Design Flow of Wireless Body Sensor Network for Human Activity Classification using Long Short-Term Memory (LSTM) Neural Network

Saumya Gupta, Boyang Wang, Won-Jae Yi and Jafar Saniie

Embedded Computing and Signal Processing (ECASP) Research Laboratory (http://ecasp.ece.iit.edu)
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago IL, U.S.A.

***Abstract*—The design aspects of a wireless body sensor network (WBSN) consider a variety of parameters to improve the network and algorithm but the placement choice often remains unjustified by the researcher. A common application of the WBSN is for the motion-capture of a limb or human, but sensor placement differs, making each algorithm very specific to a singular movement. In this paper, we explore the system architecture and design flow of an enhanced human activity classification algorithm using long short-term memory (LSTM) neural network. A WBSN system is designed and implemented to collect information for training the LSTM sequential model. The sensors are placed on waist, ankles and wrists to observe the relevance, pertinence and accuracy improvement on increasing the number of sensor nodes and positioning of the sensors. Result shows that with the presented system and algorithm, we can precisely characterize human positions and behaviors. This WBSN system can be further extended to understand different motions and different sensor positions, and further expanded to include other sensors.**

Keywords—Wireless Body Sensor Network, LSTM, Activity Classification, Machine Learning

I. Introduction

Wireless Body Sensor Network (WBSN) devices have seen a boom in the recent decades. Advancement in semiconductor technology has allowed microelectromechanical systems (MEMS) to be more compact with lower power consumption; combined with cost-effective and long-lasting small lithium batteries. The use of motion capture system exists in health, gaming and fitness industries. There are different types of methods and algorithms present but no comparison between the outputs of these devices is known. Most algorithms and hardware designs are kept classified by large corporations and reasons for the placement of sensor unknown. In commercial products, the reason can be usability and accessibility as opposed to high reliability or accuracy. Application-specific algorithms have been developed using small datasets and few specific motions, for example, tango dancers [1] and arm movement in rowing [2]. A lack of deterministic sensor positioning is absent from their literature. Motion capture modules have been designed in [3], [4] and [5] with inertial movement units (IMU), and they all map the information to make real-time animation of the human body movement that can be used in gaming or controlling robots. For instance, a game controller wand equipped with IMU is used to capture the hand movement of the users. Research in improving performance in a game or activity is another motivation for research with motion sensors including medical Internet of Things (IoT). In the literature, a system to detect fall in patient to report to required authority is introduced in [6], aimed to help elderly in assisted living facilities prone to falling. In [1], authors explore a pair of dancers wearing sensors and performing tango, and their flare is judged based on the angle movement analysis. Thresholding has been used in [2] and [6] to classify the action. More recent publications explore the use of artificial neural networks (ANN) in developing classifying the action [7], [8]. The reason for this paradigm shift is that different users have different postures and gait which can lead to a large number of false results in thresholding. Another motivation is that previously, angular velocity and acceleration were used to calculate the angle moved at a particular joint. With the ANN, it is unnecessary to map every point of interest, but it is possible to understand common behaviors. An example of commonly used behavior is hand movements while walking, which can provide an accurate approximate value.

In this paper, we introduce a system design flow to identify and compare the classification result from different sensor positions and combinations of two types of sensors: an accelerometer and a gyroscope. Common human activities considered in this study are standing, walking, sitting, stretching and typing. Some actions have large movement (e.g., walking), some are more subtle (e.g., typing); such different movements have been chosen for us to study the limitation of the sensors and algorithm. In this paper, we illustrate the importance of analyzing the number of sensors and their location to determine sensor positioning during human activity classification. Section II describes the WBSN and data capture process. The algorithm and analysis utilized in this study are elaborated in Section III. Lastly, Section IV concludes this paper.

II. System Architecture

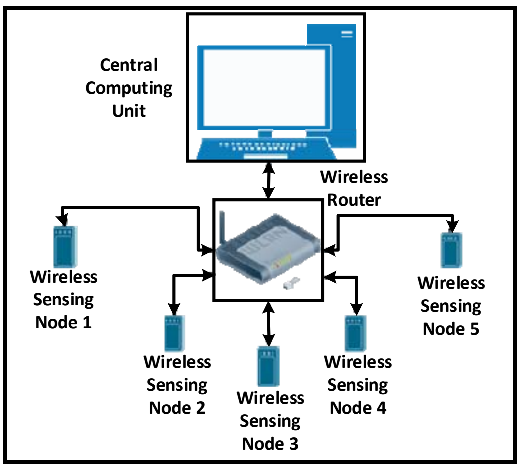


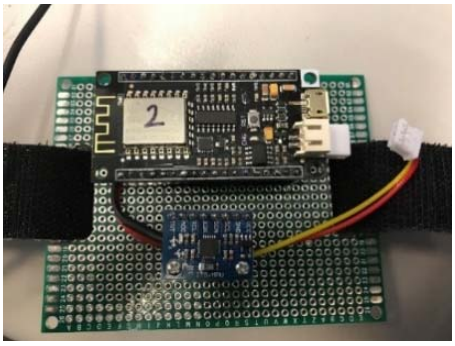
Fig. 1. Proposed WBSN System Architecture

978-1-7281-5317-9/20/$31.00 ©2020 IEEE

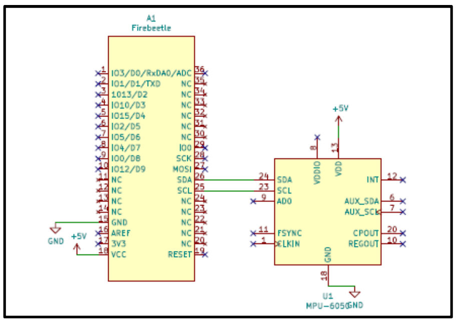
Authorized licensed use limited to: Illinois Institute of Technology. Downloaded on September 30,2020 at 21:17:10 UTC from IEEE Xplore. Restrictions apply.

As shown in Fig. 1, our WBSN system architecture consists of multiple sensing nodes, a wireless router which serves as a interconnect between all components, and a central computing unit which collects sensor data synchronously in time and analyzes them. Each sensor node in our system is equipped with an accelerometer and gyroscope coupled with a microcontroller and wireless transceiver. Any wireless protocol can be adopted to our system including classic Bluetooth, Bluetooth Low Energy, ZigBee, Z-Wave, Wi-Fi and others, by utilizing a 'universal' wireless router for protocol integration. To demonstrate the feasibility of our design, we utilize Wi-Fi connections for all wireless communications as it allowed multiple connections simultaneously.

Our WBSN system consists of 5 identical and customized wearable sensor nodes equipped with accelerometers and gyroscopes to identify movement information of the monitored user. Each sensor module has a DFRobot FireBeetle microcontroller as the main controller and MPU-6050 for capturing movements using its accelerometer and gyroscope [9] [10]. The FireBeetle has an ESP8266 microcontroller at the clock frequency of 80 MHz with integrated Wi-Fi (IEEE 802.11 b/g/n 2.4 GHz). This microcontroller can be programmed through UART using the Arduino IDE which uses a variation embedded C as the language, and provides libraries and firmware for this device. This device is powered by a 3.7 V Li-Po battery, widely used for wearable devices, and has an average operating current of 80mA. As shown in Fig. 2, we built a prototype of a wireless sensing node on a PCB board where the ESP8266 microcontroller and the MPU-6050 are connected using I^2C interface.



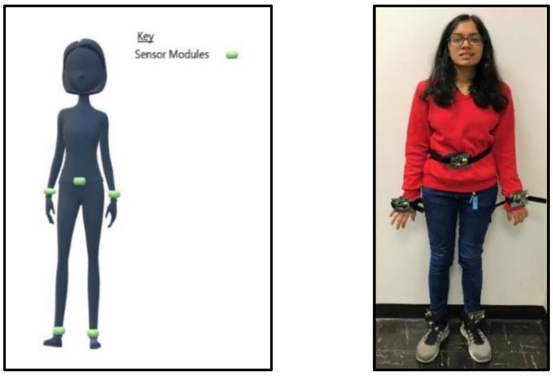
(a)



(b)

Fig. 2. Wireless Sensing Node using ESP8266 and MPU-6050; (a) Prototype, (b) Wiring Configuration

Fig. 3 shows the choice of placement of the nodes and a participant wearing 5 identical wireless sensing nodes. Sensing nodes are placed on a monitored user's wrists, ankles and waist. These locations have been chosen after surveying what are the common positioning used in the literature [1], [2], [11], [12].



(a) (b)

Fig. 3. Wireless Sensing Node Placement on a User; (a) Deployment Diagram (b) Nodes worn on a Participant

The central computing unit that collects data is running Robot Operating System (ROS) [13] on top of Ubuntu 16, a Linux operating system. ROS has low-level capability to handle messages and packets from multiple threads. It allows us to build a system that reads all the messages from the individual sensor nodes. ROS has libraries for Arduino, allowing us to communicate without defining the low-level communication protocol and arranging the message into packets. For a ROS system, an RQT Graph can be plotted by the RQT package which is used to visualize the ROS nodes, topics and the dataflow in the system. Fig. 4 shows the ROS RQT Graph of the proposed system, which visualizes the ROS computation graph of the central computing unit's application.

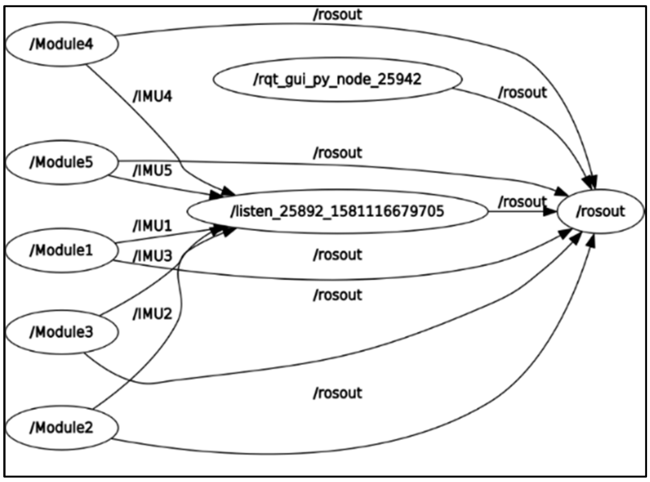


Fig. 4. ROS RQT Graph

As shown in Fig. 4, no sink node has been integrated into the design. All nodes directly publish the information to the computer. The network is designed this way to allow easier replacement of malfunctional sensor node and for better timing. A timer or counter on the controller is used to wake up the processor to publish data to the main node, this timer assures constant frequency of transmission. Using ROS, each device publishes to a unique topic. When the counter reaches

Authorized licensed use limited to: Illinois Institute of Technology. Downloaded on September 30,2020 at 21:17:10 UTC from IEEE Xplore. Restrictions apply.

to a defined constant value; the controller is waken to read the values from the IMU and load them into a ROS message packet called *imu_msg* type, which is then published to the main node. While each sensor is sending at the same frequency, it is not possible to have them start at the exact same time. The microcontrollers do not have a universal time clock but just a timer. The presence of multiple devices can lead to loss of packets or disturbances.

On the desktop computer side runs the ROS master node, it subscribes from 5 ROS topics containing *imu_msg* published by sensor nodes. The received data will be stored into a file and each received packet will be associate with a local timestamp.

There are two methods of synchronization methods: exact time synchronization and approximate time synchronization. The exact time synchronization will check the time message arrived, if all 5 arrived together then it will store them. Even if one sensor sends with a not much large delay then all the data from the four other sensor nodes; the next set is collected gets ignored. In approximate time synchronization, a few microsecond differences in measurement of packet is acceptable in the input. We use that to our advantage and group messages, keeping most of the obtained data, achieving a higher effective sampling rate than exact time synchronization. Some packets are lost, but mostly in the setup phase when all the devices aren't transmitting data. The observed packet drop in such wireless sensor networks is 10% [14]. The frequency at which we are sampling values makes a difference. Over-sampling will lead to waste of resources. Under-sampling will cause inability to identify proper features for classifications.

Total of 8 participants took part in this study. They all performed the tasks as instructed for 10 minutes each. As they were performing the actions, a ROS node was used to label the data by taking in an input from the keyboard. A ROS node is programmed to listen to all the sensor nodes and labeling node and store the data. This dataset can be further expanded to include more classification or participants. The dataset can be analyzed in a variety of methods, such as thresholding, regression or classification neural networks. We want to predict the current state of the activity from our structured dataset.

III. Long Short-Term Memory Artificial Neural Networks

In this research, we explore the use of sequential model called long short-term memory (LSTM) neural network since the dataset we've created is a function of time. There has been recent interest in using LSTM networks for motion analysis as seen in [15], [16]. The actions or classes have been chosen to be of the following categories: standing, sitting, walking, stretching and typing as shown in Fig. 5. These activities engage different parts of the body and are conducted on regular basis by all individuals. These activities were also chosen because they can be performed in an indoor environment easily and encompass different body movements.

A person walking is also standing and similarly a person typing is most likely seated. The definition of standing here has been decided to mean upright on two legs with or without movement. Correspondingly, sitting is an activity with the backside placed on a chair or a flat surface. This is further extended to other activities where a person can type whilst sitting or standing. This form of labelling allows us to do a multi-class multi-label classification, that means that multiple classes can be positive or true simultaneously. This form of data collection will more closely represent the actions of a person.

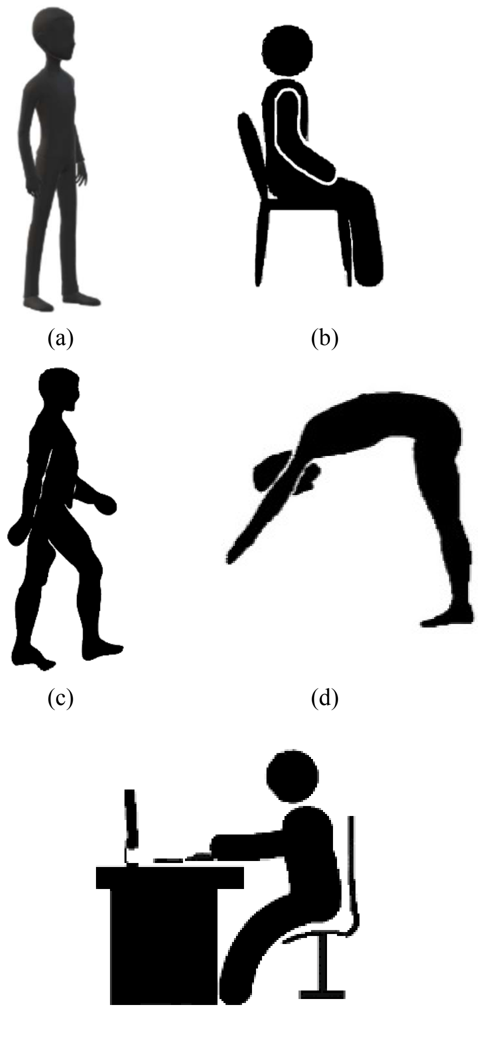


Fig. 5. Human Activities Postures; (a) Standing, (b) Sitting, (c) Walking, (d) Stretching, (e) Typing

The input data has some noise present due to sensor errors and sampling errors. The sensor uses MEMS technology which allows them to be compact, inexpensive and easily available in the industry. However, MEMs device face significant errors due to noises created by the interaction between different energies present in the devices. Environmental factors like temperature, electric and magnetic field, pressure and vibrations can trigger random output [17]. Random drifts are observed in sensor causing the output value to exceed its range of ±16384. First, we reduce the value of these random drifts' samples to the limit closest to it. Then the entire signal is put through a median filter.

Authorized licensed use limited to: Illinois Institute of Technology. Downloaded on September 30,2020 at 21:17:10 UTC from IEEE Xplore. Restrictions apply.

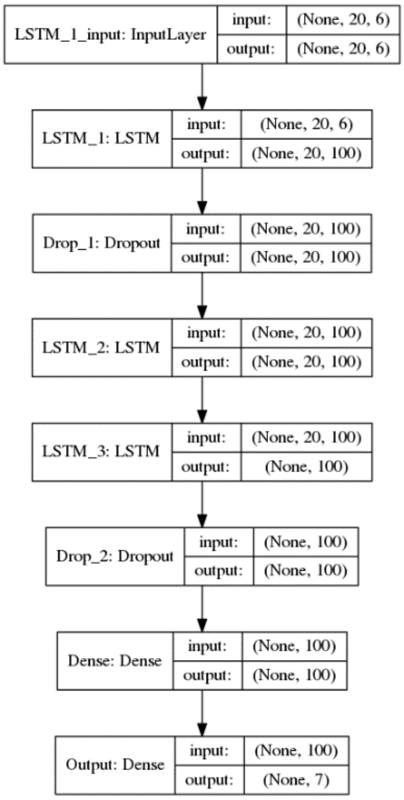


Fig 6. Keras Model Diagram

The data is divided into two parts: training and test data by the ratio of 8:2. The training data is fed to the LSTM Network. Stacked LSTM layers are used to generate complex features that are able to identify classes more accurately. Fig. 6 illustrates the model of LSTM created using Keras [14]. An increase of approximately 10% in accuracy was observed by adding one *LSTM_3* layer. Two LSTM layers have dropout activated to prevent overfitting. The last dense layer has 5 outputs corresponding to each movement.

We study the accuracy observed when the input to this model is changed. Each sensor module sends 6 values: 3 axis accelerometer and 3 axis gyroscope values. When the input is of only one sensor module, then it's corresponding 6 signals are entered into the LSTM network.

Prior to testing, we expected an increase in accuracy with the inclusion of more sensors. This would be due to an increase in data points, which would allow us to build better features for extraction. Based on Table I, we observed higher accuracy in a five sensor system compared to networks using only one or two sensors. However, this difference was small, especially compared to tests using waist sensors (*sensor E*). We also had mixed results when including the waist sensor. When adding the waist sensor to a single sensor network, the results either remained the same or increased slightly. This was different when increasing from a four sensor network as the results were two percentage points higher for the smaller network.

TABLE I. ACCURACY ACHIEVED BY DIFFERENT COMBINATION OF SENSOR NODES

Sensors[a]	Accuracy (%)					
	Stand	***Sit***	***Walk***	***Stretch***	***Type***	***Mean***
ABCDE	81.30	95.18	89.54	92.76	85.01	91.36
ABCD	89.07	97.66	82.88	98.83	92.54	93.83
A	72.78	89.65	81.81	92.34	67.01	85.71
B	78.33	90.32	90.32	86.70	91.23	90.47
C	80.08	97.72	81.97	94.29	65.99	87.96
D	80.49	92.51	83.47	99.69	69.99	88.88
E	39.54	26.43	82.91	98.66	41.01	69.28
AE	76.88	49.47	89.04	97.05	66.16	77.46
BE	82.19	85.31	80.27	93.90	98.83	90.98
CE	87.19	84.46	81.52	98.16	96.85	92.06
DE	81.14	81.02	81.58	95.18	82.32	88.30
AB	77.62	92.15	80.05	94.07	77.21	89.07
CD	89.07	97.66	82.88	98.83	92.54	88.19
AC	83.14	91.04	81.30	91.15	82.86	89.07
BD	80.28	88.23	80.02	90.48	94.41	89.97
AD	78.91	87.39	83.97	95.07	57.79	85.58
BC	82.61	89.92	77.99	91.84	93.04	90.22

a. A=Left Wrist Sensor, B= Right Wrist Sensor, C=Left Ankle Sensor, D = Right Ankle Sensor, E= Waist Sensor

IV. CONCLUSION

From this research we can conclude that, firstly, location of sensors makes a difference in the output and the selection shouldn't just be intuitive but also verified by data. Secondly, more sensors provide us more data to understand the situation, but some data can behave as noise and reduce the accuracy of the system. This form of analysis should be done for other sensors such as electromyography, pressure, humidity and others.

WBSNs will become conventional with rise in remote healthcare facilities and monitoring requirements. Currently, one-third of the population of the United States uses some form of tracking device or application to collect data about walking or cycling. A more standardized open dataset can allow for a comparison of the results of the algorithm used in the products. The dataset should have participants of different height, age, gender and race. There are differences in how people move. To optimize the algorithm, we need a large dataset encompassing lackadaisical to energetic participants. One of the reasons the artificial intelligence community is unable to cater to all its users is the lack of diversity in their datasets. The other route is to customize the product for each person allowing for their idiosyncrasies to be a part of the system and be more accurate, this will be a better method when treating patients.

REFERENCES

[1] D. K. Arvind and A. Valtazanos, "Speckled Tango Dancers: Real-Time Motion Capture of Two-Body Interactions Using On-body Wireless Sensor Networks," *2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*, 2009, pp. 312–317.

[2] J. Llosa, I. Vilajosana, X. Vilajosana, and J. M. Marquès, "Design of a Motion Detector to Monitor Rowing Performance Based on Wireless Sensor Networks," *2009 International Conference on Intelligent Networking and Collaborative Systems*, 2009, pp. 397–400.

[3] J. Wu, Z. Wang, S. Raghuraman, B. Prabhakaran, and R. Jafari, "Demonstration abstract: Upper body motion capture system using

Authorized licensed use limited to: Illinois Institute of Technology. Downloaded on September 30,2020 at 21:17:10 UTC from IEEE Xplore. Restrictions apply.

inertial sensors," *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, 2014, pp. 351–352.

[4] V. Lympourides, D. K. Arvind, and M. Parker, "Fully Wireless, Full Body 3-D Motion Capture for Improvisational Performances," SIGCHI Workshop on Whole Body Interaction, 2009, pp. 1-4.

[5] S. Huang *et al.*, "Poster abstract: Ambulatory real-time micro-sensor motion capture," *2012 ACM/IEEE 11th International Conference on Information Processing in Sensor Networks (IPSN)*, 2012, pp. 107–108.

[6] W. Yi, O. Sarkar, T. Gonnot, E. Monsef, and J. Saniie, "6LoWPAN-enabled Fall Detection and Health Monitoring System with Android Smartphone," *2016 IEEE International Conference on Electro Information Technology (EIT)*, 2016, pp. 0174–0178.

[7] P. Koch, N. Brügge, H. Phan, M. Maass and A. Mertins, "Forked Recurrent Neural Network for Hand Gesture Classification Using Inertial Measurement Data," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, 2019, pp. 2877-2881.

[8] G. M. Weiss, J. L. Timko, C. M. Gallagher, K. Yoneda, and A. J. Schreiber, "Smartwatch-based activity recognition: A machine learning approach," *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, 2016, pp. 426–429.

[9] "FireBeetle ESP8266 IOT Microcontroller." [Online]. Available: https://wiki.dfrobot.com/FireBeetle_ESP8266_IOT_Microcontroller_SKU__DFR0489#target_0. [Accessed: 05-Nov-2019].

[10] "MPU-6000/MPU-6050 Product Specification." InvenSense, Aug-2013.

[11] Y.-C. Kan and C.-K. Chen, "A Wearable Inertial Sensor Node for Body Motion Analysis," *IEEE Sensors Journal*, vol. 12, no. 3, pp. 651–657, Mar. 2012.

[12] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, 2006, pp. 1-4.

[13] J. M. O'Kane, *A gentle introduction to ROS*. Leipzig: Amazon, 2014.

[14] "Visualization - Keras Documentation." [Online]. Available: https://keras.io/visualization/. [Accessed: 15-Feb-2020].

[15] C. R. Naguri and R. C. Bunescu, "Recognition of Dynamic Hand Gestures from 3D Motion Data Using LSTM and CNN Architectures," *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 1130–1133.

[16] Z. Ni, W. Qian, Y. Gu, Z. Chen, Z. Zhao, and T. Zhang, "A Method of Motion Information Overall Sensing of Wearable Robot Based on Virtual Inertial Measurement Unit," *2019 4th International Conference on Control and Robotics Engineering (ICCRE)*, 2019, pp. 147–150.

[17] R. Bhardwaj, N. Kumar, and V. Kumar, "Errors in micro-electro-mechanical systems inertial measurement and a review on present practices of error modelling," *Transactions of the Institute of Measurement and Control*, vol. 40, no. 9, pp. 2843–2854, Jun. 2018.

Authorized licensed use limited to: Illinois Institute of Technology. Downloaded on September 30,2020 at 21:17:10 UTC from IEEE Xplore. Restrictions apply.