Optimal Location for Fall Detection Edge Inferencing

Christopher Paolini
Department of Electrical and Computer Engineering
San Diego State University
San Diego, California USA
paolini@engineering.sdsu.edu

Davit Soselia
Department of Electrical and Computer Engineering
San Diego State University
Tbilisi, Georgia
dsoselia@sdsu.edu

Harsimran Baweja
School of Exercise and Nutritional Sciences
San Diego State University
San Diego, California USA
hbaweja@sdsu.edu

Mahasweta Sarkar
Department of Electrical and Computer Engineering
San Diego State University
San Diego, California USA
msarkar2@sdsu.edu

***Abstract*— A leading cause of physical injury sustained by elderly persons is the event of unintentionally falling onto a hard surface. Approximately 32-42% of those 70 and over fall at least once each year, and those who live in assisted living facilities fall with greater frequency per year than those who live in residential communities. Delay between the time of fall and the time of medical attention can exacerbate injury if the fall resulted in concussion, traumatic brain injury, or bone fracture. Several implementations of mobile, wireless, wearable, low-power fall detection sensors (FDS) have become commercially available. These devices are typically worn around the neck as a pendant, or on the wrist, as a watch is worn. Based on features collected from IMU sensors placed at sixteen body locations, and used to train four different machine learning models, our findings show the optimal placement for an FDS on the body is in front of the shinbone.**

Keywords*— *fall detection sensor, FDS, edge inferencing, machine learning

I. Introduction

Among older adults, falls are the leading cause of injury-related morbidity and mortality [2]. Recent data show that nearly 30% of community dwelling adults 65 years and older fall each year [3], and those who live in assisted living facilities fall with greater frequency per year than those who live in residential communities. In 2015, nearly 29,000,000 falls were reported, resulting in an estimated 33,000 deaths. Nearly 40% of falls require medical attention or restriction of activity [4]. Delay between the time of fall and the time of medical attention can exacerbate injury if the fall resulted in concussion, traumatic brain injury, or bone fracture. Several implementations of mobile, wireless, wearable, low-power fall detection sensors (FDS) have become commercially available to the public. These devices are typically worn around the neck as a pendant [5], or on the wrist as a watch [6], and detect when the wearer has fallen and wirelessly notify a 24/7 call center that can dispatch emergency services, or call a caretaker or family member in case of severe impact. Companies that sell these devices will typically include the following disclaimer with their device: "Fall detection does not detect 100% of falls. If Customer is able, Customer should press the help button in the event of an emergency. Fall Detection should only be worn around the neck to allow for adequate detection of falls." [7]. It is not clear that a sensor worn around the neck or wrist is the optimal body location to detect falls. In this study we captured 183 features from sixteen inertial measurement unit (IMU) sensors placed at different body locations and used these features to train four different machine learning models to classify falls. Our results show the shinbone to be the ideal sensor location.

II. Method and Results

A. Human Subjects Experiments

We conducted human subjects fall experiments in the San Diego State University (SDSU) Neuromechanics and Neuroplasticity Laboratory in the School of Exercise and Nutritional Science (ENS). Each experiment involved the placement of sixteen Noraxon Inertial Measurement Unit (IMU) sensors [8] on participating graduate students and faculty members. The specific body locations of sensor placement are shown in Fig. 1. Each experiment involved having a participant wear a virtual reality headset and walk along a straight path in the laboratory. While walking, the participant would see a virtual environment depicting a straight sidewalk in a field, with no obstructions. In the lab, a mattress was placed a distance away from the participant. Leading up to the mattress, a cloth sheet was placed on the floor, with a student firmly holding one edge of the sheet. The participant was instructed to walk

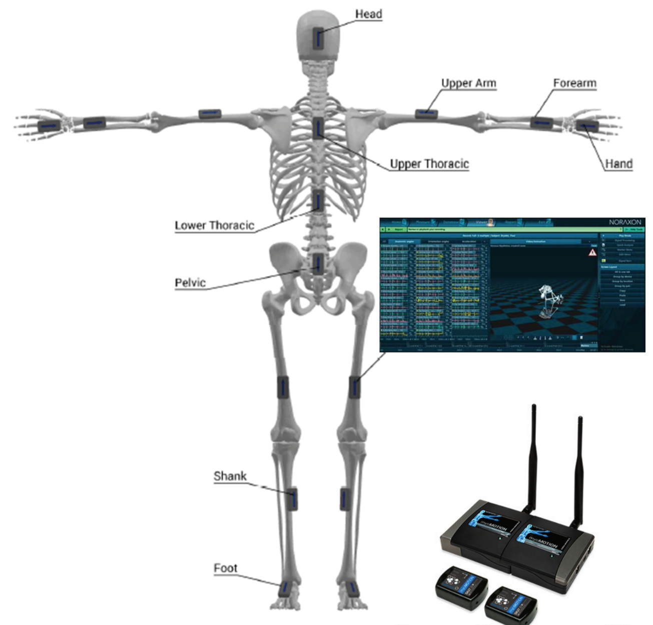

Fig. 1. Body placement locations of 16 Noraxon IMU sensors [1]. myoRESEARCH 3.14 software and myoMOTION sensors.

978-1-7281-0962-6/19/$31.00 ©2019 IEEE

forward and, at a time unbeknownst to the participant, the student holding the sheet would vigorously pull the sheet, thereby inducing a fall, stumble, trip, or no response.

The Noraxon IMU sensors recorded 183 kinematic features that measure orientation of anatomical (joint) angles and limb linear acceleration with a 200 Hz sampling rate. These sensor measurements were output to comma-separated values (CSV) files which can be read using the Noraxon myoRESEARCH software suite (also known as MR3) [9]. Each participant scenario can be read and visualized using MR3 to observe changes in limb orientation and acceleration in time (Fig. 1) and determine if the scenario resulted in a fall or near-fall. Based on visualizing the playback of each participant testcase, each testcase was labeled as having resulted in either a fall (binary value 1) or not (value 0). The labeled data was then used to train and validate four different machine learning models. Our fall dataset is available for download via URL http://iotlab.sdsu.edu/ under the *Research* section.

B. Features

The head, upper thoracic, lower thoracic, pelvic, and right and left upper arm, forearm, hand, thigh, shank (shinbone), and foot sensors all record course, pitch, and roll in degrees. In addition, x-, y-, and z-axis acceleration and rotation components are recorded in units of milligal (1 cm s^{-2}) and degrees, respectively. Combined sensor measurements are used to measure flexion, abduction, and rotation, in degrees, of the left and right shoulder and hip. Extension, radial, and supination are measured for the left and right wrist in degrees. Flexion, lateral, and axial measurements in degrees are made for the cervical, lumbar, and thoracic regions. A flexion measurement is made for the left and right knees, in degrees. A total of 183 linear acceleration, orientation, and anatomical (joint) angle features are captured in time, at a 200 Hz sampling rate, and stored in a .csv file for each participant trial.

C. Long Short-Term Memory (LSTM) Model

The transient orientation and acceleration of anatomical structures as a time series spanning a segment of time preceding a fall was used to train a recurrent neural network (RNN) using the Keras framework on top of TensorFlow. To avoid a vanishing gradient problem, a Long Short-Term Memory (LSTM) architecture [10] was configured. Shown in Fig. 3, our model consists of two LSTM layers of dimensions 183×25 and 25×20, followed by a single dense layer of dimenstion 20×1 with a sigmoid activation. The RNN/LSTM model was trained on variable segments of time leading up to, and including, a fall. Each time segment for training consisted of a different time length (period), and was fed to the model through one iteration (one forward pass and one backward pass of each batch size). The time length of input vectors for walking motion that lead to a fall was randomly chosen to lie within the range [1.5s, 7s], with a mean time of 3s. For walking motion of

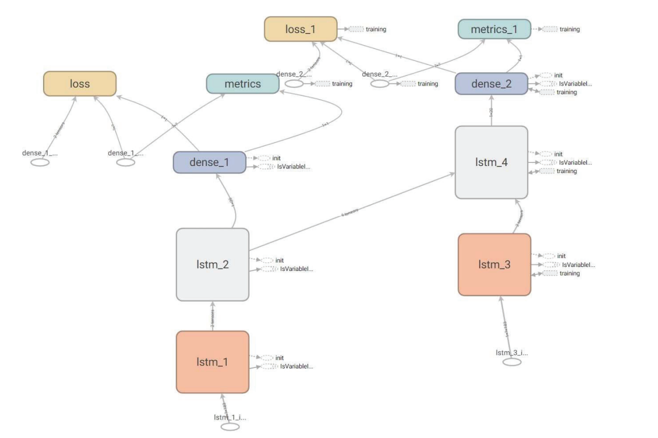

Fig. 2. RNN model consisting of two LSTM layers followed by a single dense layer.

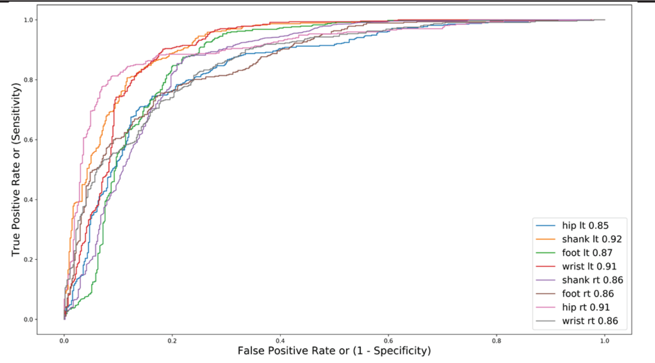

Fig. 3. Receiver operating characteristic curves (ROC) showing diagnostic ability of our RNN/LSTM binary classifier for fall detection. The "shank" sensor yielded the greatest area under curve (AUC) of 0.92.

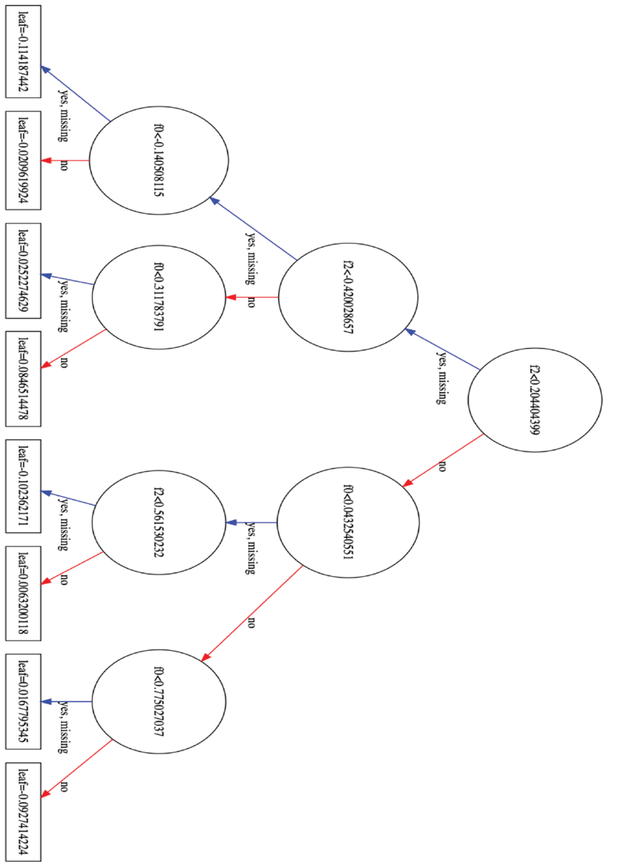

Fig. 4. Plot of the first decision tree in the model (index 0), showing the features and feature values for each split, as well as the output leaf nodes.

non- or near-falls, the range was [1.5s, 13s], with a mean time of 7s. The overall mean time length for all walking was 5s.

Results of validating the RNN/LSTM model after training are shown in Fig. 2 as receiver operating characteristic curves (ROC) for each sensor. The sensor location that yielded the greatest area under curve (AUC) was the "shank" sensor placed on the shin (AUC = 0.92). The left and right wrist locations had AUC values of 0.91 and 0.86, respectively.

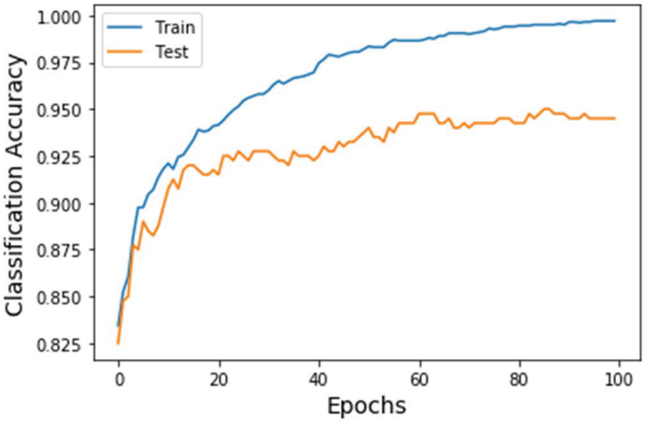

Fig. 5. Gradient boosted decision tree model accuracy as a function of the number of epochs for training and testing.

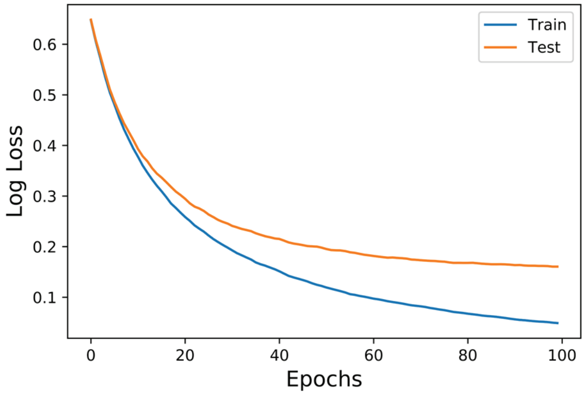

Fig. 6. Loss as a function of the number of epochs for training and testing.

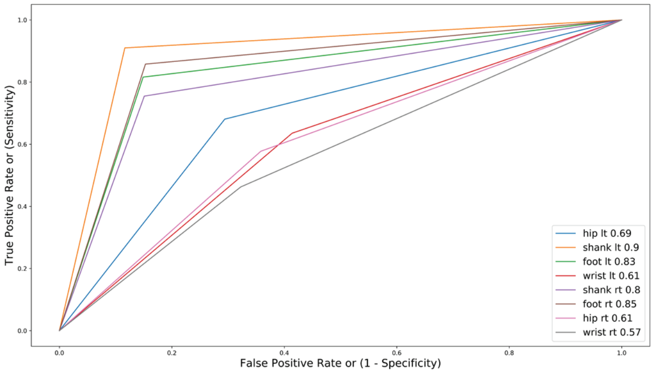

Fig. 7. ROC curves showing diagnostic ability of our gradient boosted decision tree model binary classifier for fall detection. The "shank" sensor yielded the greatest area under curve (AUC) of 0.90 at the optimal cutoff for FPR.

D. Gradient Boosted Decision Tree Model

Using the XGBoost shallow learning, scalable tree boosting framework [11], we trained a set of gradient boosted decision trees using our collected fall data. Fig. 4 shows a plot of the first trained decision tree, *$tree_0$*, in the model, showing the features and feature values for each split, as well as the output leaf nodes. An average of the ensemble of all trees yields a final classification (*fall* or *no fall*). The evaluation metric chosen for validation was *auc* (area under the ROC curve).

Gradient boosting with a binary classifier is considered state of the art for classifying non-perceptual data, such as sensor measurements. The general idea behind gradient boosting is to make a split, yielding set of predictions $y'[i]$, and then compute an error $y[i]- y'[i]$. For a binary classification (i.e. fall or no fall), we map prediction y'[i] to the [0,1] range using a logistic function $L(x)$. We can then define a mean-squared-error function $J = \Sigma(y\,[i]- y'\,[i])^2$ and a loss function $f[i] = \nabla J$ to take a step to minimize J where $y'[i] = y'[i] - \alpha\, f[i]$, for some α, and then re-iterate. Figs. 5 and 6 show gradient boosted decision tree model accuracy and loss as a function of the number of epochs (i.e. the number of entire dataset passes through the model). Fig. 7 shows ROC curves depicting the diagnostic ability of our gradient boosted decision tree model binary classifier for fall detection. The "shank" sensor yielded the greatest area under curve (AUC) of 0.90 at the optimal cutoff for false positive rate (FPR). The optimal cutoff is the FPR that yields the greatest Youden Statistic J = *sensitivity* + *specificity* – *1* [12].

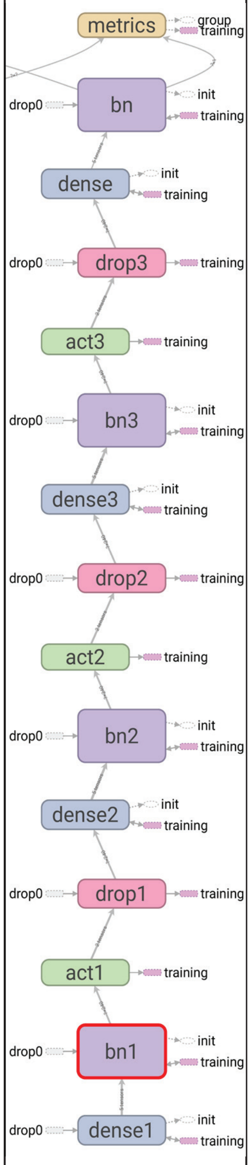

Fig. 8. Four binary dense layer BNN model for fall detection.

E. Binarized Neural Network Model

A Binarized Neural Network (BNN) is a neural network with binary weights and activations at run-time constrained to +1 or -1 [13]. We trained a BNN using four *BinaryDense* layers [14] in Keras. Fig. 8 shows an illustration of our implemented BNN model and Fig. 12 shows a plot of associated ROC curves. The left front tibia sensor position (orange colored line in Fig. 12) was found to yield the greatest AUC of 0.82. Using binary weights and activations in a binary classifier offers significant advantages when implementing an edge inferencing device. Multiply-accumulate operations, such as vector dot product, are one of the most frequent low-level linear algebra operations in deep learning inferencing. For example, the Xilinx Virtex II Pro

2 and 100 FPGAs utilize 693 and 2012 slices (sets of LUTs and flip-flops) to perform IEEE single and double precision floating-point computation, respectively [15]. With activations and weights constrained to +1 and -1, 32-bit floating point multiply-accumulations (MACs) are replaced by 1-bit XNOR-popcount operations. Popcount is a binary operation where the input is a (32-bit) word and the output is the number of bits in the word set to 1. The standard vector dot product operation is replaced in the BNN by the binarized variant that can be implemented using a popcount-XNOR operation

```
  for each i:
    C += A[row][i] * B[i][col]
```

```
 for each i:
 C +=popcount(XNOR(A[row][i],B[i][col]))
```

BNNs provide a power efficient forward pass and require a 32 times smaller memory size and 32 times fewer memory accesses than 32-bit deep neural networks (DNNs) [13].

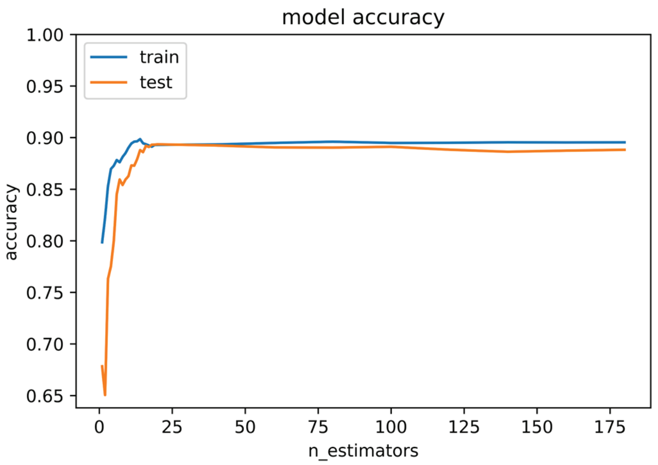

Fig. 9. Random forest classifier model results from training and testing our fall dataset. Optimal classification accuracy was found with *n_estimators=10*.

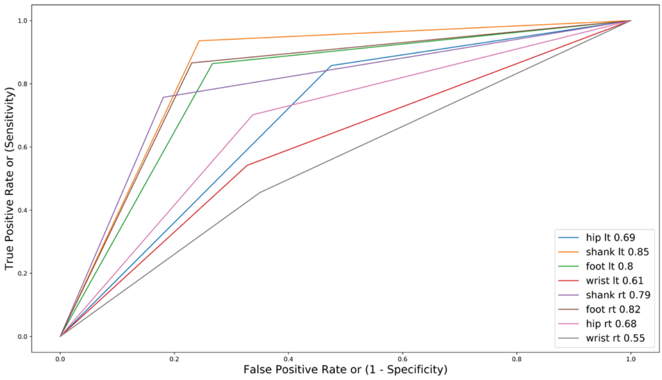

Fig. 10. ROC curves showing resuts of testing our random forests classifier model. The left "shank" (front of left tibia) sensor position (orange colored line) was found to yield the greatest AUC (.85).

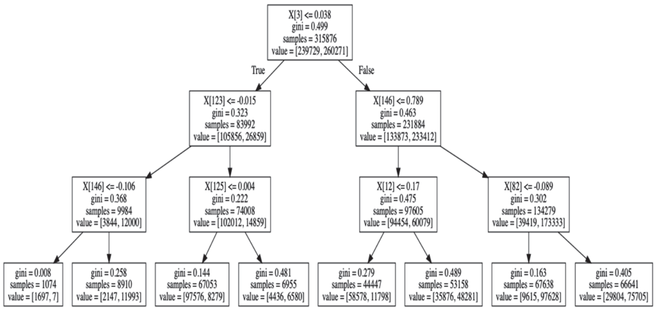

Fig. 11. Plot of the first decision tree in our random forests model (index 0) showing the Gini impurity index in each node. The Gini impurity is a metric that gives the probability of an incorrect classification of a randomly selected test datum from our labeled fall dataset, if the selected test datum was randomly labeled in either the *fall* or *no-fall* class.

F. Random Decision Forest Model

The fourth machine learning model we examined for fall classification was random decision forests. We employed the *RandomForestClassifier()* model from the *sklearn.ensemble* library [16]. Fig. 9 shows a plot of model accuracy for training and testing data from our fall dataset. Optimal classification accuracy was found with 10 trees in the forest (i.e. *n_estimators=10*). Fig. 10 shows a plot of ROC curves demonstrating the diagnostic ability of the random forest model to correctly classify falls. The left "shank" sensor position (orange colored line) was found to yield the greatest AUC (.85). The optimal cutoff is the FPR that yields the greatest Youden Statistic. Fig. 11 shows a plot of trained *tree*$_0$ from our random forests model with calculated Gini impurity indices [36].

In the gradient boosted trees model, training is performed sequentially to minimize the mean-squared-error function *J* during each iteration. The random forests model, however, utilizes a set (10 in our approach) of independent trees where training is performed in parallel on all trees and a single classification result is obtained by forming an aggregated solution from the ensemble of trees.

G. Fall Dataset Training and Testing

We used an 80:20 split of our collected fall data, 80% for training and 20% for testing. The Keras open-source neural-network library running on top of TensorFlow and the XGBoost

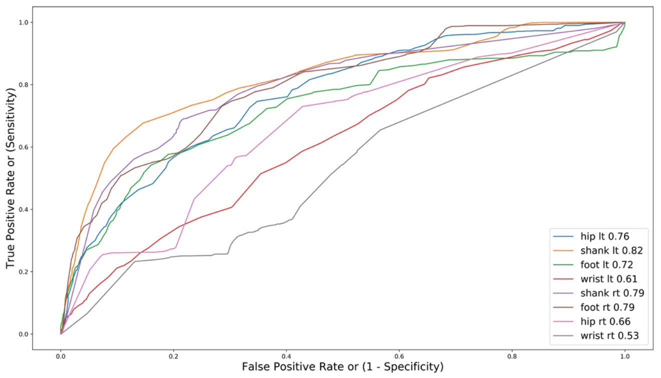

Fig. 12. ROC curves showing resuts of testing our BNN model. The left "shank" (front of left tibia) sensor position (orange colored line) was found to yield the greatest AUC (.82).

open-source gradient boosting framework was executed on a Supermicro SuperServer SYS-4028GR-TR [37] with eight NVIDIA Tesla V100 PCle 16GB GPUs [38].

III. Application

As reported by the World Health Organization's Global Report on Falls Prevention in Older Age, based on research conducted by Blake et al. [3, 19], 35% of persons 65 years and over experience at least one unintentional fall per year, due to tripping, dizziness, and blackouts. Elderly people are more likely to fall due to loss of handgrip strength used for control and stabilization when using walking aids such as walkers and canes. Arthritis, dizziness, neuromuscular, cognitive, and foot impairments also contribute to increased prevalence of falling by older people when using stairs and steps, or while turning around or reaching for objects. Research by Tinetti found that 61% of elderly nursing home residents fell at some point during their first year of residence [20], a greater proportion than elderly people who live in residential communities. Those assigned to live in assisted living facilities my fall more frequently due to greater weakness in their lower extremities, a potential contributing factor for their decision to live in such a facility, since facility staff can assist with mobility. When an elderly person falls, the resulting impact can result in significant injury due to concussion, traumatic brain injury, or bone fracture. Research by Nevitt et al. found that over half (55%) of falling incidents by elderly people resulted in minor soft tissue injury, while 6% resulted in a major injury requiring emergency hospitalization, such as a bone fracture, joint dislocation, or skin laceration requiring suture [21]. Reported by Alexander et al., 5.3% of all hospitalizations of older adults are due to fall-induced physical trauma [22]. In addition to trauma, falls among elderly persons that result in an emergency room hospital visit can pose an adverse economic consequence on the patient, and impose a burden on medical, health insurance, and social service providers [23].

When a fall does occur, any delay between the time of fall and the time of medical attention can exacerbate injury if the fall resulted in concussion, traumatic brain injury, bone fracture, or a major skin laceration that results in bleeding [24-26]. Elderly people who fall outdoors in remote locations who are undetected and unable to get up are at risk of suffering hypothermia. The fifth leading cause of death among elderly people is due to unintentional injury, of which two-thirds are death resulting from a fall [27].

To accurately detect when a person has fallen, calculate precise geographic coordinates of the person, and notify emergency medical or living facility personnel the identity and location of the person, a Fall Detection System (FDS) is required to be worn by elderly persons who are at risk of falling. Because those who live in assisted living facilities fall with greater frequency per year than those who live in residential communities, and because GPS signals may be significantly attenuated within buildings, mobile, wireless, wearable, low-power, medical devices that use a small field-programmable gate array (FPGA) for in-situ fall detection (i.e. *edge inferencing*) using machine learning models can be realized. Such devices can be integrated with pressure, gyroscope, accelerometer, and image sensors, that may use, for example, binarized and convolutional neural networks to detect when the device wearer has fallen. The device would immediately notify living facility staff of the wearer's physical location, using GPS or indoor trilateration, if the subject is outdoors or indoors, respectively [28].

IV. Conclusion

Technologies that perform machine learning in situ to make predictions and take corrective actions, in real-time, are beginning to show application in medical device engineering. Known as *inferencing at the edge*, these technologies allow one to perform computationally intensive calculations on the device, without needing to transfer sensor measurements to a central computational server that performs calculations and then transfer a solution back to the device over a wireless connection. Small form-factor FPGAs are now commercially available that provide *soft* or flexible intellectual property (IP) cores that implement Binarized Neural Networks (BNNs) that can perform continuous fall detection in the mW power range [29].

Published datasets on Activities of Daily Life (ADL) and fall movement, such as those developed at universities in the European Union using student test subjects [30, 31] have used chest, thigh, trouser pocket, waist, and wrist [32, 33] locations on the body to capture accelerometer, gyroscope, and limb orientation sensor features. Existing solutions with sensors placed at these locations only achieve an 80% classification accuracy [35].

All four machine learning models we implemented show the left shank body location to be the optimal sensor position. The left shank location yielded the greatest AUC in all four models. Gradient boosted trees, random forests, and BNN models show similar classification performance, likely because each of these models analyze a single point in time, rather than a segment of time, as the RNN model does. Our results show the wrist location is a poorly performing sensor placement position relative to other body locations.

A possible reason for the superiority of the RNN model may be that the RNN model observes a time period, and the changes in lower leg orientation and position in time during walking and falling motion provides a temporal feature set that captures the context of a sequence of body motions that characterize a fall.

Developers of wearable fall detection systems may wish to consider the shinbone region for sensor placement to achieve greater classification accuracy. We have made our respective data available for validation, evaluation, and further collaboration via URL http://iotlab.sdsu.edu/ under the *Research* section. The lead author has recently been awarded a two-year grant through the National Institute on Minority Health and Health Disparities (NIMHD) to conduct a pilot study to develop a prototype wireless, wearable, low-power *edge inferencing* FDS using a very low-power (1 mW scale) FPGA to predict an imminent fall and detect when a fall occurs, for elderly at-risk adults. An outcome of the pilot project will be to assess real-time performance, sensor comfort, ease of user attachment and removal, and feasibility of implementing our RNN model on a small, low power device. We plan to conduct human-subjects experiments involving 200 unique participants in year one and 190 unique participants in year two.

ACKNOWLEDGMENT

This work was supported by the University Grants Program (UGP) of San Diego State University.

REFERENCES

[1] Noraxon U.S.A. Inc., myoMOTION Sensor and Receiver User Manual, 2018. [Online]. Available: https://www.noraxon.com/noraxon-download/myomotion-system-user-manual/.

[2] E. Burns and R. Kakara, "Deaths from Falls Among Persons Aged ≥65 Years − United States, 2007‑2016," MMWR Morb Mortal Wkly Rep, vol. 67, pp. 509‑514, 2018.

[3] A. J. Blake et al., "Falls by elderly people at home: prevalence and associated factors," (in eng), Age Ageing, vol. 17, no. 6, pp. 365-72, Nov 1988.

[4] U. S. Preventive Services Task Force, "Interventions to Prevent Falls in Community-Dwelling Older Adults: US Preventive Services Task Force Recommendation StatementUSPSTF Recommendation: Interventions to Prevent Falls in Community-Dwelling Older Adults USPSTF Recommendation: Interventions to Prevent Falls in Community-Dwelling Older Adults," JAMA, vol. 319, no. 16, pp. 1696-1704, 2018.

[5] LifeFone Medical Alert Services, "LifeFone with Fall Detection," 2018.

[6] MobileHelp. (2019). MobileHelp Fall Button™. Available: https://buy.mobilehelp.com/

[7] Medical Alert, "Medical Alert System," 2018.

[8] Medical Guardian. (2019). Home Guardian Fall Alert Pendant. Available: https://www.medicalguardian.com/products/home-guardian

[9] Apple Inc. (2019). Use fall detection with Apple Watch Series 4. Available: https://support.apple.com/en-us/HT208944

[10] Noraxon USA. (2018). Research PRO IMU. Available: https://www.noraxon.com/our-products/portable-motion-capture/

[11] Noraxon U.S.A. Inc. (2018). myoRESEARCH 3.14 User Manual. Available: https://www.noraxon.com/noraxon-download/myoresearch-3-14-user-manual/

[12] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997/11/01 1997.

[13] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," presented at the Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, 2016.

[14] W. J. Youden, "Index for rating diagnostic tests," Cancer, vol. 3, no. 1, pp. 32-35, 1950/01/01 1950.

[15] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or -1," arXiv, vol. abs/1602.02830, 2016.

[16] K. Ding. (Aug 15, 2017). Binarized Dense and Conv2D Keras layers. Available: https://github.com/DingKe/nn_playground/tree/master/binarynet

[17] M. J. Beauchamp, S. Hauck, K. D. Underwood, and K. S. Hemmert, "Embedded floating-point units in FPGAs," presented at the Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays, Monterey, California, USA, 2006.

[18] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825-2830, 2011.

[19] "WHO Global Report on Falls Prevention in Older Age," in "Ageing and Life Course, Family and Community Health," World Health Organization2007.

[20] M. E. Tinetti, "Factors associated with serious injury during falls by ambulatory nursing home residents," (in eng), J Am Geriatr Soc, vol. 35, no. 7, pp. 644-8, Jul 1987.

[21] M. C. Nevitt, S. R. Cummings, and E. S. Hudes, "Risk factors for injurious falls: a prospective study," (in eng), J Gerontol, vol. 46, no. 5, pp. M164-70, Sep 1991.

[22] B. H. Alexander, F. P. Rivara, and M. E. Wolf, "The cost and frequency of hospitalization for fall-related injuries in older adults," (in eng), Am J Public Health, vol. 82, no. 7, pp. 1020-3, Jul 1992.

[23] P. Scuffham, S. Chaplin, and R. Legood, "Incidence and costs of unintentional falls in older people in the United Kingdom," Journal of Epidemiology and Community Health, 10.1136/jech.57.9.740 vol. 57, no. 9, p. 740, 2003.

[24] N. Noury et al., "Fall detection - Principles and Methods," in 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007, pp. 1663-1666.

[25] R. J. Gurley, N. Lum, M. Sande, B. Lo, and M. H. Katz, "Persons found in their homes helpless or dead," (in eng), N Engl J Med, vol. 334, no. 26, pp. 1710-6, Jun 27 1996.

[26] D. Wild, U. S. Nayak, and B. Isaacs, "How dangerous are falls in old people at home?," British Medical Journal (Clinical research ed.), vol. 282, no. 6260, pp. 266-268, 1981.

[27] L. Z. Rubenstein, "Falls in older people: epidemiology, risk factors and strategies for prevention," (in eng), Age Ageing, vol. 35 Suppl 2, pp. ii37-ii41, Sep 2006.

[28] K. Gala, P. Bryden, M. Wang, C. Paolini, M. Sarkar, and A. Dimitrova, "Real-Time Indoor Geolocation Tracking for Assisted Healthcare Facilities," presented at the Wireless Telecommunications Symposium (WTS 2019), New York City, April 9-12. 2019.

[29] "Accelerating Implementation of Low Power Artificial Intelligence at the Edge," Lattice SemiconductorMay 2018, Available: http://www.latticesemi.com/view_document?document_id=52384.

[30] E. Casilari, J. A. Santoyo-Ramón, and J. M. Cano-García, "UMAFall: A Multisensor Dataset for the Research on Automatic Fall Detection," Procedia Computer Science, vol. 110, pp. 32-39, 2017/01/01/ 2017.

[31] S. B. Khojasteh, J. R. Villar, E. de la Cal, V. M. González, and J. Sedano, "Fall Detection Analysis Using a Real Fall Dataset," in International Joint Conference SOCO'18-CISIS'18-ICEUTE'18, Cham, 2019, pp. 334-343: Springer International Publishing.

[32] S. Abbate, M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini, and A. Vecchio, "A smartphone-based fall detection system," Pervasive Mob. Comput., vol. 8, no. 6, pp. 883-899, 2012.

[33] S. B. Khojasteh, J. R. Villar, C. Chira, V. M. González, and E. de la Cal, "Improving Fall Detection Using an On-Wrist Wearable Accelerometer," (in eng), Sensors (Basel, Switzerland), vol. 18, no. 5, p. 1350, 2018.

[34] Purch.com. (Jan 3, 2019). Best Fall Detection Sensors of 2019. Available: http://www.toptenreviews.com/health/senior-care/best-fall-detection-sensors/

[35] F. Bagala et al., "Evaluation of accelerometer-based fall detection algorithms on real-world falls," (in eng), PLoS One, vol. 7, no. 5, p. e37062, 2012.

[36] Leo Breiman, J. F., Charles J. Stone, R.A. Olshen (January 1, 1984). Classification and Regression Trees, Chapman and Hall/CRC.

[37] Super Micro Computer, I. (2019). "SuperServer 4028GR-TR." from https://www.supermicro.com/products/system/4U/4028/SYS-4028GR-TR.cfm.

[38] NVIDIA Corporation (2018). "NVIDIA® Tesla® V100 GPU Accelerator." from https://images.nvidia.com/content/technologies/volta/pdf/tesla-volta-v100-datasheet-letter-fnl-web.pdf.