Design Flow of Neural Network Application for IoT based Fall Detection System

Won-Jae Yi, Boyang Wang, Bruno Fernandes dos Santos, Eduardo Fonseca Carvalho and Jafar Saniie
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago, Illinois, USA

***Abstract*— In the remote health monitoring system, it is crucial to identify and analyze the current users' status accurately. The accuracy depends on many different aspects including physical conditions, surrounding environmental conditions, users' distinct features and other factors. In this paper, we investigate the enhacement possibility of IoT based health monitoring system by applying neural network. By training the collected user data from different types of medical emergency-related scenarios, the system would gain better accuracy over the traditional thresholding data analysis systems. In this study, we focus on applying neural network to the fall detection application which involves wireless wearable sensors with accelerometers and a gyroscope. We utilize multilayer perceptron neural network to train user movement datasets including positive falls (falling events) and negative falls (non-falling events). This system design approach has the potential to be extended to multi-purpose user activity and health monitoring system, including people who have potential in needs of medical attentions and daily activity tracking.**

I. Introduction

Internet of Things (IoT) is described as interconnected physical objects designed to communicate over the Internet for exchanging valuable information. With the IoT infrastructure and associated technology, quality of life can be improved by means of distributed wireless embedded devices [1] [2] [3]. In the field of healthcare, there are many studies involved in utilizing the Internet to monitor users who are distant, immobilized and require constant care [4] [5]. Conventional remote health monitoring systems focus on a certain type of medical conditions such as monitoring cardiac activity through ECG (electrocardiogram) sensor, blood sugar level monitoring using glucose sensor, and many others including pulse oximeter, air flow sensor, and blood pressure sensor [6] [7] [8] [9]. These sensors are typically worn or attached to the remote patient, where a comprehensive diagnosis is achieved by the physician based on the recorded information. In [10], we introduced a system architecture that combines multiple wearable and environmental sensors to determine the cause of a sudden fall. Wearable sensors including accelerometers and a gyroscope were utilized to obtain real-time posture and detect fall events. Calculated heart beat rate from ECG sensor, and on-body temperature sensor data were also sent together to the server, so that the physicians could determine the cause of the fall. An Android smartphone was utilized to add mobility to the user, where the analyzed data is either received via the Bluetooth or the Internet connection. By this configuration, all devices would be online at any time, and be accessible over the Internet. Figure 1 is an overview of the real-time health monitoring system.

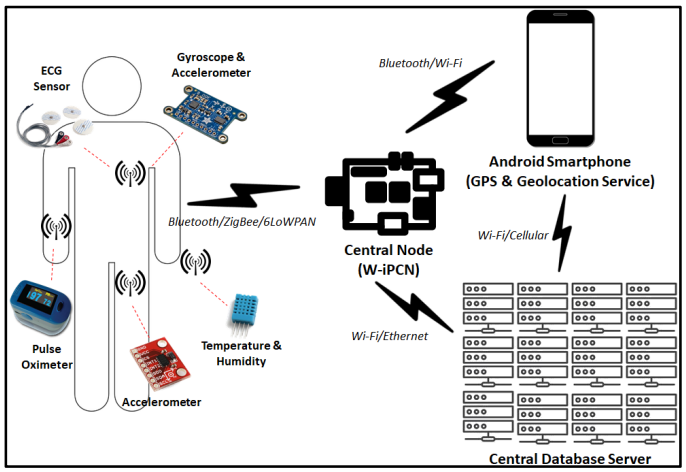

Figure 1. Overview of Real-time Health Monitoring System

In this configuration, user status warning system is usually subordinate to only one abnormal sensor data that would trigger a warning signal. Also, this is achieved based on the thresholding mechanism on each individual sensor data. For example, any sensor reading goes lower or higher than the predefined threshold value, the system will generate an alert and warning signals will be sent to the designated person. Coupling with IFTTT (If This Then That) scheme [11], an appropriate action could be executed based on the trigger generated from the analyzed user data. For example, a high room temperature reading could trigger an action of sending an alert to the caregiver and turning on the air conditioner in the room.

Drawback of this system configuration is that thresholds must be adjusted for every user who uses this system. For example, if the warning signal thresholds for heart beat rate is set to 75 beats per minute for a patient whose regular heart beat rate around 90 beats per minute would trigger a false alarm at

978-1-5386-5398-2/18/$31.00 ©2018 IEEE

any time. Even for the same user, threshold values may need to be adjusted depending on the type of the user's activity. To overcome the limitation of the thresholding-based monitoring system, a system applied with neural network for analyzing user sensor data is introduced. The neural network in this system is trained by the collected sensor data deployed on the user, and its surrounding environmental sensors. In this configuration, the system would deliver more precise analysis of the user's condition. In this paper, we have applied to our fall detection application [12] [13], utilizing two accelerometers and a gyroscope worn on the user, to demonstrate the feasibility of the applied neural network and evaluate its accuracy.

II. System Design

In our existing system design, we focused on applying the Bluetooth protocol to deliver sensor data to the central node, known as the Wireless Intelligent Personal Communication Node (W-iPCN) [14] (shown in Figure 1). However, due to its limitation on concurrent multiple connections and relatively high battery consumption, a different wireless protocol has been applied to the system. The ZigBee protocol is a wireless communication based on the IEEE 802.15.4 standard. This protocol is intended to be used on low-power, low-cost and embedded systems with less-requiring bandwidth devices. It is widely used in various areas in the market such as smart homes, connected lighting, utility industries and more [15]. Due to its low power consumption, it has a common usage on control systems and Machine-to-Machine (M2M) applications [16]. By following the IEEE 802.15.4 standard, the ZigBee operates in 868 MHz, 900 MHz or 2.4 GHz frequencies, depending on the requirements of transmission distance and throughput. This protocol has become more popular with the IoT applications. Other than low power consumption features of this protocol, it allows to create a variety of network topologies for different purposes. It is flexible to form Point-to-Point (P2P) network between two nodes, and to form mesh networks where it is not limited to communicate to other nodes in only specific route, but rather flexible depending on the available network resource. Due to these facts, ZigBee transceivers replaced the Bluetooth transceivers on the sensor node, where the multiple connection between the W-iPCN and the sensor nodes lack flexibility and more complex when using the Bluetooth. Table 1 is a brief comparison between the Bluetooth and ZigBee specifications. Even though the Bluetooth provides higher bandwidth for data transmissions, the ZigBee provides longer battery life of the device which is more important in real-time persistent monitoring system. Also, utilized sensors, in an average, does not require more than the bandwidth of the ZigBee protocol for real-time data transmission. In addition to the flexibility of adding more sensors to the network, the ZigBee is more reliable compared to the Bluetooth configuration.

Table I. Bluetooth and ZigBee Specifications

Features	ZigBee	Bluetooth
Standard	IEEE 802.15.4	IEEE 802.15.1
Topology	Mesh, Star, Tree	Star
Bandwidth	250 Kbps	1 Mbps
Maximum Nodes	65,536	1 master, 7 slaves
Power Profile	Very Low (Months ~ Years)	Low (Days ~ Weeks)
Range	100 m+ (984 ft+)	10 m (32 ft)
Complexity	Simple	Complex

Specifically, two accelerometers and a gyroscope are used in this paper to collect user's movement, where the accelerometer and gyroscope used in this system are triaxial sensing devices that can be used to sense the direction and the angular rate of the movement. They are coupled with a microcontroller called Arduino Pro Mini and the ZigBee transceiver. The Arduino boards are one of the microcontroller boards with highly customizable flexibility in peripherals and other external devices. In particular, the Arduino Pro Mini is a smaller form factor of the regular Arduino boards with ATmega328 running at 8 MHz with 3.3V, which does not require any interfacing circuits to external modules [17]. Plentiful of open-source libraries are helpful and easy to add other peripherals to the Arduino boards such as the accelerometer, gyroscope, temperature sensor, ECG sensor and more. As mentioned in the previous section, as a health monitoring system, it is important to determine other factors of the cause of the emergency event. Other than the accelerometer and gyroscope, a temperature sensor and a ECG sensor can be attached to the Arduino board for data transmission over the ZigBee transceiver. Figure 2 and Figure 3 illustrate two sensor nodes that would be deployed on the chest and the thigh of the user.

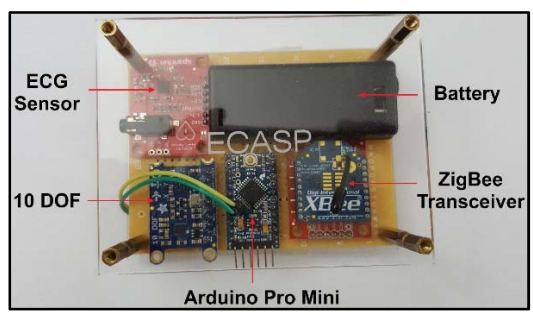

Figure 2. Overview of Chest Node

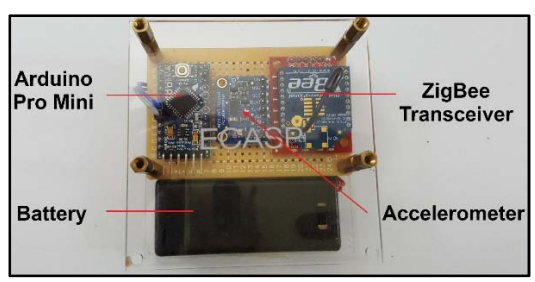

Figure 3. Overview of Thigh Node

In Figure 2, the chest node has an Arduino Pro Mini with ATmega328 operating at 3.3V at 8 MHz, a XBee Series 2 ZigBee transceiver set as router API 2 mode [18], a 10 DOF breakout board containing an accelerometer, a gyroscope and a temperature sensor [19], and a AD8232 Heart Monitor sensor which all are connected and soldered onto the PCB along with 2 AAA batteries. As shown in Figure 3, the thigh node is also equipped with the same microcontroller and the ZigBee transceiver also set as a router API 2 mode [18], and a ADXL345 accelerometer [20] all soldered onto the PCB along with 2 AAA batteries to supply 3.3V to the devices. These two nodes are attached and strapped on to the user for experiment purposes as shown in Figure 5, where Sensor 1 is the chest node, and Sensor 2 is the thigh node.

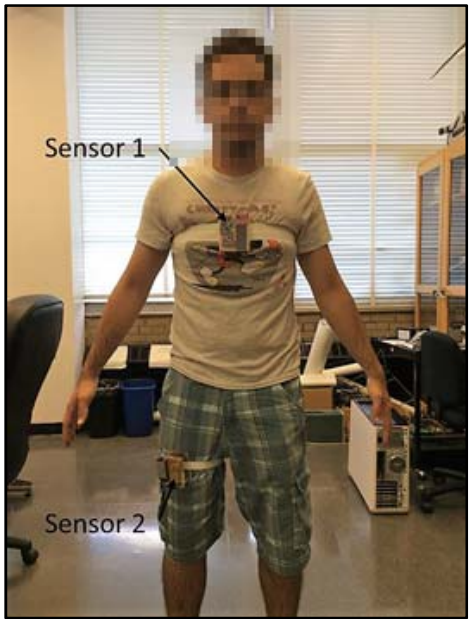

Figure 5. Sensor Nodes Configuration (Sensor 1: Chest Node, Sensor 2: Thigh Node)

These two nodes automatically connect to the W-iPCN, where a Raspberry Pi 3 [21] is utilized in this study with a ZigBee transceiver attached via a USB port through the serial communication. Internet connection on this W-iPCN is possible by either using built-in Wi-Fi or the Ethernet port to establish connections to the database, as well as to the Android smartphone. Figure 6 shows the W-iPCN using Raspberry Pi 3 and the XBee Series 2 ZigBee transceiver connected via the USB port. This ZigBee transceiver is configured as coordinator API 2 mode [18] and communicates with both sensor nodes in router API 2 mode [18] and stores all accelerometer and gyroscope data with timestamps for training the neural network of the fall detection system.

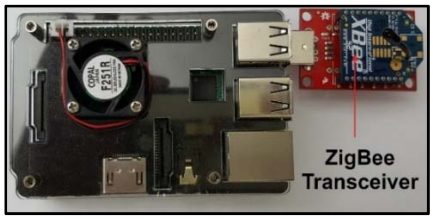

Figure 6. W-iPCN with ZigBee Transceiver connected via USB Port

III. Applied Neural Network for Fall Detection

As described in the previous section, the datasets used in training the neural network are generated from the two nodes worn on the user, as shown in Figure 5. Each node communicates with the W-iPCN concurrently via the ZigBee transceivers set as router API 2 mode, where the ZigBee transceiver on the W-iPCN is set as coordinator API 2 mode. There are two different types of datasets used in the neural network training where the positive fall datasets are collected data from different types of fall scenarios, and the negative fall datasets are collected data from different types of non-falling scenarios including standing up to sitting down, sitting down to standing up, walking, jogging and other test cases. Figure 6 shows portion of the sampled fall datasets from the chest and thigh nodes at 20 Hz.

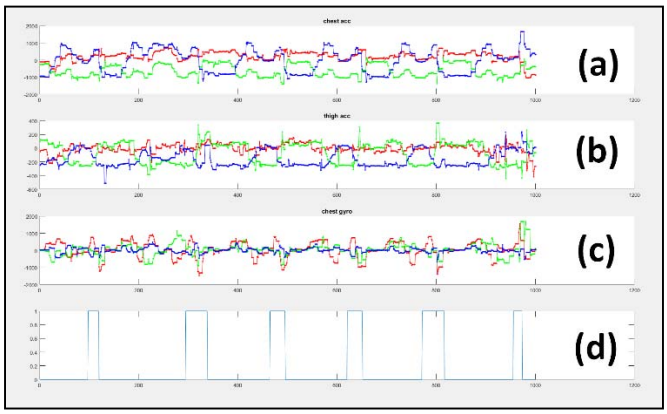

Figure 6. Sampled Fall Datasets

Figure 6a represents the 3-axis accelerometer data from the chest node, Figure 6b represents the 3-axis accelerometer data from the thigh node, and Figure 6c represents the 3-axis gyroscope data from the chest node. The start and end sequence of the positive fall events are marked as shown in Figure 6d, where 0 means a negative fall (non-falling event) and 1 means a positive fall (falling event). Positive fall events always come with big variance of sensor data. Machine learning algorithms can automatically detect falls after trained by a certain amount of training datasets. We have used the data presented in Figure 6 as the training input for the multilayer perceptron neural network [22]. In this paper, TensorFlow [23] is used to build and train the neural network. The TensorFlow environment is built based on a desktop computer equipped with an Intel i7-6700 CPU and an NVIDIA GeForce GTX 1070 GPU.

One of the most important tasks of training the neural network is to prepare the training input. Figure 7 is a frame of 30 samples segmented from the original dataset which is used as the training input for the neural network.

Figure 8 is the system diagram of the fall detection system based on the neural network. First, all accelerometer and gyroscope data are buffered and re-organized. The buffered data then are passed into the neural network. The neural network is trained by using backward propagation algorithm with falling procedures manually marked as "1"s. The prediction result of the neural network is passed into another

buffer. Finally, the output of the fall detection is generated by the summation of the buffered data.

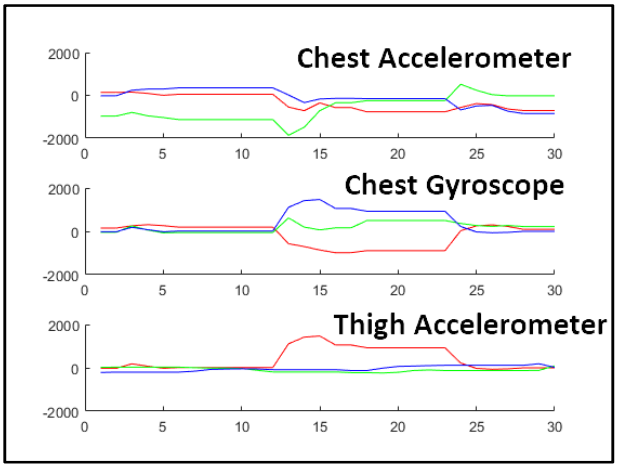

Figure 7. One Training Input Dataset

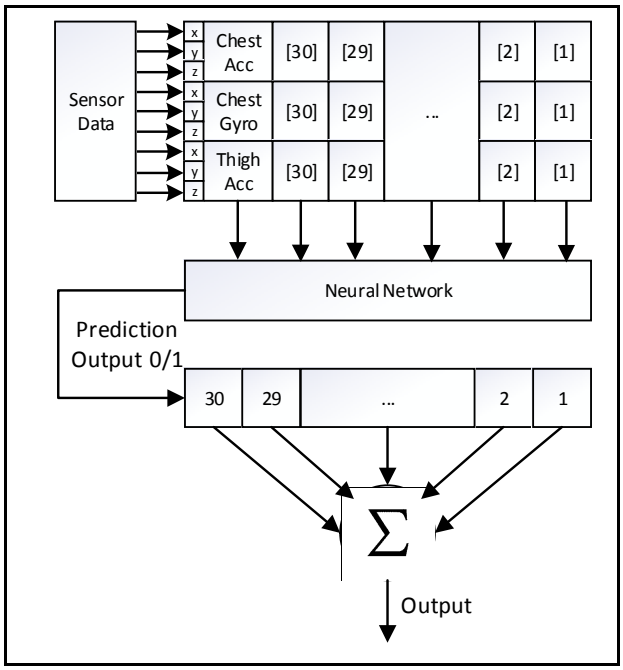

Figure 8. System Diagram of the Neural Network

Figure 9 shows the plot of randomly selected training input. The dataset is obtained by merging all 9 axes sensor data, 3 axes per sensor, into one. The datasets plotted in red mean that they are positive falls, and in blue represent for negative falls. Each frame of data contains 270 data points, meaning the data is in a 270-dimension. To gain a better intuition of this high dimension dataset, an algorithm called Principle Component Analysis (PCA) [24] is used to project the high dimension data into 3D dimension. Figure 10 shows the PCA projection of the 270-dimension data using the Embedding Projector [25]. The fall transitions are marked in orange and the negative fall data are marked in blue. This indicates that the neural network is able to differentiate the positive fall events from negative fall events accurately.

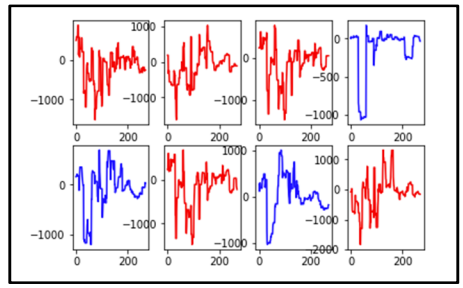

Figure 9. Plots of Randomly Selected Training Inputs

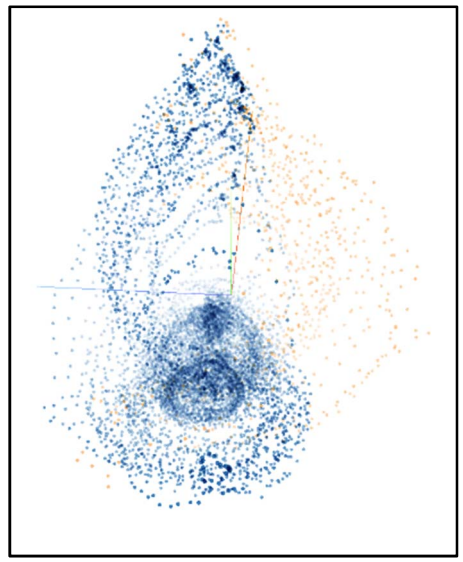

Figure 10. Datasets projected in 3D Space using PCA Algorithm

Figure 11 shows the forward propagation prediction result on the test dataset. In this figure, positive fallings are marked in red colored bars. Green colored bars indicate a series of sitting down to standing up transitions and blue colored bars shows a series of standing up to sitting down transitions. Prediction results are the results obtained from the neural network, and manually marked falls are the fall events marked manually for comparison. Many other position transitions are also tested with the system. The result shows that the neural network based fall detection system is capable of detecting fall events from other position transitions precisely.

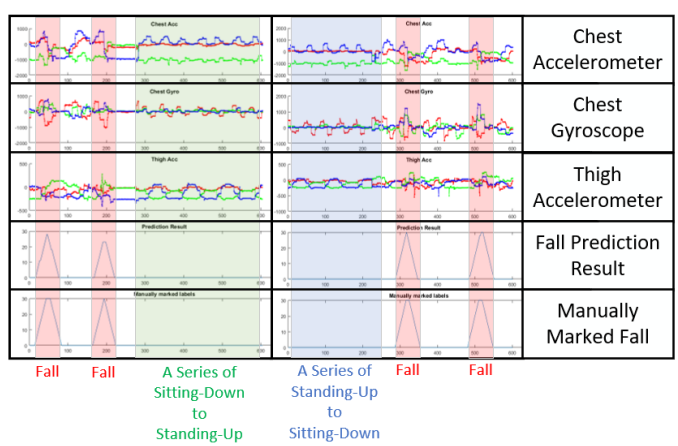

Figure 11. Prediction Results of the Neural Network

IV. Conclusion

In this paper, we have introduced a design flow of neural network application for real-time health monitoring system, specifically for the fall detection application. The neural network is trained based on the movement data obtained from the real-world user datasets from wearable accelerometers and a gyroscope. Collected movement data used for neural network training contains positive and negative fall events, where negative fall events include transitions from/to standing up and sitting down, walking, jogging and other non-falling activities. The result of the neural network training shows accurate fall detections by distinguishing the falling and non-falling events. This system configuration has the potential to be extended for a complete and advanced health monitoring system by the fusion of many sensor datasets to be trained using the neural network. This technique does not limit towards fall detections, but extensive and precise remote diagnosis solution for users who needs constant health monitoring.

References

[1] L. Knott, "Prevention of Falls in the Elderly. Information on elderly falls | Patient," EMIS Group, 7 July 2013. [Online]. Available: http://patient.info/health/prevention-of-falls-in-the-elderly-leaflet.

[2] C. Reyes, H. Vaca, M. Calderon, L. Montoya and W. Aguilar, "NilNova: An Approach to the IoT Solution based on Model-driven Engineering for the Military Health Monitoring," *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON),* pp. 1-5, 2017.

[3] A. Kamilaris and A. Pitsillides, "Mobile Phone Computing and the Internet of Things: A Survey," *IEEE Internet of Things Journal,* vol. 3, no. 6, pp. 885-898, 2016.

[4] M. Yu, A. Rhuma, S. Naqvi, L. Wang and J. Chambers, "A Posture Recognition-based Fall Detection System for Monitoring and Elderly Person in a Smart Home Environment," *IEEE Transactions on Information Technology in Biomedicine,* vol. 16, no. 6, pp. 1274-1286, 2012.

[5] J. Stevens, L. Hasbronck, T. Durant, A. Dellinger, P. Batabyal, A. Crosby, B. Valluru, M. Kresnow and J. Guerrero, "Surveillance for Injuries and Violence among Adults," *Morbidity and Mortality Weekly Report,* vol. 48, no. 17, pp. 27-50, 1999.

[6] S. Chen, H. Lee, C. Chen and C. Luo, "Wireless Body Sensor Network With Adaptive Low-Power Design for Biometrics and Healthcare Application," *IEEE Systems Journal,* vol. 3, no. 4, pp. 398-409, 2009.

[7] H. Ghasemzadeh, V. Loseu and R. Jafari, "Structural Action Recognition in Body Sensor Networks: Distributed Classification Based on String Matching," *IEEE Transactions on Information Technology in Biomedicine,* vol. 14, no. 2, pp. 425-523, 2010.

[8] C. James and S. Kumar, "Detection of Posture and Motion by Accelerometer Sensors," *3rd International Conference on Electronics Computer Technology,* pp. 639-371, 2011.

[9] I. Korhonen, J. Parkka and M. Van Gils, "Health Monitoring in the Home of the Future," *IEEE Engineering in Medicine and Biology Magazine,* vol. 22, no. 3, pp. 66-73, 2003.

[10] W.-J. Yi and J. Saniie, "Patient Centered Real-Time Mobile Health Monitoring System," *E-Health Telecommunication Systems and Networks,* vol. 5, no. 4, pp. 75-94, 2016.

[11] IFTTT, "IFTTT helps your apps and devices work together," IFTTT, 2018. [Online]. Available: https://ifttt.com/. [Accessed 11 2 2018].

[12] W.-J. Yi, O. Sarkar, S. Mathavan and J. Saniie, "Wearable Sensor Data Fusion for Remote Health Assessment and Fall Detection," *2014 IEEE International Conference on Electro/Information Technology (EIT),* pp. 303-307, 2014.

[13] W.-J. Yi, O. Sarkar, S. Mathavan and J. Saniie, "Design Flow of Wearable Heart Monitoring and Fall Detection System using Wireless Intelligent Personal Communication Node," *2015 IEEE International Conference on Electro/Information Technology (EIT),* pp. 314-319, 2015.

[14] W.-J. Yi, S. Niu, T. Gonnot and J. Saniie, "System Architecture of Intelligent Personal Communication Node for Body Sensor Network," *2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC),* pp. 171-1720, 2013.

[15] ZigBee Alliance, "What is ZigBee? | zigbee alliance," 2017. [Online]. Available: http://www.zigbee.org/what-is-zigbee/.

[16] Digi, "ZigBee Wireless Standard - Digi International," 2017. [Online]. Available: https://www.digi.com/resources/standards-and-technologies/rfmodems/zigbee-wireless-standard. [Accessed 12 2 2018].

[17] Sparkfun, "Arduino Pro Mini 328 - 3.3V/8MHz - DEV-11114 - Sparkfun Electronics," 2017. [Online]. Available: https://www.sparkfun.com/products/11114.

[18] A. Elahi and A. Gschwender, ZigBee Wireless Sensor and Control Network, Prentice Hall, 2009.

[19] K. Townsend, "Adafruit 10-DOF IMU Breakout," Adafruit, 14 9 2016. [Online]. Available: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-10-dof-imu-breakout-lsm303-l3gd20-bmp180.pdf. [Accessed 10 2 2018].

[20] Analog Devices, "3-Axis Digital Accelerometer ADXL345," 2013. [Online]. Available: http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf.

[21] Raspberry Pi Foundation, "Raspberry Pi 3 Model B - Raspberry Pi," 2017. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[22] LISA Lab, "Multilayer Perceptron - DeepLearning 0.1 documentation," 9 3 2018. [Online]. Available: http://deeplearning.net/tutorial/mlp.html. [Accessed 11 3 2018].

[23] Google, "TensorFlow," 2018. [Online]. Available: https://www.tensorflow.org/. [Accessed 10 1 2018].

[24] I. Jolliffe, Principal Component Analysis, New York: Springer-Verlag, 2002.

[25] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F. Viegas and M. Wattenberg, "Embedded Project: Interactive Visualization and Interpretation of Embeddings," in *NIPS 2016 Workshop on Interpretable Machine Learning for Complex Systems*, Barcelona, Spain, 2016.