AWARD NUMBER: W81XWH-20-1-0164

TITLE: Determination of Fall Risk for Lower Limb Amputees

PRINCIPAL INVESTIGATOR: Richard R. Neptune

CONTRACTING ORGANIZATION: The University of Texas, Austin, TX

REPORT DATE: January 2024

TYPE OF REPORT: Final

PREPARED FOR: U.S. Army Medical Research and Development Command
Fort Detrick, Maryland 21702-5012

DISTRIBUTION STATEMENT: Approved for Public Release; Distribution Unlimited

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE	**2. REPORT TYPE**	**3. DATES COVERED**
January 2024	Final	01Apr2020-30Sep2023

4. TITLE AND SUBTITLE

Determination of Fall Risk for Lower Limb Amputees

5a. CONTRACT NUMBER W81XWH-20-1-0164

5b. GRANT NUMBER OP190008

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Richard R. Neptune

E-Mail: rneptune@mail.utexas.edu

5d. PROJECT NUMBER 0011455055

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Texas At Austin
110 Inner Campus Dr Mn 13
Austin TX 78712-1139

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Medical Research and Development Command
Fort Detrick, Maryland 21702-5012

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Falling is a common problem for lower limb amputees, which can lead to reduced physical and emotional health. The overall aims of this project were to: 1) establish a baseline fall detection algorithm derived from simulated falls in a laboratory setting, and 2) utilize and refine the initial laboratory-based algorithm to provide detection of fall events during activities of daily living in real-world environments. To achieve these aims we performed two human subject experiments. The first experiment used 30 non-amputee and 5 lower limb amputee individuals to simulate falls in a laboratory setting while wearing IMU sensors. Due to the COVID-19 pandemic, we were delayed in starting the data collection. However, in January 2021 we were given approval to start data collection, which was completed along with the development of the fall detection algorithm (Aim 1). We then performed our second experiment where we recruited 20 lower limb amputees to wear the sensor in the real-world. The data collection was completed and the fall detection algorithm was further refined (Aim 2). This research resulted in three conference presentations and two journal manuscripts.

15. SUBJECT TERMS

Biomechanics, amputation, balance, fall detection, sensors, algorithms

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** USAMRDC
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)*
Unclassified	Unclassified	Unclassified	Unclassified	15	

TABLE OF CONTENTS

1. Introduction

Falling is a common problem for lower limb amputees, which can lead to reduced physical and emotional health. The overall aims of this project were to: 1) establish a baseline fall detection algorithm derived from simulated falls in a laboratory setting, and 2) utilize and refine the initial laboratory-based algorithm to provide detection of fall events during activities of daily living in real-world environments. To achieve these aims we performed two human subject experiments. The first experiment used 30 non-amputee and 5 lower limb amputee individuals to simulate falls in a laboratory setting while wearing inertial measurement unit (IMU) sensors. The IMU sensors recorded the motion of the body while falling and performing activities of daily living that allowed us to use machine learning techniques to create an algorithm to detect a fall relative to normal daily activities. The second experiment recruited 20 lower limb amputees to wear IMU sensors in the real-world for two 5-hour periods. Participants were also asked to report any fall events that were not detected so that the algorithm could be further improved. The outcome of this work is a fall detection algorithm that will allow clinicians to better understand the number of falls that occur for lower limb amputees. This initial study provides the foundation for a future large scale clinical trial where large numbers of amputees can be supplied with IMU sensors in order to better quantify falling in the larger amputee community and other communities that are at high risk for falling.

2. Keywords

Biomechanics, amputation, balance, fall detection, sensors, algorithms

3. Accomplishments

What were the major goals of the project?

Specific Aim 1: Establish a baseline fall detection algorithm derived from simulated falls in a laboratory setting.	**Timeline (months)**	**Status**
Major Task 1.1: Human subject experiment (n=35)	1-7	
Milestone 1.1.1: Obtain approval from the governing Institutional Review Boards.	2	Complete
Milestone 1.1.2: Complete enrollment of all participants and collect experimental data.	5	Complete
Major Task 1.2: Analyze Human Subject Data	5-8	
Milestone 1.2.1: Perform machine learning analysis of falling data from healthy subjects and amputees to determine the initial fall detection algorithm.	1	Complete
Milestone 1.2.2: Perform hypothesis tests to evaluate the effectiveness of the falling algorithm.	1	Complete
Milestone 1.2.4: Complete writing of manuscript and conference abstract describing initial algorithm development and results.	2	Two Abstracts Completed, Manuscript in review
Specific Aim 2: Utilize and refine the initial laboratory-based algorithm to provide detection of fall events during activities of daily living in pragmatic, real-world environments.	**Timeline (months)**	
Major Task 2.1: Human subject experiment (n=20)	6-21	

Milestone 2.1.1: Obtain approval from the governing Institutional Review Boards.	2	Complete
Milestone 2.1.2: Complete enrollment of all participants and collect experimental data.	15	Complete
Major Task 2.2: Analyze Human Subject Data	21-24	
Milestone 2.2.1: Perform machine learning analysis on complete dataset to determine final algorithm with all data.	2	Complete
Milestone 2.2.2: Perform hypothesis tests to evaluate the effectiveness of the falling algorithm in the real-world.	3	Complete
Milestone 2.2.3: Complete writing of manuscript and conference abstracts describing the algorithm development, validation and results.	3	One Abstract Complete, Manuscript in preparation

What was accomplished under these goals?

We successfully completed Aims 1 and 2. In Aim 1, IMU sensors were placed on both shanks of healthy young adults (n=30) and individuals with a lower-limb amputation (n=5) while they performed four types of simulated falls and twelve activities of daily living. Sixty-four datasets were created with different setup parameters. Three types of machine learning algorithms (K-Nearest Neighbor, Support Vector Machine, and Random Forest) were used for activity classification. Total accuracy, fall detection accuracy and number of false positives were determined. Fall detection accuracy was not the same across fall types, and the type of algorithm and setup parameters played a key role in all accuracy outcome measures. Fall detection accuracy reached up to 94% depending on the machine learning algorithm used. In Aim 2, data were collected from 20 individuals with a lower limb amputation in real world environments during two, 5-hour periods, which was used to further refine the fall detection algorithms. Unfortunately, no real world falls occurred to test the algorithm performance. In addition, we also developed a novel automated neural network framework that uses multi-objective optimization to train both machine learning and deep learning models for fall detection. This framework facilitates the deployment of these models on hardware with limited resources, which will be critical for future development of IMU-based fall detection devices.

What opportunities for training and professional development has the project provided?

This project has provided professional development opportunities for graduate students Lindsey Lewallen and Mojtaba Mohasel through the development of research skills, training in technical writing and presenting their work at scientific conferences.

How were the results disseminated to communities of interest?

The results from Aim 1 were accepted and presented at the annual meeting of the *American Society of Biomechanics* in August, 2021. A research poster describing this work was also presented at the UT Austin Department of Mechanical Engineering Graduate Student Research Poster Session. A manuscript describing this work was reviewed and currently being revised based on the reviewer comments. A second conference abstract related to Aim 1was accepted and presented at the *North American Congress on Biomechanics* in August, 2022. Research related to Aim 2 was accepted and presented at the annual meeting of the *America Society of Biomechanics* in August 2023. A manuscript describing the Aim 2 work in currently in preparation. The three conference abstracts can be found in the Appendix.

What do you plan to do during the next reporting period to accomplish the goals?

Nothing to report

4. Impact

What was the impact on the development of the principal discipline(s) of the project?

This work represents an important first step towards improved understanding of the incidence of falls. More in-depth and immediate information regarding a patient's falling history will help clinicians identify individuals who may need increased rehabilitation intervention to help prevent future falls. In addition, information collected in a future large-scale clinical trial using this algorithm could help identify individuals at risk of falling based on their clinical profile (age, sex, height, weight, amputation etiology, comorbidities, activity level, K-level and prosthetic hardware), allowing clinicians to be more proactive with an individuals' rehabilitation and help prevent future falls. In addition, an understanding of fall frequency could significantly improve rehabilitation outcomes and accelerate the service member's return to active duty. In the general DoD and VA community, reducing and/or preventing falls can assist in maximizing patient outcomes and quality of life while reducing the emotional and monetary costs associated with falls.

What was the impact on other disciplines?

The primary outcome from this research (a validated algorithm to detect falls in lower limb amputees) primarily constitutes a Knowledge Product that directly impacts rehabilitation science.

What was the impact on technology transfer?

Nothing to report

What was the impact on society beyond science and technology?

This research provides a platform for quantitative analysis of falling frequency in the lower limb amputee community. While it is accepted that lower limb amputees are at higher risk for falls than the general population, quantifying falls can be used to better understand the level of risk that falls present for amputees. This information can help clinicians reduce amputee falling through improved clinical interventions. A clinician's ability to identify which patients experience a higher number of falls or are at greater risk for falling will allow them to better personalize a rehabilitation plan and improve prosthesis prescription to help reduce falling. The future goal of this project is to create an algorithm that will be able to detect fall incidence in the amputee population in real-time. This will allow medical personnel to be more quickly alerted to potentially injurious falls and reduce delays for emergency intervention. In addition, providing a patient's fall history to clinicians in real-time will allow them to determine which individuals may need increased attention and rehabilitation before their next planned visit, helping prevent future falls. This project is an important first step in realizing this goal and provides the foundation for analyzing falls in other patient populations.

5. Changes/Problems

As noted in previous reports, our initial Aim 1 data collection was delayed due to the COVID-19 pandemic. In addition, in Aim 2 the original plan was to implement the best performing algorithm developed in Aim 1 in IMU-based activity monitors such that the system would send an alert to the investigative team if a fall was detected. This information would be used to refine the algorithm in the case of a false negative or false positive fall detection. The activity monitor data was to be uploaded to the cloud server through a low band frequency LTE signal that AT&T is currently broadcasting. The proposed work was based on the agreement by AT&T to let us use this signal. However, due to the COVID-19 pandemic and fiscal pressures, AT&T subsequently discontinued the availability of the signal and cloud service so we are not able to complete Aim 2 as proposed.

Revised Statement of Work

In our approved revised statement of work, we modified Aim 2. We proposed to fit up to 20 individuals with a lower limb amputation with an IMU sensor and data logger (3-Space™ Data Logger - Yost Labs) over two separate 5-hour periods. Five hours was the limit of onboard data storage for the IMU sensors. Individuals were asked to leave the clinic and perform their normal activities of daily living. In the event of a fall, they were asked to record the approximate time of the fall and the activity they were doing. They were also asked to complete a fall questionnaire at the end of each 5-hour period. The raw IMU data was downloaded upon the individuals' return following each 5-hour period. The fall detection algorithm developed in Aim 1 was then applied in a post-hoc analysis to determine the sensitivity/specificity of the algorithm in real world environments. The updated algorithm was then re-applied to the original laboratory based data to determine how the algorithm performed on the original data after being modified for the real-world environments. We successfully completed the revised statement of work and a corresponding manuscript is currently in preparation.

6. Products

Publications, conference papers, and presentations

Lewallen, L.K., Pew, C.A., Wurdeman, S.R., and Neptune, R.R. (2021). Detection of different fall types in healthy young adults. *45th Annual Meeting of the American Society of Biomechanics*, August 10-13, Atlanta, GA.

Mohasel, M., Lewellen, L.K., Pew, C., Neptune, R.R. (2022). A machine learning scheme to identify falling for lower limb amputees. *North American Congress on Biomechanics*, August 21-25, Ottawa, ON, Canada.

Lewallen, L.K., Pew, C.A., Wurdeman, S.R., and Neptune, R.R. (2022). Detection of different fall types in healthy young adults. *Department of Mechanical Engineering Graduate Student Research Poster Session*, March 4, Austin, TX.

Molina, L.K., Pew, C.A., Wurdeman, S.R., and Neptune, R.R. (2023). The influence of machine learning configurations on the detection of fall types in individuals with lower-limb amputations. *IEEE Sensors* (in revision).

Mohasel, M., Molina, L.K., Wurdeman, S.R., Neptune, R.R., and Pew, C.A. (2023). Development of an automated framework for a TinyML-based fall detection system. 47th Annual Meeting of the American Society of Biomechanics, August 8-11, Knoxville, TN.

Mohasel, M., Molina, L.K., Wurdeman, S.R., Neptune, R.R. and Pew, C.A. (2023). A machine learning scheme to identify falling for lower limb amputees, *IEEE Sensors* (in preparation).

Website(s) or other Internet site(s)

Nothing to report

Technologies or techniques

The algorithms generated by this work along with the human subject data used to generate the algorithm is available to the research community upon request.

Inventions, patent applications, and/or licenses

Nothing to report

Other Products

Nothing to report

7. Participants & Other Collaborating Organizations

What individuals have worked on the project?

Name:	Richard R. Neptune
Project Role:	PI
Researcher Identifier:	NIH eRA Commons ID: rneptune
Nearest person month worked:	1
Contribution to Project:	Dr. Neptune helped put together the IRB application for approval from both UT Austin and HRPO. He oversaw the entire project and supervised the graduate student working on the project.
Name:	Lindsey K. Lewallen
Project Role:	Graduate Student
Researcher Identifier:	N/A
Nearest person month worked:	6
Contribution to Project:	Ms. Lewallen helped put together the two IRB applications, collected and processed the Aim 1 data, and developed the machine learning algorithms.
Name:	Corey A. Pew
Project Role:	Collaborator
Researcher Identifier:	N/A
Nearest person month worked:	1
Contribution to Project:	Dr. Pew helped develop and refine the machine learning algorithms.
Name:	Shane R. Wurdeman
Project Role:	Collaborator
Researcher Identifier:	NIH eRA Commons ID: wurdemans
Nearest person month worked:	1
Contribution to Project:	Dr. Wurdeman served as the project’s clinical partner and collected the real-world data in Aim 2.

Has there been a change in the active other support of the PD/PI(s) or senior/key personnel since the last reporting period?

Nothing to report

What other organizations were involved as partners?

- **Organization Name:** Hanger Clinic
- **Location of Organization:** Austin, TX
  - **Partner's contribution to the project**: Collaboration, helped with subject recruitment and data collection.

- **Organization Name:** Montana State University,
- **Location of Organization:** Bozeman, MT
  - **Partner's contribution to the project**: Collaboration, helped with algorithm development.

8. Special Reporting Requirements

Collaborative Awards: Not applicable

Quad Charts: Uploaded separately

9. Appendices

- Lewallen, L.K., Pew, C.A., Wurdeman, S.R., and Neptune, R.R. (2021). Detection of different fall types in healthy young adults. *45th Annual Meeting of the American Society of Biomechanics*, August 10-13, Atlanta, GA.

- Mohasel, M., Lewellen, L.K., Pew, C., Neptune, R.R. (2022). A machine learning scheme to identify falling for lower limb amputees. *North American Congress on Biomechanics*, August 21-25, Ottawa, ON, Canada.

- Mohasel, M., Molina, L.K., Wurdeman, S.R., Neptune, R.R., and Pew, C.A. (2023). Development of an automated framework for a TinyML-based fall detection system. 47th Annual Meeting of the American Society of Biomechanics, August 8-11, Knoxville, TN.

- Lewallen, L.K., Pew, C.A., Wurdeman, S.R., and Neptune, R.R. (2022). Detection of different fall types in healthy young adults. *Department of Mechanical Engineering Graduate Student Research Poster Session*, March 4, Austin, TX.

DETECTION OF DIFFERENT FALL TYPES IN HEALTHY YOUNG ADULTS

Lindsey K. Lewallen[1], Corey A. Pew[2], Shane R. Wurdeman[3], and Richard R. Neptune[1]
[1]Walker Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX
[2]Department of Mechanical and Industrial Engineering, Montana State University, Bozeman, MT
[3]Department of Clinical and Scientific Affairs, Hanger Clinic, Austin, TX
email: lindsey.lewallen@utexas.edu

Introduction

Individuals with a lower-limb amputation are at an increased risk of falling compared to young healthy adults. Approximately 50% of individuals with unilateral amputation report at least one fall annually.[1,2] Falls are dangerous, occasionally leading to injury, hospitalization or death.[3] Fortunately, individuals who obtain aid within 1 hour of a fall have a 50% increased survival rate compared to individuals who obtain aid after 72 hours.[4] Thus, devices that have the ability to detect falls and alert proper personnel could serve to help lower the consequences of falling for individuals with a lower-limb amputation.

A number of studies have developed body worn sensors that can detect fall events. These devices primarily use inertial measurement units (IMUs) to record signals from 3-axis accelerometers, gyroscopes, and/or magnetometers. Individuals with a lower-limb amputation utilize a prosthesis that allows for fall detection sensors to be conveniently integrated within the prosthesis (e.g., directly attached to the pylon). However, it is not clear if such sensors are able to detect a wide range of fall types. Therefore, the purpose of this study was to investigate the accuracy of detecting different fall types with an IMU placed on an individual's shank in preparation for application and validation on individuals with a lower limb amputation.

Methods

IMU sensors (XSens, Enschede, Netherlands) were placed on both shanks of 15 healthy young adults in positions analogous to the pylon of a prosthesis distal to the knee. Tri-axis accelerometer and gyroscope data were recorded from these devices at 100-Hz while subjects completed an overground course with simulated falls and near-falls. The course was designed to simulate activities of daily living (ADL: walking/running in a straight line at a self-selected pace, navigating turns, sitting and rising from a chair, laying down and getting up from a bed, picking up an object on the floor, and ascending/descending stairs/slopes). Subjects performed 4 types of simulated falls: forward/backward trips (i.e., subjects walked forward/backward until they impacted a fall pad and fell) and left/right lateral falls (i.e., subjects stood with their left/right side adjacent to the fall pad while a lab technician pushed them until they lost balance and fell onto the fall pad). For the simulated near falls, subjects walked until their left/right foot struck the fall pad and then recovered from the stumble.

Raw data were analysed using the MATLAB Classification Learner Toolbox. First, data were split into two categories: ADL or Fall. Data were divided into 0.5 second windows with a 0.25 second overlap. During these 0.5 second windows, a total of 40 features were computed (Table 1). Data were randomly split into training (80%) and model verification (20%) sets for each subject and each category. Three different classification algorithms were used for activity classification and validated with 5-fold cross validation: support vector machine with a cubic kernel (SVM), K nearest neighbor with weighted dimensions (kNN), and a bagged decision tree ensemble (Tree).[5]

Table 1: Features extracted for each 0.5 s window for each accelerometer (accel) and gyroscope (gyro).

Vector resultant (r_{accel}, r_{gyro})	Median, Mean, Standard Deviation, Skewness, Kurtosis, IQR, Minimum, Maximum
Each axis (x_{accel}, y_{accel}, z_{accel}, x_{gyro}, y_{gyro}, z_{gyro})	Mean, Max, Min, IQR

To determine algorithm accuracy, a simple control scheme was created. First, models were implemented on the verification data set. A fall was identified if at least two adjacent windows contained a label associated with a fall. If this occurred within the duration of the fall (~1s), a correct fall classification was made. Falls were labelled by type: forward/backward trips and lateral falls with the sensor placed on the inside/outside leg. Finally, fall detection accuracy was calculated, defined as the number of correct classifications divided by total number of falls (Table 2).

Results and Discussion

Forward falls had the lowest detection accuracy for each algorithm. When falling forward, participants can more easily protect their body with their hands and knees, acting to reduce the acceleration on impact. On average, inside falls had the highest detection accuracy. The inside shank is often the first part of the body that impacts the ground during lateral falls, possibly contributing to the higher accuracy. This is in contrast to previous work that noted highest classification accuracy with backward falls when an IMU sensor is placed on the waist of each participant.[6]

Significance

This study highlighted that fall detection accuracy is not the same across fall types and classification algorithms. Future work should seek to improve detection of forward falls (e.g., placing sensors in different locations, implementing different classification algorithms such as threshold algorithms,[7] and exploring different features) and validate these results on individuals with a lower limb amputation.

Acknowledgments

This work was supported by CDMRP W81XWH2010164.

References

[1]Miller WC, et al. *Arch Phys Med Rehabil.* 2001. [2]Kulkarni J, et al. *Physiotherapy.* 1996. [3]Rubenstein LZ, et al. *Age Ageing.* 2006. [4]Gurley RJ, et al. *N Engl J Med.* 1996. [5]Pew C, et al. *IEEE.* 2018. [6]Hwang SY, et al. *Int Conf on CS and Tech.* 2012.

Table 2: Accuracy for each type of fall and algorithm

	Type of Fall				
	Forward	**Backward**	**Outside**	**Inside**	**All Falls**
SVM	76.7%	93.3%	100%	100.0%	92.5%
kNN	73.3%	90.0%	90.0%	100.0%	88.3%
Tree	76.7%	86.7%	93.3%	93.3%	87.5%

Detection of Different Fall Types in Healthy Young Adults

Lindsey K. Lewallen[1], Corey A. Pew[2], Shane R. Wurdeman[3], and Richard R. Neptune[1]

[1] Walker Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX
[2] Department of Mechanical and Industrial Engineering, Montana State University, Bozeman, MT
[3] Department of Clinical and Scientific Affairs, Hanger Clinic, Austin, TX

Introduction

- Individuals with a lower-limb amputation are at an increased risk of falling compared to healthy adults.[1,2]
- Body worn sensors have the potential to detect fall events and alert proper personnel.
- Body worn sensors primarily use inertial measurement units (IMUs) to record signals from 3-axis accelerometers, gyroscopes, and/or magnetometers.
- Individuals with a lower-limb amputation use a prosthesis that allow for fall detection sensors to be conveniently attached to the pylon (Fig. 1).
- However, it is not clear if such sensors can detect a wide range of fall types.

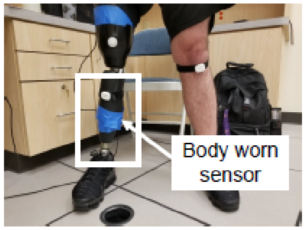


Figure 1: Location of body worn sensor on an individual with a lower-limb amputation.

Purpose

Investigate the accuracy of machine learning algorithms in detecting different fall types with an IMU placed on an individual's shank.

Methods

Experimental Data:

- Tri-axis accelerometer and gyroscope data were recorded from IMU sensors placed on both shanks of 15 healthy young adults.
- Subjects completed an overground course designed to simulate activities of daily living (ADL) and performed 3 types of simulated falls/near falls:
  - Trips: subjects walked forward (backward) until they impacted a fall pad and fell.
  - Lateral falls: subjects stood with their left (right) side adjacent to the fall pad and a lab technician pushed them onto the pad (Fig. 2).

Methods cont.

Figure 2: Lateral fall.

- Near falls: subjects walked until their left (right) foot struck the fall pad and recovered from the stumble

Machine Learning Algorithms:

- Data were split into two categories (ADL or Fall) and divided into 0.5s windows with a 0.25s overlap.
- 40 features were computed for each window (Table 1).

Table 1: Features extracted for each 0.5 s window for each accelerometer (accel) and gyroscope (gyro).

Vector resultant (r_{accel}, r_{gyro})	Median, Mean, Standard Deviation, Skewness, Kurtosis, IQR, Min, Max
Each axis (x_{accel}, y_{accel}, z_{accel}, x_{gyro}, y_{gyro}, z_{gyro})	Mean, Max, Min, IQR

- Data were randomly split into training (80%) and model verification (20%) sets for each subject and category.
- Support vector machine with a cubic kernel (SVM), K nearest neighbor with weighted dimensions (kNN), and bagged decision tree ensemble (Tree)[3] were used for classification and validated with 5-fold cross validation.

Accuracy Calculation:

- A fall was identified when 2+ adjacent windows contained a fall label and was correctly classified if this occurred within the duration of the fall (~1s).
- Falls were separated by type: forward and backward trips and lateral falls with the sensor placed on the inside or outside leg.
- Fall detection accuracy was defined as the number of correct classifications divided by total number of actual falls (Table 2).

Results

Table 2: Accuracy for each type of fall and algorithm.

	Type of Fall				
	Forward	Backward	Outside	Inside	All Falls
SVM	81.7%	93.3%	96.7%	96.7%	92.1%
kNN	73.3%	90.0%	93.3%	100.0%	87.2%
Tree	63.3%	85.0%	86.7%	96.7%	85.4%

- Forward falls had the lowest detection accuracy for each algorithm while inside falls had the highest detection accuracy (Table 2).

Discussion and Significance

- Fall detection accuracy is not the same across fall types and classification algorithms.
- When falling forward, participants can more easily protect their body with their hands and knees, acting to reduce the acceleration on impact.
- The inside shank is often the first part of the body that impacts the ground during lateral falls.
- Previous work noted highest classification accuracy with backward falls when an IMU sensor is placed on the waist of each participant.[4]
- Future work should seek to improve detection of forward falls (e.g., different classification algorithms such as threshold algorithms[4])
- Future work should validate these results on individuals with a lower limb amputation.

References & Acknowledgements

[1]Miller WC, et al. *Arch Phys Med Rehabil*. 2001.
[2]Kulkarni J, et al. *Physiotherapy*. 1996.
[3]Pew C, et al. *IEEE*. 2018.
[4]Hwang SY, et al. *Int Conf on CS and Tech*. 2012.

This work was supported in part by CDMRP W81XWH2010164.

A Machine Learning Scheme to Identify Falling for Lower Limb Amputees

Mojtaba Mohasel[1], Lindsey Lewallen[2], Shane R. Wurdeman[3], Richard R. Neptune[2], Corey Pew[1]
[1]Department of Mechanical and Industrial Engineering, Montana State University, Bozeman, MT
[2]Walker Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX
[3]Clinical and Scientific Affairs, Hanger Clinic, Austin, TX
Email:*Corey.Pew@montana.edu

Introduction

Falls present a major health risk for individuals with lower limb amputation [1]; however, real-world falls are difficult to objectively measure. One method to detect falls is to use inertial measurement units (IMUs) and machine learning to classify falling events relative to normal activities of daily living [2]. However, most existing algorithms process the data offline and there is a delay in identifying a fall. The purpose of this study was to develop machine learning methods that can detect fall incidence in the amputee population in real-time.

Methods

Fall detection algorithms were developed using data from 30 intact and 5 lower limb amputee participants. An IMU sensor attached in the middle of the shank measured acceleration and angular velocity in the x, y, and z directions. Participants navigated a course in the laboratory that consisted of various activities of daily living (ADL) and controlled falling. The collected data was used as input to a customized machine learning pipeline to process the data and optimize settings for a classification algorithm [2]. Data was divided into Training Data (algorithm construction, 30 intact participants), Validation Data (feedback for optimization, 2 amputee participants), and Test Data (validation of classifier ability, 3 amputee participants). Falling data was outnumbered compared to ADL and so the pipeline utilized the Synthetic Minority Oversampling Technique (SMOTE) [3] to create a balanced dataset. The trained classifier is specifically developed for deployment on a platform with limited processing power and memory (ESP32 processor with 512 KB of onboard memory) which guided the selection of two possible classifiers. The first method tested was a Multilayer Perceptron neural network (MLP). An MLP is not time dependent, classifications utilize raw data directly (6 raw IMU channels) without a sliding time window, and can be modified for low computing power by reducing the number of layers and neurons [4]. The second method utilized a Support Vector Machine (SVM) with a Radial Basis Function kernel which showed promising results in distinguishing near-falls from ADL [5]. To reduce the classifier power requirement and size, input features were restricted to resultant acceleration and angular velocity ($x^2+y^2+z^2= r$ for each). The SVM utilized a sliding time window, the length and overlap of which was optimized by the pipeline to obtain the highest detection accuracy. The number of support vectors is influenced by the number of samples in the training data (a larger dataset requires more memory), so after using SMOTE to balance the dataset the pipeline randomly under sampled the Training Data to reduce the size of the classifier. Classifiers were then compared by their ability to accurately detect both falling events and ADL events in the Test Data.

Results and Discussion

On average, the MLP had better detection ability and a smaller memory requirement (Table 1).

Table1: Comparison of detection rates between MLP and SVM algorithms. Training Data was used to create the algorithm, Validation Data (Val Data) was used for optimization, and Test Data were used to assess the performance of the final classifiers. Values for Fall indicate the rate for identifying falls while ADL indicates the rate for identifying activities of daily living. Run Time Size indicates the size of the compiled C-Code classifier.

Data Type	MLP		SVM	
	Fall	ADL	Fall	ADL
Training	99%	94%	99%	93%
Val Data	88%	95%	84%	84%
Test Data	87%	93%	83%	82%
Run Time Size	77 KB		344 KB	

An important note, the MLP was trained using all Training Data while the SVM was trained with a reduced portion of the Training Data which will likely adversely affect real-world performance. Furthermore, the MLP utilized all 6 channels independently whereas the SVM was reduced to two resultant features. This suggests that training with the full, unmodified feature set can increase classifier performance. In future work, we will incorporate these classifiers into hardware that will be used with lower limb amputees to determine the validity of the laboratory-based classifier and objectively quantify falling in the real world. Final algorithm selection will be determined by memory usage, processor performance, and prediction speed during real-time use on our ESP32 hardware.

Significance

In this study, we developed two algorithms capable of detecting falls in real time on limited processor and memory hardware. Previous research with a similar single sensor placement has obtained 85% detection for fall and ADL [6], however, they were not limited by processor or memory. These fall detection algorithms, when implemented on individuals in the real world, will have the ability to provide clinicians with accurate and objective information about patient falling. This will allow for a better understanding of which patients may need interventions to mitigate future falling such as modifications to their prosthetic components or prescription of specific exercise protocols.

Acknowledgements

Funding provided by CDMRP Grant W81XWH-20-1-0164. The authors thank Bryce Billings from Hanger Clinic for feedback on this work.

References

1) Miller et al., Arch. Phys. Med. Rehab. (82)20011; **2)** Mohasel & Pew, 45th Meeting ASB, 2021; **3)** Chawla et al., Journal of AI Research, (16)2002; **4)** Goodfellow et al, Deep Learning, 2015; **5)** Aziz et al, IEEE, 2012; **6)** Ramón et al, Sensors, (18)2018

A Machine Learning Scheme to Identify Falling for Lower Limb Amputees

Mojtaba Mohasel[1], Lindsey Lewallen[2], Shane R. Wurdeman[3], Richard R. Neptune[2], *Corey A. Pew[1]
[1]Department of Mechanical and Industrial Engineering, Montana State University, Bozeman, MT [2]Walker Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX [3]Clinical and Scientific Affairs, Hanger Clinic, Austin, TX ,
Email:*Corey.Pew@montana.edu

Background

Falling

- Falls present a major health risk for individuals with lower limb amputation including fractures, traumatic brain injuries, lacerations, sprains, hematomas, and death [1]
- Amputee falling has not been fully quantified, and an improved understanding of how often and why amputees fall is needed
- Falls are often underreported [2,3], quantitative accounts needed
- An improved understanding of amputee falling will provide clinicians with objective measurements of fall risk

Fall Detection

- Inertial measurement units (IMUs) (multi-axis accelerometers, gyroscopes, and/or magnetometers) can be used to detect motion
- Machine learning (ML) techniques can then interpret IMU data by classifying activity of daily living (ADL) and fall events

Problem

Fall detection using IMUs and ML classifiers often utilize laboratory-based data for training and testing, however, the functionality of lab-based classifiers has not been fully evaluated in the real world [4]

Purpose

Develop fall detection classifiers from laboratory based falling for use in real-world hardware

Real-world hardware consists of limited processing and onboard memory capabilities (Fig 1.)

Fig 1. ESP32 processor with 512 KB of onboard memory

Methods (Experimental Design)

- Data were collected from 30 intact and 5 lower limb amputee participants.
- IMU sensors were attached mid shank on both limbs (White boxes, Fig 2)
- Classifiers were developed for deployment on a platform with limited processing power and memory (Fig 1).
- ESP32 hardware was limited to 240 MHz processing and 512 KB of onboard memory

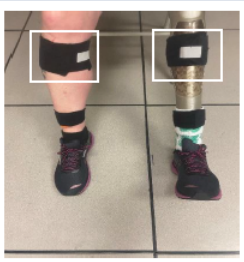


Fig 2. IMU sensor on the shank of amputee

Methods (Data collection)

Participants performed ADL movements (walking, turning, sitting, laying, ramps and stairs) in addition to four simulated falls (forward, backward, right and left lateral)

Fig 3. Backward fall

Fig 4. Lateral fall

Method (Model Development)

Model development utilized an automated Pipeline to create classifiers for movement identification [5]

1. Balancing data

- 99% of data was labeled ADL
- Random under sampling used to balance

2a. Neural Networks

- Multi-layer Perceptron (MLP) neural networks with N feasible architectures ideal for low-memory applications [6]

2b. Support Vector Machine (SVM)

- Simpler SVM models have previously shown success with fall detection [7]

3. Model Optimization

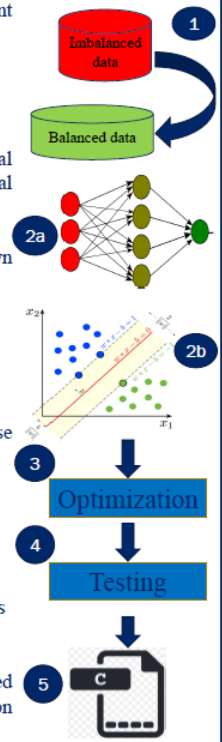
- Intact Data (single sensor, dominant limb)
- Grid search used to tune hyper parameters
  - SVM: Kernel, C, and Gamma
  - MLP: Layers, Neurons, and Reg Term
- 5-fold cross validation indicated better/worse model performance (Val Data)

4. Testing Performance

- Amputee data (single sensor, prosthesis)
- 2 participants used to tune window size
  - Between 2-15 samples
- 3 participants used to test outcome
- 2/3 test-tune ratio rotated between 5 amputees
  - Average of all outcomes reported

5. Generating C code

- C-code of the best models were generated to evaluate memory size and implement on real-world hardware



Results

Table1: Comparison between MLP and SVM Classifiers

Data Type	MLP		SVM	
	Fall	ADL	Fall	ADL
Training	99%	94%	99%	93%
Val Data	90%	93%	87%	85%
Test Data	87%	93%	83%	82%
Memory Size	77 KB		344 KB	

- % = True detection (ADL or Fall)/(True detection + false detection)
- Memory Size = size of the compiled C-Code classifier.
- MLP showed better detection ability on test data and a smaller memory requirement

Discussion

- Successful outcomes were achieved with a single sensor on the shank and limited processing ability
- Previous work with single sensor placement obtained 85% detection for Fall and ADL combined [8], without processor or memory limits
- In SVM the number of support vectors and the number of features determined the size of the model while in MLP the number of features and the number of neurons as well as biases in different layers determined the size of the model

Conclusion

- Two algorithms were developed capable of detecting falls in real time on limited processor and memory hardware
- Objective fall detection data will provide clinicians with important information on patient falling
- This data can be used to mitigate future falling through rehabilitation programs and modifications of prosthesis hardware

Acknowledgements

- Funding provided by OPORP Grant W81XWH-20-1-0164.

References

1) Miller et al, " Archives of physical medicine and rehabilitation 82(8), 2001; 2) Ganz et al, Journal of the American Geriatrics Society, 53(12), 2003; 3) Cummings et al, Journal of the American Geriatrics Society 36 (7), 1988; 4) Granacher et al, 47(6) 2014 5) Mohasel & Pew, 45th Meeting ASB, 2021; 6) Goodfellow et al., Deep Learning, 2016; 7) Aziz et al, IEEE, 2012; 8) Ramón et al, Sensors, (18), 2018

Development of an Automated Framework for a TinyML-Based Fall Detection System

Mojtaba Mohasel[1], Lindsey K. Molina[2,3], Shane R. Wurdeman[4], Richard R. Neptune[2], Corey A. Pew[1]
[1]Department of Mechanical and Industrial Engineering, Montana State University, Bozeman, MT
[2]Walker Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX
[3]Exponent, Inc., Phoenix, AZ
[4]Clinical and Scientific Affairs, Hanger Clinic, Austin, TX
Email:*Corey.Pew@montana.edu

Introduction

Falls pose a significant risk of injury and even mortality for individuals with lower limb amputations [1]. Although fall detection devices can objectively track fall incidence, they often have limited memory and low power. Therefore, many previous studies have relied on simple machine learning (ML) algorithms, which can suffer from high false alarms and low detection rates [2]. In contrast, deep learning (DL) models have the potential to reduce false alarms by automatically learning features from input data using neural networks. However, designing a TinyML [3] model architecture (i.e., to be run on low-power, small footprint devices) that achieves a high detection rate relies on multiple interdependent variables, making manual configuration challenging. This study aims to automate both ML and DL workflows and optimize their performance, thereby enabling the development of an efficient TinyML system.

Methods

Data were collected from 35 individuals, 30 intact controls (model training) and 5 lower limb amputees (model testing). Two inertial measurement unit sensors [4] placed on the anterior of each shank measured acceleration and angular velocity in the x, y, and z directions. Participants navigated a laboratory course that involved a range of activities of daily living (ADL) and controlled falling movements.

Data was highly imbalanced, with 98.3% ADL versus 1.7% falls, requiring appropriate methods and metrics (F-score) for training and model comparison. RUSBoost [5] and Easy Ensemble [6] are designed for imbalanced data and represent ML models. For DL, a one-dimensional Convolutional Neural Network (CNN) has shown high accuracy on time series data [2]. CNNs extract features from input data with convolutional layers, allowing parallel processing and faster inference time compared to similar deep models. The final model was designed to be implemented on an ESP32 processor with onboard memory of 512 KB. Automation utilized a weighted sum approach [7] with F-beta score and the number of inference operations as objectives, and memory capacity as a constraint.

ML classifier data was segmented into windows of 15 consecutive samples based on hardware restrictions. Optimal performance of models considered several hyperparameters including the number of estimators, maximum depth of trees, minimum sample leaves, minimum number of samples required to split a node, the cost-complexity parameter (ccp_alpha), sampling strategy, and window size. A Bayesian optimization method [8] and 10-fold cross-validation techniques were employed to tune the hyperparameters and determine the optimal combination that yielded the highest F-beta score while minimizing the number of inference operations.

CNN data was segmented into fixed windows of 100 consecutive samples, representing the duration of a fall or ADL. A weighting method by percent of sample count was utilized to handle the class imbalance [9]. Hyperparameters including the number of convolutional, pooling, and dropout layers, as well as their order, filter size, number of fully connected layers, and number of neurons were fine-tuned using Bayesian optimization. The developed neural architecture search approach simultaneously scales all dimensions of the network (width, depth, and resolution) and constructs architectures that use memory below 512 KB.

Results and Discussion

Table1: Comparison of developed models. Best performance bolded.

Model	RUSboost		EasyEnsemble		CNN	
Class	Fall	ADL	Fall	ADL	Fall	ADL
Recall	91%	79%	81%	93%	**97%**	**98%**
Precision	6%	100%	15%	100%	**43%**	**100 %**
F-score	87%		95%		**98%**	
Run-time	**18 KB**		27KB		228 KB	

The CNN model outperformed in all metrics except run time size (Table 1). The RUSboost model ranked second in fall detection with a 91% recall, but with a high false alarms rate (6% precision). The EasyEnsemble model lowered the false alarm rate but at the expense of misclassifying fall incidence (81% recall). The success of the CNN model can be attributed to three factors: 1) CNN employs time domain features to distinguish between falls and ADLs, whereas the raw data was directly used in the other models as generating features for traditional ML models was not feasible in real time due to the hardware constraint of our ESP32, 2) availability of 7 million samples for training favors deep models more, and 3) the weighting method utilized for CNN is effective for high imbalance ratios.

Significance

This study presents a novel automated framework that uses multi-objective optimization to train ML and DL models. It facilitates the deployment of these models on hardware with limited resources, which is ideal for settings where resources are constrained. The framework also addresses the persistent challenge of class imbalance in fall detection studies.

Prior research [2] developed a CNN architecture for TinyML, achieving a 96% recall rate without class imbalance, indicating our present work meets or exceeds previous methods. Building upon this work, the proposed automated framework can export TinyML with high detection rates without requiring tedious manual model tuning. Our CNN model can provide clinicians with accurate and objective information about patient falls, enabling them to develop appropriate interventions and prosthetic prescriptions to improve patient care and safety.

Acknowledgements

Funding provided by OPORP Grant W81XWH-20-1-0164. The authors thank Bryce Billings from Hanger Clinic for feedback on this work.

References

1) Miller et al., Arch. Phys. Med. Rehab. 82(2001). 2) Salah et al, Int. J. Elect. Comp. Eng. 12, no. 4(2022). 3) Warden p et al, O'Reilly media 2016 4) Choi, A et al, IEEE TNSRE 30 (2022) 5) Seiffert et al, IEEE T-Systems, 1(2009) 6) Liu et al, IEEE IBSIC, (2009) 7) Kim et al, SMO, 29 (2005) 8) Wu et al, J. Electron. Sci. Technol.17 (1) 2019 9) King et al., Political Analysis 9 (2) 2001.

Development of an Automated Framework for a TinyML-Based Fall Detection System

Mojtaba Mohasel[1], Lindsey Molina[2], Shane R. Wurdeman[3], Richard R. Neptune[2], Corey A. Pew[1]
[1]Department of Mechanical and Industrial Engineering, Montana State University, Bozeman, MT [2]Walker Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX [3]Clinical and Scientific Affairs, Hanger Clinic, Austin, TX ,
Email:Seyedmojtaba.mohasel@student.montana.edu

MONTANA STATE UNIVERSITY

Introduction

- Falls are a major health risk for lower limb amputees [1].
- Falls are often underreported in surveys [2,3], more quantitative method of fall identification is needed.
- Fall detectors provide objective information on a patient's fall history, enabling improved rehabilitation intervention and device prescription.
- Machine learning (ML) techniques are commonly used to interpret body-worn inertial measurement unit (IMU) data by classifying activities of daily living (ADL) versus a fall.
- Microcontrollers are ideal for fall detection [4], but their memory hinders Deep Learning (DL) deployment.
- Manually designing memory-efficient DL for high performance is difficult and inefficient.

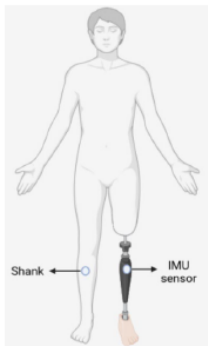


Fig 1. IMU sensor on the shanks of amputee

Purpose

- Develop a body-worn fall detector for lower limb amputees.
- Utilize automated ML (AutoML) [5] for ML and DL model creation.
- Leverage tiny machine learning (TinyML) techniques [6] for DL.

Hypothesis: DL (convolutional neural network (CNN) and gated recurrent unit (GRU)) achieves higher F-scores (harmonic average of recall and precision) compared to ML models (RUSboost [7] and EasyEnsemble [8]) on a microcontroller with 400 KB memory.

Methods (Data collection)

1. **Data collected** from 30 non-amputees and 5 lower limb amputees.
2. **Sensor placement:** IMU on both shanks (Fig 1).
3. **Trials:** ADL (walking, turning, sitting, laying, ramps and stairs) and simulated falls (forward, backward, right and left lateral, and near falls (Figs 2-5))

Fig 2. Forward fall

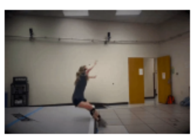


Fig 3. Backward fall

Fig 4. Lateral fall

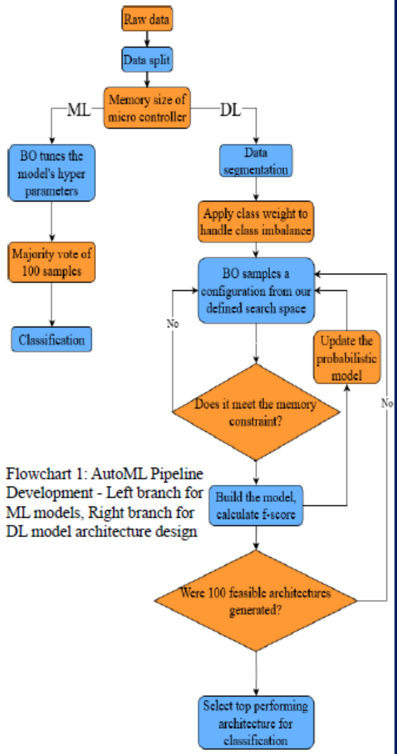
Fig 5. Near fall

Methods (Model creation)

- Espressif Systems 32-bit SoC (ESP32) microcontrollers (Fig. 6) are low-cost and low power consumption, compact, and ideal for prosthetic leg integration. Their memory (400 KB) guides model development in AutoML framework (Flowchart 1).
- Bayesian optimization (BO) tunes ML model hyperparameters: estimators, maximum depth, learning rate, and sampling strategy.

Fig 6. ESP32

- The DL model's search space involves optimizing hyperparameters such as filter number and size, stride, padding, activation function, and pooling size. It includes diverse architecture topologies with convolutional, GRU, pooling, dropout, and fully connected layers.



Flowchart 1: AutoML Pipeline Development - Left branch for ML models, Right branch for DL model architecture design

Results

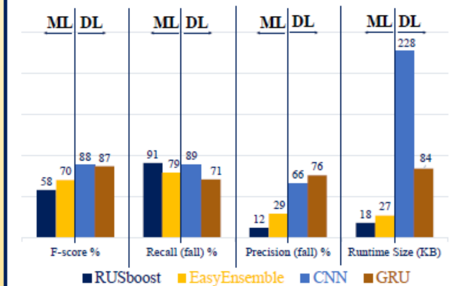


Fig 7: Comparison of Developed Models

Discussion & Conclusion

- F-score performance: DL > ML (Fig 7)
- DL performance attributed to:
  1) Time series design
  2) Enhanced data representation through feature extraction
  3) Effective weighting method for class imbalance
  4) Effective weight tuning achieved with 7 million samples
- ML models unable to use feature engineering
  - ➢ Information loss during under sampling
- ML lower precision than DL = high false alarms
- Single shank sensor on prosthesis

Future work

- Real-world data from 20 lower limb amputees (Fig 8) has been collected.
- Performance of our lab-based model (CNN) will be evaluated in real-world.
- CNN will utilize transfer learning and subject-specific data to enhance fall detection, reduce false alarms, and offer reliable information for clinicians.

Fig 8: Real-World data collection

Takeaway

AutoML Improves Fall Detection with Deep Learning Models on Microcontrollers with Limited Memory.

Acknowledgements

Funding provided by CDMRP Grant W81XWH-20-1-0164.

References

1) Miller et al, " Archives of physical medicine and rehabilitation 82(8), 2001; 2) Ganz et al, Journal of the American Geriatrics Society, 53(12), 2003; 3) Cummings et al, Journal of the American Geriatrics Society 36 (7), 1988; 4) Diab & Rodriguez-Villegas, IEEE Access 10, (2022), 5)Zöller et al., J. Artif. Intell. Res., (70), 2021. 6) Ray et al., J. King Saud Univ.-Comput. Inf. Sci., (34, 4), 2022. 7) Seiffert et al., IEEE Trans. Syst. Man Cybern. A, (40, 1), 2009. 8) Liu, T.-Y., In 2009 Int. Joint Conf. Bioinformatics, Syst. Biol. Intell. Comput., pp. 517-520, IEEE, 2009