

Development of a Machine Learning Based Fall Detection System for the Elderly and Disabled

by Farhan Ahnaf Rashid

Thesis submitted in fulfilment of the requirements for
the degree of

Master of Engineering (Research)

under the supervision of Dr. Xiaoying Kong

University of Technology Sydney
Faculty of Faculty of Engineering and Information
Technology

December 2021

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Farhan Ahnaf Rashid declare that this thesis, is submitted in fulfilment of the requirements for the award of Master of Engineering (Research), in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed
prior to publication.

Signature of Student: Farhan Ahnaf Rashid

Date: 20 December 2021

ACKNOWLEDGMENT

I wish to express my deepest gratitude to my first principal supervisor: Assoc. Prof. Dr. Kumbesan Sandrasegaran (Retired), my present principal supervisor: Dr. Xiaoying Kong and co-supervisor: Dr. Gengfa Fang for their encouragement to start this work and for the opportunity to be a member of the inspiring research group. Their continuous support and constructive criticism have been precious during this works.

My thanks go to all non-academic staffs of SEDE HDR team of UTS for their valuable communication during the COVID-19 pandemic.

My warmest thanks belong to my parents AKM Harun Ar Rashid and Prof. Dr. Fazilatun Nessa for their confidence in me and for being always so supportive and interested in my work and well-being and for providing unfailing support to finish this work. I wish to extend thanks to my mother for her inspiring suggestions during writing this thesis.

RELATED PUBLICATION/ RESEARCH PAPERS PRESENTED AT CONFERENCES

Farhan Ahnaf Rashid, Kumbesan Sandrasegaran and Xiaoying Kong, “Simulation of SisFall Dataset for Fall Detection Using MATLAB Classifier Algorithms”. In 12th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP’2021), December 10-12, 2021, Xian, China. (Paper published in the conference proceeding).

Farhan Ahnaf Rashid, Kumbesan Sandrasegaran and Xiaoying Kong, “Simulation of MobiFall Dataset for Fall Detection Using MATLAB Classifier Algorithms” In 14th International Conference on Developments in eSystems Engineering (DeSE 2021), December 7-10, 2021, Sharjah, UAE. (Paper published in the conference proceeding).

TABLE OF CONTENTS

CERTIFICATE OF ORIGINAL AUTHORSHIP	II
ACKNOWLEDGMENT	III
RELATED PUBLICATION/ RESEARCH PAPERS PRESENTED AT CONFERENCES	IV
LIST OF FIGURES	VIII
LIST OF TABLES	XII
LIST OF ACRONYMS	XIV
ABSTRACT.....	XV
1 INTRODUCTION.....	1
1.1 Fall Event	1
1.2 Motivation	2
1.3 Aims, objectives and significance	3
1.3.1 Aims	3
1.3.2 Objectives.....	3
1.3.3 Significance.....	3
2 LITERATURE REVIEW	5
2.1 Fall detection methods.....	6
2.1.1 Image processing approach.....	7
2.1.2 Location sensor method	8
2.1.3 Accelerometer in smartphones.....	10
2.1.4 Wristband and smartwatches.....	11
2.1.5 RFID technology	12
2.1.6 Surface electromyography technology.....	12
2.1.7 Low-cost accelerometer	13
2.1.8 Calculations for processing accelerometer output	17
2.2 Navigation systems.....	20
2.3 Commercially available fall detection systems	21
2.4 Machine learning algorithms.....	23
2.4.1 Support Vector Machine (SVM).....	24
2.4.2 Artificial Neural Network (ANN).....	25
2.4.3 Naïve Bayesian Network.....	26
2.4.4 Random Forest method	27
2.4.5 k-Nearest Neighbors (KNN) algorithm.....	29
2.4.6 Hidden Markov Model (HMM)	30
3 METHODOLOGY	31
3.1 Requirements.....	31

3.2	System design.....	32
3.3	Dataset acquisition for simulation.....	33
3.3.1	SisFall dataset	33
3.3.2	MobiFall dataset.....	35
3.4	Dataset feature selection.....	36
3.5	Algorithms selected for simulation	38
3.6	Cross-validation.....	38
3.7	Experiment for algorithm simulation	39
3.8	Dataset extraction for training.....	41
3.8.1	Training data classification	42
3.8.2	Saving results	47
3.9	Dataset extraction for testing.....	48
3.9.1	Testing data classification	50
3.9.2	Saving results	51
3.10	Ensemble method.....	52
3.11	Visualizing results with confusion matrix	55
4	SIMULATION RESULTS AND DISCUSSION.....	56
4.1	Experiment 1 (using SisFall’s ADXL345 accelerometer model and 5-fold cross-validation)	56
4.1.1	Experiment 1.1 (classifying multiple ADLs using 10 features).....	56
4.1.2	Experiment 1.2 (classifying multiple ADLs using 16 features).....	58
4.1.3	Experiment 1.3 (classifying multiple ADLs and Falls using 16 features)	60
4.1.4	Experiment 1.4 (classifying only an instance of ADL and Fall using 16 features).....	62
4.2	Experiment 2 (using SisFall’s ADXL345 accelerometer model and 10-fold cross-validation).....	64
4.2.1	Experiment 2.1 (classifying multiple ADLs using 10 features).....	64
4.2.2	Experiment 2.2 (classifying multiple ADLs using 16 features).....	66
4.2.3	Experiment 2.3 (classifying multiple ADLs and Falls using 16 features)	67
4.2.4	Experiment 2.4 (classifying only an instance of ADL and Fall using 16 features).....	69
4.3	Experiment 3 (using SisFall’s MMA8451Q accelerometer model and 10-fold cross-validation).....	70
4.3.1	Experiment 3.1 (classifying multiple ADLs using 10 features).....	70
4.3.2	Experiment 3.2 (classifying multiple ADLs using 16 features).....	72
4.3.3	Experiment 3.3 (classifying multiple ADLs and Falls using 16 features)	74
4.3.4	Experiment 3.4 (classifying only an instance of ADL and Fall using 16 features).....	76

4.4	Experiment 4 (using MobiFall’s LSM330DLC model and 10-fold cross-validation)	77
4.4.1	Experiment 4.1 (classifying multiple ADLs using 10 features).....	77
4.4.2	Experiment 4.2 (classifying multiple ADLs using 16 features).....	78
4.4.3	Experiment 4.3 (classifying multiple ADLs and Falls using 16 features)	80
4.4.4	Experiment 4.4 (classifying only an instance of ADL and Fall using 16 features).....	81
4.5	Brief comparison of Experiments 1 to 4.....	83
4.6	Testing simulation results and comparison with trained results.....	84
4.6.1	Experiment 1 (using SisFall’s ADXL345 model and 5-fold cross-validation)	84
4.6.2	Experiment 2 (using SisFall’s ADXL345 model and 10-fold cross-validation)	89
4.6.3	Experiment 3 (using SisFall’s MMA8451Q model and 10-fold cross-validation)	94
4.6.4	Experiment 4 (using MobiFall’s LSM330DLC model and 10-fold cross-validation)	99
4.7	Results achieved using the ensemble classifier	103
5	CONCLUDING DISCUSSION AND FUTURE SUGGESTIONS	108
6	APPENDIX	114
6.1	The following shows an instance of the code for training Fine KNN for Experiment 2.4	114
6.2	The following shows an instance of the code for training Cubic SVM for Experiment 4.1	116
6.3	The following shows an instance of the code for saving the training results into readable form and extracting the results for external processing.....	118
6.4	The following shows an instance of the code for saving the testing results into readable form and extracting the results for external processing.....	121
6.5	The following shows an instance of the code for the stacked ensemble classifier system model.....	125
7	REFERENCES	130

LIST OF FIGURES

Figure 1.1: Example of a fall detection system [7]	2
Figure 2.1: Example of a fall detection system.....	7
Figure 2.2: Example of a fall detection system.....	9
Figure 2.3: Signal output of an accelerometer during a Fall Event	18
Figure 2.4: Graph showing the hyperplanes	24
Figure 2.5: Artificial Neural Network.....	25
Figure 2.6: Naïve Bayesian method.....	26
Figure 2.7: Random Forest method.....	27
Figure 2.8: Binary Tree	28
Figure 2.9: KNN example as a diagram showing the data classes and datapoint “d”	29
Figure 2.10: Hidden Markov Model	30
Figure 3.1: Flowchart of the project.....	31
Figure 3.2: Diagram of the proposed system	32
Figure 3.3: System data flow for the system.....	33
Figure 3.4: Raw data for walking slowly (D01)	35
Figure 3.5: Raw data for fall with front knees lying (accelerometer).....	36
Figure 3.6: Example of 5-fold cross-validation	39
Figure 3.7: An instance of text files for D01 to be read being initialised	41
Figure 3.8: Code to read data from text	41
Figure 3.9: An instance of variables extracted for the ADL D01 in a table	41
Figure 3.10: Tabulating all data together for ADXL345	41
Figure 3.11: Code to read data from text	42
Figure 3.12: Tabulating all data together for MMA8451Q.....	42
Figure 3.13: Initialise predictors and response for classification.....	42
Figure 3.14: Setting up classification type and properties	43
Figure 3.15: Setting up the predict function.....	43
Figure 3.16: An instance of k-fold cross-validation.....	43
Figure 3.17: Setting up classification type and properties	44
Figure 3.18: Setting up Linear SVM classification type and properties	45
Figure 3.19: Setting up Quadratic SVM classification type and properties.....	45
Figure 3.20: Setting up Fine Gaussian SVM classification type and properties.....	45
Figure 3.21: Setting up Fine KNN classification type and properties	46

Figure 3.22: Setting up Cosine KNN classification type and properties	46
Figure 3.23: Setting up Cubic KNN classification type and properties	46
Figure 3.24: Saved results of the simulation	47
Figure 3.25: An instance of the tabulated results for Experiment 1.1.....	47
Figure 3.26: An instance of text files for D01 to be read.....	48
Figure 3.27: Code to read data from text	48
Figure 3.28: The mean, maximum, minimum extracted for the ADL D01 in an array ..	48
Figure 3.29: Calculating the orientation of the signal and standard deviation	49
Figure 3.30: Variables extracted for the ADL D01 in a table.....	49
Figure 3.31: Tabulating all data together for ADXL345	49
Figure 3.32: An instance of text files to be read	50
Figure 3.33: Code to read data from text for MMA8451Q.....	50
Figure 3.34: Tabulating all data together for MMA8451Q.....	50
Figure 3.35: Test set data used for the trained classifier.....	50
Figure 3.36: Part of code for calculating recall, precision and specificity.....	51
Figure 3.37: Code for saving the results of the simulation	51
Figure 3.38: Code for tabulated results of the simulation.....	52
Figure 3.39: Stacked Ensemble Classifier	53
Figure 3.40: Part of base classifier code (I)	53
Figure 3.41: Part of base classifier code (II)	53
Figure 3.42: Part of base classifier code (III).....	54
Figure 3.43: Part of base classifier code (IV)	54
Figure 3.44: K-fold cross-validation scores	54
Figure 3.45: Stacked Ensemble initialization.....	55
Figure 3.46: Classifier Testing.....	55
Figure 3.47: Confusion matrix example	55
Figure 4.1: Confusion matrix for Fine KNN.....	57
Figure 4.2: Confusion matrix for Cubic SVM	58
Figure 4.3: Confusion matrix for Cubic SVM	60
Figure 4.4: Confusion matrix for Fine KNN.....	60
Figure 4.5: Confusion matrix for Cubic SVM	62
Figure 4.6: Confusion matrix for Fine KNN.....	62
Figure 4.7: Confusion matrix for Fine KNN.....	64

Figure 4.8: Confusion matrix for Fine KNN.....	66
Figure 4.9: Confusion matrix for Cubic SVM	67
Figure 4.10: Confusion matrix for Cubic SVM	69
Figure 4.11: Confusion matrix for Fine KNN.....	70
Figure 4.12: Confusion matrix for Fine KNN.....	72
Figure 4.13: Confusion matrix for Cubic SVM	74
Figure 4.14: Confusion matrix for Cubic SVM	75
Figure 4.15: Confusion matrix for Medium Gaussian SVM	77
Figure 4.16: Confusion matrix for Fine KNN.....	78
Figure 4.17: Confusion matrix for Fine KNN.....	80
Figure 4.18: Confusion matrix for Fine KNN.....	81
Figure 4.19: Confusion matrix for Medium Gaussian SVM	83
Figure 4.20: Confusion matrix for tested Fine KNN (Experiment 1.1).....	85
Figure 4.21: Confusion matrix for tested Cubic SVM (Experiment 1.2).....	86
Figure 4.22: Confusion matrix for tested Cubic SVM (Experiment 1.3).....	87
Figure 4.23: Confusion matrix for trained and tested Fine KNN (Experiment 1.4)	89
Figure 4.24: Confusion matrix for tested Fine KNN (Experiment 2.1).....	90
Figure 4.25: Confusion matrix for tested Cubic SVM (Experiment 2.2).....	91
Figure 4.26: Confusion matrix for tested Cubic SVM (Experiment 2.3).....	92
Figure 4.27: Confusion matrix for trained and tested Fine KNN (Experiment 2.4)	94
Figure 4.28: Confusion matrix for tested Fine KNN (Experiment 3.1).....	95
Figure 4.29: Confusion matrix for tested Cubic SVM (Experiment 3.2).....	96
Figure 4.30: Confusion matrix for tested Cubic SVM (Experiment 3.3).....	97
Figure 4.31: Confusion matrix for tested Cubic SVM (Experiment 3.4).....	99
Figure 4.32: Confusion matrix for tested Medium Gaussian SVM (Experiment 4.1)..	100
Figure 4.33: Confusion matrix for tested Medium Gaussian SVM (Experiment 4.2)..	101
Figure 4.34: Confusion matrix for tested Medium Gaussian SVM (Experiment 4.3)..	102
Figure 4.35: Confusion matrix for trained and tested Medium Gaussian SVM (Experiment 4.4)	103
Figure 4.36: Confusion matrix trained stacked ensemble (No.1)	105
Figure 4.37: Confusion matrix of tested stacked ensemble (No.1).....	105
Figure 4.38: Confusion matrix of trained stacked ensemble (No.3).....	106
Figure 4.39: Confusion matrix of tested stacked ensemble (No.3).....	106

Figure 4.40: Confusion matrix trained stacked ensemble (No.4)	106
Figure 4.41: Confusion matrix of tested stacked ensemble (No.4).....	107
Figure 4.42: Confusion matrix trained stacked ensemble (No.6)	107
Figure 4.43: Confusion matrix of tested stacked ensemble (No.6).....	107

LIST OF TABLES

Table 2.1: Examples of commercially available fall detection systems	21
Table 3.1: Available public datasets considered for simulation	33
Table 3.2: ADL categories [83]	34
Table 3.3: Fall Event categories [83]	34
Table 3.4: ADL categories [84]	36
Table 3.5: Fall Event categories [84]	36
Table 3.6: List of algorithms investigated in this study	38
Table 3.7: Simulation experiments done.....	40
Table 4.1: Results from Experiment 1.1 (mean \pm SD; n = 10)	56
Table 4.2: Results from Experiment 1.2 (mean \pm SD; n = 10)	59
Table 4.3 Results from Experiment 1.3 (mean \pm SD; n = 10)	61
Table 4.4: Results from Experiment 1.4 (mean \pm SD; n = 10)	63
Table 4.5: Results from Experiment 2.1 (mean \pm SD; n = 10)	65
Table 4.6: Results from Experiment 2.2 (mean \pm SD; n = 10)	66
Table 4.7: Results from Experiment 2.3 (mean \pm SD; n = 10)	68
Table 4.8: Results from Experiment 2.4 (mean \pm SD; n = 10)	69
Table 4.9: Results from Experiment 3.1 (mean \pm SD; n = 10)	71
Table 4.10: Results from Experiment 3.2 (mean \pm SD; n = 10)	73
Table 4.11: Results from Experiment 3.3 (mean \pm SD; n = 10)	74
Table 4.12: Results from Experiment 3.4 (mean \pm SD; n = 10)	76
Table 4.13: Results from Experiment 4.1 (mean \pm SD; n = 10)	77
Table 4.14: Results from Experiment 4.2 (mean \pm SD; n = 10)	79
Table 4.15: Results from Experiment 4.3 (mean \pm SD; n = 10)	80
Table 4.16: Results from Experiment 4.4 (mean \pm SD; n = 10)	82
Table 4.17: Comparison of classifiers for given simulations.....	83
Table 4.18: Top trained and tested results from Experiment 1.1 (mean \pm SD; n = 10)..	84
Table 4.19: Top trained and tested results from Experiment 1.2 (mean \pm SD; n = 10)..	86
Table 4.20: Top trained and tested results from Experiment 1.3 (mean \pm SD; n = 10)..	87
Table 4.21: Top trained and tested results from Experiment 1.4 (mean \pm SD; n = 10)..	88
Table 4.22: Top trained and tested results from Experiment 2.1 (mean \pm SD; n = 10)..	89
Table 4.23: Top trained and tested results from Experiment 2.2 (mean \pm SD; n = 10)..	91
Table 4.24: Top trained and tested results from Experiment 2.3 (mean \pm SD; n = 10)..	92

Table 4.25: Top trained and tested results from Experiment 2.4 (mean \pm SD; n = 10)..	93
Table 4.26: Top trained and tested results from Experiment 3.1 (mean \pm SD; n = 10)..	94
Table 4.27: Top trained and tested results from Experiment 3.2 (mean \pm SD; n = 10)..	96
Table 4.28: Top trained and tested results from Experiment 3.3 (mean \pm SD; n = 10)..	97
Table 4.29: Top trained and tested results from Experiment 3.4 (mean \pm SD; n = 10)..	98
Table 4.30: Top trained and tested results from Experiment 4.1 (mean \pm SD; n = 10)..	99
Table 4.31: Top trained and tested results from Experiment 4.2 (mean \pm SD; n = 10)	100
Table 4.32: Top trained and tested results from Experiment 4.3 (mean \pm SD; n = 10)	101
Table 4.33: Top trained and tested results from Experiment 4.4 (mean \pm SD; n = 10)	103
Table 4.34: Trained results from SisFall dataset.....	104
Table 5.1: Comparison of the best trained and tested classifiers of Experiment 1 and Experiment 2 for given simulations	108
Table 5.2: Comparison of the best trained and tested classifiers of Experiment 3 and Experiment 4 for given simulations	109

LIST OF ACRONYMS

ADL	Activities of Daily Living/Life
ANN	Artificial Neural Network
DBN	Dynamic Bayesian Network
DOF	Degrees of freedom
DSP	Digital Signal Processing
DT	Decision Tree
FPGA	Feld programmable gate array
EMG	Surface electromyography
GPS	Global Positioning System
HMM	Hidden Markov Model
I2C	Inter-integrated circuit
KNN	k-Nearest Neighbors
ML	Machine learning
MLP	Multilayer perceptron
NB	Naïve Bayes
NN	Neural Network
RFID	Radio Frequency Identification Devices
RPROP	Resilient Backpropagation
STPRtool	Statistical Pattern Recognition Toolbox
SVM	Support Vector Machine

ABSTRACT

Fall accidents from accidental injury are considered one of the significant global public health concerns, and the largest proportion of fatal accidents are experienced by elderly people. Currently, there is a demand for creating an effective machine learning-based fall detection system that is portable at a low cost.

Hence, this project is aimed at undertaking a research study on the various fall detection and navigation systems designed for the elderly and disabled and developing an effective machine learning-based low-cost fall detection system. The methodology included a study on the various fall detection systems as well as general features used in machine learning for fall and ADL (Activities of Daily Living) classifications. The most suitable potential combination of machine learning algorithms that will provide the best accuracy, precision, sensitivity, specificity and lowest training time were developed via simulation models using MATLAB. The SisFall and MobiFall datasets were used for both classification and testing for input data in simulations. SisFall dataset used ADXL345 and MMA8451Q accelerometer model, while the MobiFall dataset used LSM330DLC inertial module from a Samsung Galaxy S3 smartphone.

Up to 24 algorithms were simulated, including Decision Trees, Naïve Bayes Classifiers, Support Vector Machines, KNN and available Ensemble Classifiers. Four experiments were done, with Experiment 1 using the ADXL345 accelerometer with 5-fold cross-validation, Experiment 2 using 10-fold cross-validation, Experiment 3 using the MMA8451Q accelerometer with 10-fold cross-validation and Experiment 4 using the MobiFall Dataset with 5-fold cross-validation. The classifier models were run ten times, and each iteration was saved for further processing. Classifiers showing the most accuracy (up to 99%) in both training and testing phases included Quadratic SVM, Cubic SVM, Medium Gaussian SVM and Fine KNN.

A stacked ensemble method was simulated as well utilizing classifiers such as Medium Gaussian SVM, Cubic SVM and Fine KNN. SisFall and MobiFall Datasets were further used to train and test the classifier system. Initial testing and training had shown an improvement in accuracy (up to 99% in binary classifications) when compared to the system using individual classifiers. Results had shown a remarkable increase in precision, sensitivity and accuracy when classifying data in a binary classification system compared to classifying the data based on the specific categories (the various types of falls and

ADLs). Hence an effective system model was developed and trained to identify the data between both Fall Events and ADLs as well as separately identifying the actual type of Fall/ADLs.