the Adam optimizer.

It is also noted that to realize a better generalization of the model, early-stopping approach is employed, i.e., training continues to the point where validation accuracy starts to decrease. It is noted that the hyperparameters used in the proposed fall detection method were selected using a grid search cross validation technique [25], where the objective was to maximize the classification accuracy of the validation set. To this end, the training set was divided into three folds; two for training and one for validation. The goal was to minimize the cross validation loss.

In the following, summary of the different layers of the proposed CapsFall model are presented.

- The spectrograms of size 64×68 are used as input to the capsule model.
- The second layer is comprised of a convolutional layer having 64 filters of size 3×3 with a stride of 1, which leads to 64 feature maps of size 62×66.
- The third layer is a primary capsule layer including 8 convolutional filters of size 3×3 with strides of 2. This layer consists of 8 capsules component with dimension of 8, resulting in 64 feature maps of size 30×32.
- The final capsule layer includes 2 capsules of length 8, i.e., class capsules, corresponding to fall and non-fall activities.
- The decoder is composed of fully-connected layers having 512, 1024 and 4352 neurons, respectively, where the input image can be reconstructed from the final fully-connected layer and the sum of squared differences between the input image and its reconstructed version is minimized.

IV. Experimental Results

Experiments were conducted on a set of radar data collected in a room environment at University of Ottawa, as discussed in Section II, to evaluate the performance of the proposed fall detection method[1]. In the proposed CapsFall method, the radar scattering matrix is first processed to obtain the radar time series. Time-frequency representation, i.e., spectrogram, of the radar time series is then obtained and used as input to the capsule network. The spectrogram contains the energy content of a particular activity, i.e., a fall incident results in an instantaneous high frequency content with specific distribution of energy over time, whereas non-fall activity exhibits lower frequency peaks and a different energy distribution. A total of 3360 spectrograms obtained from different activities were used in the experiments of the proposed work.

All of the deep learning tasks are implemented using Keras that is backended by TensorFlow package. In addition to the proposed method, support-vector machine (SVM) [8], [9], decision tree (DT), Gaussian naive Bayes (GNB), multi-layer perceptron (MLP) and CNN [1], [8] are also implemented for comparison purposes. In the case of MLP, following the method in [14], two layers of 300 and 150 neurons with

[1]The dataset and code used in this research work will be made publicly available.

TABLE II

ACCURACY, PRECISION, SPECIFICITY VALUES C OBTAINED USING THE PROPOSED CAPSFALL METHOD AS WELL AS THOSE OBTAINED USING LSVM, GSVM, DT, GNB, MLP AND CNN, WHEN 70% OF DATA ARE CONSIDERED FOR TRAINING.

	Accuracy	Precision	Sensitivity	Specificity
LSVM	79.41∓1.02	82.01∓1.77	80.91∓1.36	77.57∓1.47
GSVM	86.69∓0.83	90.27∓1.04	86.27∓1.01	87.32∓1.35
DT	75.40∓1.34	78.56∓2.23	77.23∓1.44	73.13∓2.31
GNB	79.70∓1.18	78.99∓1.70	83.32∓1,23	75.76∓1.47
MLP	88.61∓1.47	88.61∓2.15	88.39∓0.95	88.94∓2.57
CNN	92.84∓1.56	94.53∓1.49	92.58∓1.77	93.42∓1.66
CapsFall	**94.22∓1.21**	**95.66∓1.07**	**93.99∓1.32**	**94.55∓1.34**

l_2 regularization are considered. In addition, the softmax regression classifier is used in the output layer. In the case of CNN, for a fair comparison, a similar structure to the capsule network is considered. In particular, two convolutional layers followed by two fully-connected layers are considered. The convolutional layers respectively have 64 and 8 kernels of size 3×3, each followed by a non-overlapping 2×2 max-pooling layer. The two fully-connected layers have 1024 and 512 neurons. Rectified linear unit activation function is used for all the convolutional and fully-connected layers, whereas the softmax function is used for the output layer to provide the probability of the predicted classes. For the sake of consistency, the number of epochs is set to 100 for the MLP and CNN. The training set is divided into batches each of size 50, except for the last batch. Thus each epoch has 67 iterations (backpropagations).

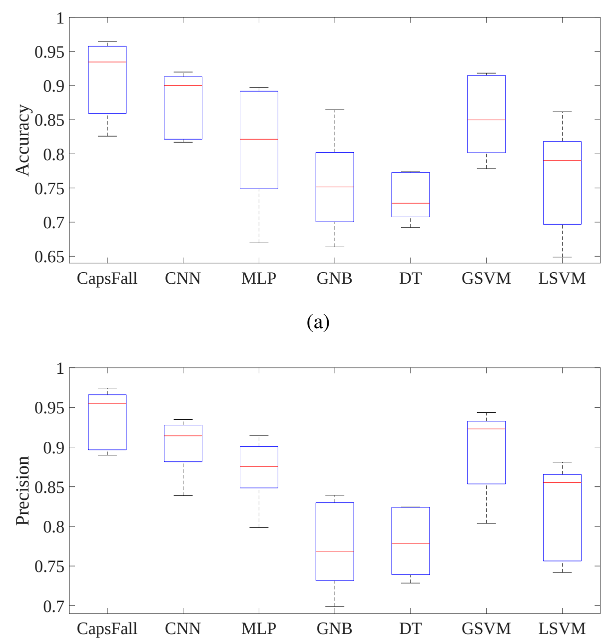
In the case of DT, gini index is used with best split at each node [26]. In case of SVM, the resulting spectrogram obtained in Section III.C are vectorized and used as input to these classifier. Both the linear (LSVM) and kernel-based SVM (GSVM) are employed [27], with squared hinge loss and l_2 regularization for the LSVM and $C = 1$ and $\gamma = 0.001$ for the GSVM.

In the first experiment, the classifiers are trained using the spectrograms for 70% of the data and tested using the remaining data. The results are averaged over 10 runs. Table II gives mean and standard deviation of classification metrics obtained using the proposed method and those of the other methods. It is seen from this table that the proposed CapsFall method outperforms the other methods by providing higher classification rates on the test set in all the cases. In particular, the proposed method achieves 94.22% accuracy, 95.66% precision, and 93.99% sensitivity, which are higher than those yielded by the other methods. Noticeably, the proposed method based on capsule network performs better than its closest counterparts, i.e., CNN-based method, in detecting fall incidents with almost 1.5% increase in accuracy and sensitivity values. The superior performance of the proposed method using capsule network is due to the fact that the structure of the capsule network realizes a more accurate learning of the existing shapes in the time-frequency representation of the radar data and their local relationships. In addition, the capsule network may be generalized better on the unseen data.

TABLE III

ACCURACY, PRECISION, SENSITIVITY, SPECIFICITY VALUES OBTAINED USING THE PROPOSED CAPSFALL METHOD AS WELL AS THOSE OBTAINED USING LSVM, GSVM, DT, GNB, MLP AND CNN, IN A 5-FOLD CROSS VALIDATION.

	Accuracy	Precision	Sensitivity	Specificity
LSVM	76.34	81.99	78.16	75.63
GSVM	85.39	89.39	85.70	85.87
DT	73.57	77.96	75.76	71.14
GNB	74.48	77.52	79.58	71.80
MLP	81.07	87.01	81.76	81.58
CNN	87.41	90.17	87.49	87.26
CapsFall	**91.01**	**93.63**	**90.38**	**91.78**



(a)
(b)

Fig. 4. Box plots for classification metrics of various methods in 5-fold cross-validation. (a) Accuracy, (b) Precision.

Table III gives classification metrics obtained using the proposed CapsFall method and those of the other existing methods. The classifiers are trained using four folds of the radar data and tested using the data from the remaining fold in a 5-fold cross validation, i.e., one fold is regarded as testing data and the rest are used for training. The process is repeated five times and results are then averaged over all folds. It is seen from this table that the CapsFall outperforms the other methods by providing higher detection rates in presence of an unseen set of data. In particular, the proposed method achieves 91.01% accuracy, 93.63% precision, 90.38% sensitivity and 91.78% specificity, which are higher than those of the other methods.

The statistical significance of the proposed method is also investigated. To this end, one way analysis of variance (ANOVA) is used, which compares the amount of variation between groups with the amount of variation within groups. More specifically, it tests the hypothesis that the samples

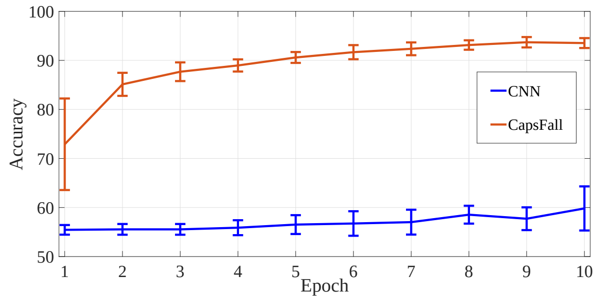


Fig. 5. Comparison between the classification accuracy of the proposed CapsFall method and traditional CNN-based classifier in the first 10 training iterations.

in 5-fold classification accuracy of each method are drawn from populations with the same mean against the alternative hypothesis that the population means are not all the same. The between-groups variation, F-statistic and p-value are respectively equal to 0.13325, 4.76 and 0.0018. The p-value is the probability that the test statistic can take a value greater than the value of the computed test statistic, i.e., $P(F > 4.76)$. The small p-value indicates that differences between column means are significant. In addition, a multiple comparison test is also performed to determine which pairs of group means are significantly different. It is observed that the proposed CapsFall method provides mean accuracy value which is significantly different from SVM, DT and GNB. Yet, against CNN and MLP, p-value is equal to 0.97 and 0.28, respectively, indicating that a significant difference does not exist between the proposed method and these two methods.

Fig. 4 shows box plots for classification metrics of various methods. It is seen from this figure that the proposed method using convolutional neural network provides higher accuracy and precision on a 5-fold cross validation. In order to validate the effectiveness of the proposed CapsFall method, we depict the classification accuracy of the proposed method and that of the CNN for the first 10 training iterations (epochs). Fig. 5 shows comparison between the two methods when the classification accuracy is averaged over ten runs. It is seen from this figure that the proposed CapsFall method provides significantly higher classification accuracy than the method based on CNN in only few epochs and can converge much faster. In other words, using the proposed method, higher number of fall incidents can accurately be detected even with lower number of training iterations.

We now investigate the performance of the proposed Capsfall method and CNN-based method, when small amounts of labeled data is used. To this end, the classifiers are trained using the spectrograms for 5% and 10% of the data and tested using the remaining data. The results are averaged over 10 runs. Table IV gives classification accuracy for the proposed fall detection method and that of the CNN-based method. It is seen from this table that the proposed CapsFall is capable of providing high classification accuracy, even when the network is trained with small number of labeled training samples.

To investigate the hyperparameter sensitivity, we obtain the classification accuracy of the proposed CapsFall method

TABLE IV
ACCURACY VALUES (MEAN∓ STANDARD DEVIATION) OBTAINED USING THE PROPOSED CAPSFALL METHOD AS WELL AS THAT OBTAINED USING CNN, WHEN 5% AND 10% OF THE DATA ARE USED FOR TRAINING.

Accuracy	CNN	**CapsFall**
5%	55.76∓0.37	**77.98∓1.69**
10%	74.81∓2.31	**86.97∓1.42**

TABLE V
HYPERPARAMETER (BATCH SIZE AND LEARNING RATE) SENSITIVITY ANALYSIS OF THE PROPOSED CAPSFALL METHOD.

Learning rate \ Batch size	50	100	150	200
5×10^{-3}	93.96	94.79	83.28	70.42
2×10^{-3}	95.62	95.00	94.17	93.79
10^{-3}	94.79	95.83	95.43	95.21
5×10^{-4}	95.42	95.21	94.58	94.79
10^{-4}	94.38	92.92	93.33	92.71

when the hyperparameters, i.e., batch size and learning rate, are varying. To this end, the dataset is divided into training (2880 samples) and test (480 samples). Table V gives the classification accuracy of the proposed method when batch size and learning rate take values in $[50, 100, 150, 200]$ and $[5 \times 10^{-3}, 2 \times 10^{-3}, 10^{-3}, 5 \times 10^{-4}, 10^{-4}]$, respectively. It is seen from this table that the optimum values for batch size and learning rate are 100 and 10^{-3}, providing the highest classification accuracy for the proposed method.

V. CONCLUSION

A new fall detection method, CapsFall, has been proposed using an ultra-wideband radar and supervised learning approach based on capsule network. The radar time series has been derived from the radar returns and time-frequency representation has been obtained using the short-time Fourier transform to obtain spectrograms for different activities. The resulting spectrograms have been fed into the capsule network for automatic feature learning. Experiments have been conducted to assess the performance of the proposed CapsFall method and to compare it with that of the other existing works. The results have shown that CapsFall method outperforms the other methods in terms of providing higher classification metrics. The proposed capsule network-based method has demonstrated the ability to circumvent feature engineering and tedious preprocessing adopted by traditional approaches. It has been observed that the proposed fall detection method using the capsule network, fewer number of training iterations are needed, and still a significantly higher number of fall incidents can accurately be detected when compared to that using the convolutional neural network. It has also been observed that the proposed method using the capsule network can provide better classification accuracy when using a small number of labeled training data than CNN-based method.

VI. ACKNOWLEDGMENT

We would like to thank NSERC for funding this research through Idea to Innovation program. We would also like to thank Xander May, Abdullah Al-kaf and Hamza Qassoud for conducting experiments and collecting data, and Dr. Shervin Shirmohammadi and his research Lab (DISCOVER Lab) at the University of Ottawa for providing us with the lab space for performing the experiments.

References

[1] M. Yu, A. Rhuma, S. M. Naqvi, L. Wang and J. Chambers, "A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment," *IEEE Transactions on Information Technology in Biomedicine,* vol. 16, no. 6, pp. 1274-1286, 2012.

[2] [Online]. Available: http://www.who.int/violence-injury-prevention/other-injury/falls/en/

[3] A. T. Ozdemir and B. Barshan, "Detecting falls with wearable sensors using machine learning techniques," *Sensors,* vol. 14, pp. 10691-10708, 2014.

[4] H. Sadreazami, S. Rajan and M. Bolic, "On the use of ultra-wideband radar and stacked LSTM-RNN for at home fall detection," in *Proc. Life Sciences Conference (LSC),* pp. 1-4, 2018.

[5] I. Nejadgholi, H. Sadreazami, S. Rajan and M. Bolic, "Classification of Doppler radar reflections as preprocessing for breathing rate monitoring," *IET Signal Processing,* vol. 13, no. 1, pp. 21-28, 2019.

[6] B. Y. Su, K. Ho, M. J. Rantz and M. Skubic, "Doppler radar fall activity detection using the wavelet transform," *IEEE Transaction on Biomedical Engineering,* vol. 62, no. 3, pp. 865-875, 2015.

[7] Q. Wu, Y. D. Zhang, W. Tao and M. G. Amin, "Radar-based fall detection based on Doppler time-frequency signatures for assisted living," *IET Radar, Sonar & Navigation,* vol. 9, no. 2, pp. 164-172, 2015.

[8] L. Liu, M. Popescu, M. Skubic, M. Rantz, T. Yardibi and P Cuddihy, "Automatic fall detection based on Doppler radar motion signature," in *Proc. International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth),* pp. 222-225, 2011.

[9] B. Erol and M. G. Amin, "Fall motion detection using combined range and Doppler features," in *Proc. IEEE European Signal Processing Conference (EUSIPCO),* pp. 2075-2080, 2016.

[10] B. Erol and M. G. Amin, "Effects of range spread and aspect angle on radar fall detection," in *Proc. IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM),* pp. 1-5, 2016.

[11] A. Gadde, M. G. Amin, Y. D. Zhang and F. Ahmad, "Fall detection and classifications based on timescale radar signal characteristics," in *Proc. SPIE 9077 Radar Sensor Technology XVIII,* pp. 907712, 2014.

[12] L. Stankovic, M. Dakovic and T. Thayaparan, *Time-frequency signal analysis with applications,* Artech House, 2013.

[13] D. Zhou, X. Wang, Y. Tian and R. Wang, "A novel radar signal recognition method based on a deep restricted Boltzmanm machine," *Engineering Review,* vol. 37, no. 2, pp. 165-171, 2017.

[14] B. Jokanovic, M. G. Amin and F. Ahmad, "Radar fall motion detection using deep learning," in *Proc. IEEE Radar Conf. (RadarConf),* pp. 1-6, 2016.

[15] B. Jokanovic, M. G. Amin and F. Ahmad, "Effect of data representations on deep learning in fall detection," in *Proc. IEEE Sensor Array Multichannel Signal Process. Workshop (SAM),* pp. 1-5, 2016.

[16] Y. Wang, Y. Chen, Md. Z. A. Bhuiyan, Y. Han, Sh. Zhao and J. Li, "Gait-based Human identification using acoustic sensor and deep neural network," *Future Generation Comp. Sys.,* vol. 86, pp. 1228-1237, 2018.

[17] D. Wagner, K. Kalischewski, J. Velten and A. Kummert, "Activity recognition using inertial sensors and a 2D convolutional neural network," in *Proc. 10th Int. Workshop on Multidimensional Systems,* pp. 1-6, 2017.

[18] Novelda's XeThru X4M03. [Online]. Available: https://www.xethru.com/x4-radar-chip.html/

[19] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature,* vol. 521, no. 7553, pp. 436-444, 2015.

[20] S. Sabour, N. Frosst and G. E. Hinton, Dynamic routing between capsules; Advances in Neural Information Processing Systems 30, Curran Associates, Inc., pp. 3856-3866, 2017.

[21] G. E. Hinton, S. Osindero and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural comput.,* vol. 18, no. 7, pp. 1527-1554, 2006.

[22] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. International Conference on Learning Representations (ICLR),* pp. 1-13, 2015.

[23] H. Huihui and R. Xia, "Joint binary neural network for multi-label learning with applications to emotion classification," *arXiv preprint arXiv:1802.00891*, 2018.

[24] O. C.-Reyes, P. Scully and K. B. Ozanyan, "Deep neural networks for learning spatio-temporal features from tomography sensors," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 1, pp. 645-653, 2018.

[25] Pedregosa et al., Scikit-learn: "Machine Learning in Python," *JMLR 12,* pp. 2825-2830, 2011.

[26] T. Hastie, R. Tibshirani and J. Friedman, *Elements of statistical learning,* Springer, 2009.

[27] C. Cortes and V. Vapnik, "Support vector machine," *Machine learning,* vol. 20, no. 3, pp. 273-297, 1995.