

Fall Detection Using Neural Network Based on Internet of Things Streaming Data



Zana Azeez Kakarash^{1,2}, Sarkhel H. Taher Karim^{3,4}, Mokhtar Mohammadi⁵

¹Department of Engineering, Faculty of Engineering and Computer Science, Qaiwan International University, Sulaymaniyah, Iraq, ²Department of Computer Engineering and Information Technology, Faculty of Engineering, Razi University, Kermanshah, Iran, ³Department of Computer Science, College of Science, University of Halabja, Halabja, Iraq, ⁴Department of Computer Network, Technical College of Informatics, Sulaimani Polytechnic University, Sulaymaniyah, Iraq, ⁵Department of Information Technology, Lebanese French University, Erbil, Kurdistan Region, Iraq

ABSTRACT

Fall event has become a critical health problem among elderly people. We propose a fall detection system that analyzes real-time streaming data from the Internet of Things (IoT) to detect irregular patterns related to fall. We train a deep neural network model using accelerometer data from an online physical activity monitoring dataset named, MobiAct. An IBM Cloud-based IoT data processing framework is used to manage streaming data. About 96.71% of accuracy is achieved in assessing the performance of the proposed model.

Index Terms: Fall Detection, Internet of Things, Artificial Neural Networks, Machine Learning

1. INTRODUCTION

These days, we can see a quick increment in people using wearable devices for more different purposes and reasons in their lives. According to the previous studies, the employment of connected wearable detector devices is expected to extend from 325 million in 2016 to 1105 million in 2022 [1]. In this way, where an unmeasurable number of wearable devices is connected together will produce a huge amount of information consistently, at that time, we face or interaction with a critical test of storing, handling, and processing that information's to produce usable information and to make a keen world. Distinguishing regularities and abnormalities or inconsistencies in gushing information from that measure of usable information, we previously produced subsequent

to putting away and preparing it can give us an encounter and can possibly give experiences, and is useful in human services, money, security, web-based social networking, and numerous applications [2], [3].

Over the most recent few years, totally different types of methodologies and techniques have been proposed for identifying regularities and irregularities pattern in streaming data for fall detection, which may be a wearable tool based, ambiance sensor-based, and vision-based [4]. Above all, wearable devices typically take some different benefit of embedded sensors to observe the movement and placement of the body, such as measuring system, accelerometer, magnetometer, and gyroscope [5], [6]. As well as the value of wearable tool based any methodologies or techniques are very low, also because the installation and operation are not complex, and the task is not difficult for the elderly [7], [8]. If from now the physician could not specifically monitoring and identify falls, then there is no chance to save or protect you in future to preventing an accident. For such reasons, fall identification and anticipation have turned into a significant issue then must be find and propose a good way to solve

Access this article online

DOI: 10.21928/uhdjst.v4n2y2020.pp91-98

E-ISSN: 2521-4217

P-ISSN: 2521-4209

Copyright © 2020 Kakarash, *et al.* This is an open access article distributed under the Creative Commons Attribution Non-Commercial No Derivatives License 4.0 (CC BY-NC-ND 4.0)

Corresponding author's e-mail: Zana Azeez Kakarash, Department of Software Engineering Faculty of Engineering and Computer Science, Qaiwan International University Sulaimani, Iraq. E-mail: zana.azeez@kti.edu.krd

Received: 10-03-2020

Accepted: 28-09-2020

Published: 25-10-2020

in the method of detecting regularities and irregularities in streaming data for fall detection [9], [10].

Fall in the human daily life is one of the main health risks and dangerous, mostly for the older community in our today's society, because of the rose growing in mortality, morbidity, incapacity, disability, and frailty [11]. Nearly for fall initiate injuries as collected and represented, over 80% of all damage related clinic confirmations among peoples for more than 65 years [12], [13]. According to the reasons as we mentioned before, falls affect a huge number of the elderly all through the world. For instance, falls some of the aged cost the National Health Service more than £4.6 million every day as indicated by a report by the Centre for Social Justice UK [14].

The notable studies and researches in this filed are detecting fall or anomaly in real streaming data [15]-[20] and outliers [21], [22]. We will probably investigate the circumstance of unpredictable human development identification, for example, fall, by utilizing continuous sensor information. At that point, we map that issue as sporadic example location issue, thinking about a fall as a capricious action with respect to standard human action and attempt to perceive the exceptional fall situations from ordinary development or human activities, for example, walking, sitting, lying (LYT), sleeping, standing, and every other movement such as playing, cooking, and running.

In general, Artificial Neural Networks (ANN) have systematically achieved higher results in the detection falls from physical activity observation knowledge. Ozdemir and Barshan have used a pair of 2520 trials to make a huge amount of dataset [23]. Their fall detection system achieved 95% accuracy by employing a multi-layer perceptron (MLP) for binary classification between activities of daily living (ADL) and fall. Kerdegari *et al.* recorded 1000 movement acceleration data collection using a waist-worn measuring system and obtained 91.6% accuracy for binary classification for ADL against fall using MLP [24]. Nukala *et al.* collected knowledge from 322 tests achieved 98.7% accuracy with MLP victimization scaled conjugate graduate learning [25].

Theodoridis *et al.* [26] developed two long short-term memory (LSTM) models, one with easy measuring system data collection associated another with accelerometer data revolved at an angle, using a published dataset referred to as UR Fall Detection. The LSTM model with rotation obtained the most effective results with 98.57% accuracy.

The work by class Musci *et al.* used one public dataset (SisFall) dataset implementing Recurrent Neural Network with

underlying LSTM blocks for developing online fall detection system [27]. They achieved 97.16% accuracy victimization associate best window breadth of $w = 256$. Ajerla *et al.* [42] slightly changed the preprocessing done by Vavoulas *et al.* on other public dataset (MobiAct) dataset and achieved quite 90% accuracies on most of their experiments victimization MLP and LSTM [28], [29]. They also obtained 99% accuracy victimization LSTM in two of their experiments. Table 1 shows the compares performance of the ANN techniques.

Nowadays, there are a lot of wearable sensors devices exist which may observe falls automatically and send a notification to the caregivers, machine services, or ambulance offerings. However, most of them are expensive or need a subscription of month-to-month service. A large number of studies and research has been done on fall detection the usage of sensors such as accelerometers and gyroscopes, because of low cost and incorporated into a large number of cell phones accessible today's in the world such as smartphones and tablets. Rather than fall identification, there is a dire need for prediction and prevention systems [30], [31].

In this paper, we propose a system for detecting abnormal pattern of falling behavior in real-time streaming data in internet of things (IoT). The data were obtained from wearable sensor systems used for human health and activity tracking and control. In our research, we focus on how to train the proposed system to recognize and distinguish irregular activity pattern related to specific kinds of fall according to three different annotated published datasets: MobiAct, using Vavoulas *et al.* in 2016 [32], SisFall, using Sucerquia *et al.* in 2017 [33], and UMAFALL by way of Eduardo *et al.* in 2018 [34] for implementing our framework from streaming data for fall detection, we used an ANN model for fall recognition giving (96.71%) accuracy. At that point, we integrate our system for that huge amount of data recordings from streaming sensor (online) into free IBM Streams to fabricate an IBM Cloud-based IoT information preparing structure.

TABLE 1: Comparison of the ANN techniques

Research paper	Technique	Accuracy
Ozdemir <i>et al.</i> , 2014 [35]	MLP	95%
Kerdegari <i>et al.</i> , 2013 [36]	MLP	91.6%
Nukala <i>et al.</i> , 2014 [37]	MLP	98.7%
Theodoridis <i>et al.</i> , 2018 [38]	LSTM	98.57%
Abbate <i>et al.</i> , 2012 [39]	Feed-forward ANN	100%
Musci <i>et al.</i> , 2018 [40]	RNN with LSTM	97.16%
Ajerla <i>et al.</i> , 2018 [41]	MLP and LSTM	99% (LSTM)

MLP: Multi-layer perceptron, RNN: Recurrent Neural Network, LSTM: Long short-term memory, ANN: Artificial Neural Networks

2. METHODOLOGY

We proposed a profound learning model to recognize fall and in this manner, another structure for utilizing the model with steaming information from wearable sensor frameworks is proposed. Simulated fall detection is used due to the scarcity of actual fall data, as fall is a dangerous event. We consider a framework that includes ANN as one of the evaluating procedure.

ANN has ceaselessly achieved better outcomes in fall recognition from physical checking spilling information. The architecture comprises three essential layers (I) gathering stream for information ingestion, (ii) streaming extract, transform, and load engine for real-time query processing, and (iii) Online Analytical Processing backend for taking care of long-running inquiries. The means of the proposed framework are as per the following data collection

2.1. Dataset

All open datasets or we can say public data sets for wearable fall detection frameworks are distinguished that the selection criteria for any datasets should give priority to the experimental subjects, the quantity of tests and sorts of ADL and falls included in the study [42].

Based on that, we have chosen three different datasets, MobiAct, SisFall, and UMAFall, to evaluate our framework dependent on the number of subjects and number of exercises secured by each dataset.

It is possible to collect data from one or several sources. For instance, in an IoT workload, data are ingested simultaneously from thousands of separate data sources. Each source submits fresh tuples to a stream (likely to send them through a socket), which a data collector mechanism then receives. This data collector mainly act as a queue for messaging [43].

As we explained in Table 2, MobiAct dataset contains labeled data for four differing kinds of falls and nine completely different ADLs were collected from 70 subjects (Male/Female) and quite a pair of 2500 trials buy using so many different kind of a smartphone. The activities are depicted employing a time stamp, raw measuring system values, raw rotating mechanism values, and orientation information. Table 3 shows the activities covered within the MobiAct dataset. In the MobiAct dataset is that it does not embrace fall data from any old people [44]. SisFall remedies this situation by mixing with information from 15 old people aged between 60 and 75 years. Still, in these two datasets real fall data it does not available or do not included.

SisFall contains annotated data for 15 differing kinds of falls and 19 different types of ADLs were collected from 38 subjects and quite 4500 trials using a custom measuring instrument containing two completely different models of 3D accelerometers and a rotating mechanism positioned on a buckle. Furthermore, UMAFall datasets include 12 different types of fall and 15 types of ADLs were collected from 2600 trials by using so many kinds of smartphone and remote accelerators. Like first dataset as I mention above these activities are also depicted employing a time stamp, raw measuring system values, raw rotating mechanism values, and orientation information

2.2. Data Preprocessing

2.2.1. Segmentation

Using the same procedure given by Aziz *et al.* [45], the raw data are segmented into 200 blocks and then the feature sets for each block are generated.

2.2.2. Feature extraction

List of features A: A sum of 54 features were created in include setA [46]. For every pivot (x, y, and z) of the quickening, 21 highlights were determined from the mean, middle, Standard Deviation (STD), slant, kurtosis, least, and most extreme. Utilizing the outright estimations of every hub (x, y, and z) of the quickening, another 21 highlights were determined from the mean, middle, STD, slant, kurtosis, least, and most extreme. Incline, another component, was determined utilizing Eq. 1, one for the given tomahawks esteems (x, y, and z) and another for the supreme qualities ($|x|$, $|y|$, $|z|$).

TABLE 2: Activites covered in datasets

Code	Activity	Description
FALL		
FOL	Forward-lying	Fall forward from standing, use of hands to dampen fall
FKL	Front-knees-lying	Fall forward from standing, first impact on knees
SDL	Sideward-lying	Fall sideward from standing, bending legs
BSC	Back-sitting-chair	Fall backward while trying to sit on a chair
ADL		
STD	Standing	Standing with subtle movements
WAL	Walking	Normal walking
JOG	Jogging	Jogging
JUM	Jumping	Continuous jumping
STU	Stairs up	10 stairs up
STN	Stairs down	10 stairs down
SCH	Sit chair	Sitting on a chair
SCI	Car step in	Step in a car
SCO	Car step out	Step out of a car

TABLE 3: Datasets used for training

Dataset	No. of subjects (male/female)	Age range	No. of activities ADLs/falls	No. of samples (ADLs/falls)
MobiAct [32]	70 (45/25)	20–47	9/4	2526 (1879/647)
SisFall [33]	60 (30/20)	19–30, 60–75	19/15	4505 (2707/1798)
UMAFall [34]	55 (35/15)	30–47	15/12	2650 (1700/950)

ADL: Activities of daily living

$$\text{Slope} = \sqrt{\frac{(\max_x - \min_x)^2 + (\max_y - \min_y)^2}{(\max_z - \min_z)^2}} \quad (1)$$

$$X = \frac{x - \text{Min}(\text{feature})}{\text{Max}(\text{feature}) - \text{Min}(\text{feature})} \quad (5)$$

Four different features were determined utilizing mean, STD, slant, and kurtosis of the tilt edge (TAi) between the gravitational vector and the Y-axis utilizing Eq. 2.

$$TAi = \sin^{-1}(y_i / (\sqrt{x_i^2 + y_i^2 + z_i^2})) \quad (2)$$

Where i denotes the sequence of the sample.

Utilizing the magnitude of the acceleration vector, six features were resolved from the mean, STD, least, greatest, contrast among most extreme and least, and zero-intersection rate. The size was determined utilizing Eq. 3.

$$\text{Magnitude} = \sqrt{x_i^2 + y_i^2 + z_i^2} \quad (3)$$

Where i denotes the sequence of samples.

For each of the three axes (x, y, and z), the average absolute difference was calculated [47]. Furthermore, the average resultant acceleration of all the three axes was generated using Eq. 4.

$$\text{Average resultant acceleration} = \left(\frac{1}{n}\right) * \sum_i \sqrt{x_i^2 + y_i^2 + z_i^2} \quad (4)$$

Where i denotes the sequence of samples.

Combined Feature Set. The feature sets A, B, and C are merged to generate the dataset with the combined features. We had 7670 samples in the dataset. After feature extraction, each sample had one of six classification values and 58 extracted features. Class values include four kinds of falls and two kinds of non-falls as defined in the MobiAct dataset. The two non-fall classes denote Standing (STD) and LYI positions.

2.2.3. Normalization

The extracted features were normalized using Matlab R2017b with the min-max scaling formula given in Eq. 5.

2.2.4. Data balancing

The obtained dataset after normalization was very unbalanced containing 5830 non-fall and 1840 fall data. We made data balanced for our dataset using Matlab R2017b as follows: The various fall data categories were merged into a single fall data classification while the two non-fall data categories were merged into a single non-fall data classification. Then, the fall data were oversampled to create 2000 samples and the non-fall data were under-sampled to create 2000 samples to create a combined balanced dataset containing 4000 samples.

2.2.5. Feature selection

Selection of discriminable features affects the performance of the classification in terms of accuracy and complexity. Among different feature selection methods, the relief-F method is used for feature selection, the relief-F method in python is used with number of neighbor set to two and number of features to be kept set to four. The relief-F compared to the relief method is more robust and can deal with incomplete and noisy data [48].

2.3. Predictive Modeling

We created a simple predictive Deep Neural Network (DNN) model consisting of five layers including three hidden layers to detect fall based on data stream sensor.

The structure of DNN is displayed in Fig. 1. The model was constructed and designed using offline datasets and then was implemented in our streaming data processing system.

We are building a streaming data processing system with IBM tools, and then we tested and validated our prediction model with a static fall data.

2.4. Streaming Data Processing Framework

After finalizing, testing and approving our proposed model using specific static data for fall detection, we build up a streaming data processing and handling structure with IBM devices. The framework structure is appeared in Fig. 2.

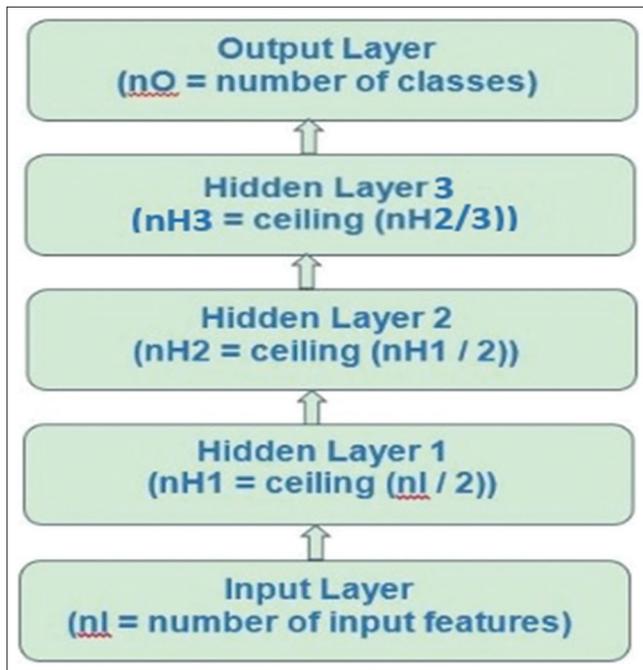


Fig. 1. Deep learning model utilized for the proposed framework.

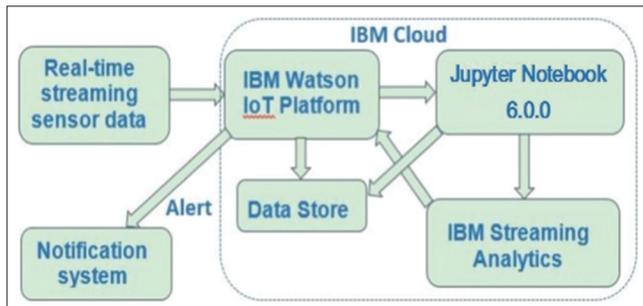


Fig. 2. The framework of the data stream processing system to detect irregular patterns for internet of things streaming data.

2.5. Architecture and Components

- Data source: Sensor information is recovered from the cell phone or any wearable devices must be fixed in the patient body.
- Data ingestion system: To receiving the sensor data from a smartphone or a wearable sensor and go about as the Message Queuing Telemetry Transport [49] message broker, we used IBM Watson IoT Platform. And also Apache Kafka [50] is used here as an open source alternative, which uses depends on its own protocol.
- Data stream processing: To start with we utilized Jupyter Notebook with IBM Cloud [51] as appeared in Fig. 3, for offline information preparing to run the element extraction and to execute the AI model utilizing the sample data. The Python code utilized in Jupyter

Notebook and the middle information records have been posted on Github. We had an arrangement to utilize the IBM Streaming Analytics [52], an organization for IBM Streams on IBM cloud, to continually screen the sensor information from mobile phone or wearable sensors and send an admonition to the watching application to alert human services suppliers about the crisis care required by patients in the unexpectedly occasion or some other of a fall.

- Data store: We have the plan to use IBM Cloud [53] as the NoSQL database to store the sensor information from the Watson IoT platform and classification of that streaming data which are obtained from Watson Studio.
 - Data sink: The framework will have different output channels:
 - Knowledge base.
 - Visualization
 - Monitor
 - IBM cloud.
1. Implementation: The tests were run on a framework with the accompanying least equipment and programming prerequisites:
 - Platform: Keras with tensor flow backend
 - CPU: Intel Core i5-2467M, 1.6 GHz
 - RAM: 10 GB
 - Hard disk: 128 GB
 - OS: Windows 10 64 bit .

To actualize the framework, we utilized a free IBM cloud account and the relevant services. We included the following accompanying assets.

- Jupyter notebook 6.0.0
- Cloud object storage
- Streaming analytics
- IoT platform.

3. RESULTS AND DISCUSSION

The machine learning model for the complete feature set was first tried on MobiAct as explained in Section 2.1. This resulted in our model in an accuracy of (96.71%) for binary classification, fall and ADL (non-fall) using Eq 6. The model was left running for 200 epochs.

$$ACC = \frac{(\text{True positive} + \text{True negative})}{\text{Number of all samples}} \quad (6)$$

The normalized and un-normalized confusion matrices were implemented for classifying fall and non-fall (ADL) cases out of 7670 samples are appeared in Fig. 4. This shows the

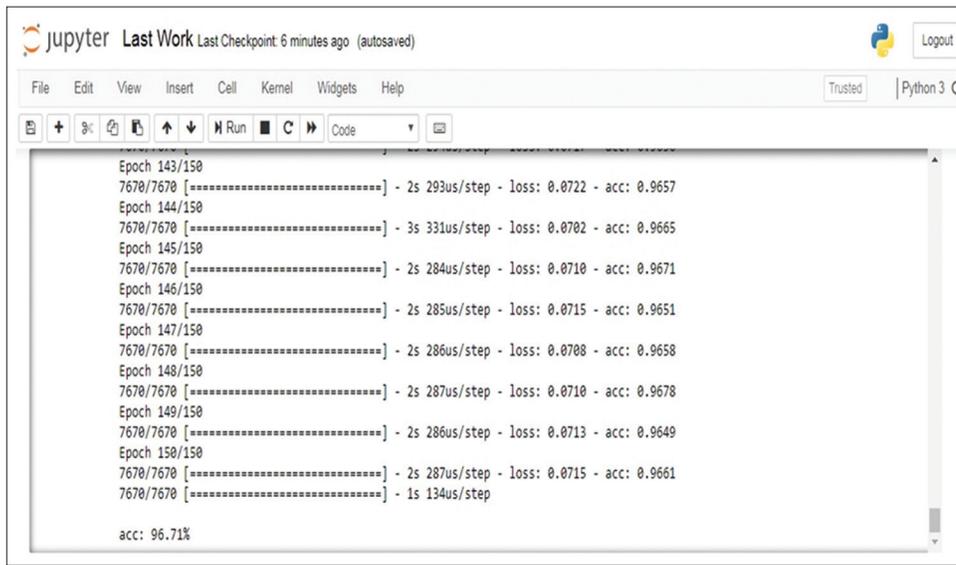


Fig. 3. Machine learning model executed using Jupyter Notebook.

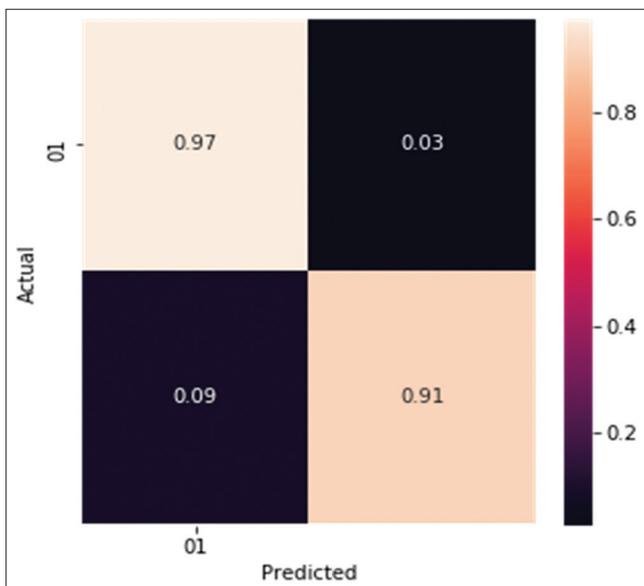


Fig. 4. Normalized confusion matrix for the trained model.

created model performs incredible and can be utilized to develop the proposed framework.

4. CONCLUSION

In this paper, we addressed the problem of irregularity pattern detection from online streaming data. Here, we especially focused on detecting fall as an irregular human activity which is common amongst elderly people. We implemented a DNN model that can classify fall from non-

fall activities based on the dataset MobiAct and two more datasets with an accuracy of 96.71%. For future, we plan to compare the proposed system with other systems that use open source streaming data analytics tools to evaluate the functionality and performance of the IBM tools used in our framework.

REFERENCES

- [1] H. Tankovska. "Statistic", 2020. Available from: <https://www.statista.com/statistics/487291/global-connected-wearable-devices>.
- [2] Mahfuz. "Detecting Irregular Patterns in IoT Streaming Data for Fall Detection". 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2018.
- [3] E. Bahmani, M. Jamshidi and A. Shaltoolki. "Breast cancer prediction using a hybrid data mining model". *International Journal on Informatics Visualization*, vol. 3, no. 4, pp. 327-331, 2019.
- [4] J. Liu. "Development and evaluation of a prior-to-impact fall event detection algorithm". *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 7, pp. 2135-2140, 2014.
- [5] P. Pierleoni. "A high reliability wearable device for elderly fall detection". *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4544-4553, 2015.
- [6] R. Freitas. "Wearable Sensor Networks Supported by Mobile Devices for Fall Detection". *SENSORS*, IEEE, 2014.
- [7] X. Zhuang. "Acoustic fall detection Using Gaussian Mixture Models and Gmm Super Vectors". 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 2009.
- [8] Y. Li. "Efficient source separation algorithms for acoustic fall detection using a microsoft kinect". *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 3, pp. 745-755, 2014.
- [9] R. Hamedanizad, E. Bahmani, M. Jamshidi and A. M. Darwesh. "Employing data mining techniques for predicting opioid withdrawal

- in applicants of health centers". *Journal of Science and Technology*, vol. 3, no. 2, pp. 33-40, 2019.
- [10] A. Shaltoolki and M. Jamshidi. "The use of data mining techniques in predicting the noise emitted by the trailing edge of aerodynamic objects". *International Journal on Informatics Visualization*, vol. 3Z, no. 4, pp. 388-393, 2019.
- [11] A. S. Cook. "Falls in the Medicare population: incidence, associated factors, and impact on health care". *Physical Therapy*, vol. 2009, no. 2, pp. 324-332, 2009.
- [12] P. Kannus. "Fall-induced injuries and deaths among older adults". *JAMA*, vol. 281, no. 20, pp. 1895-1899, 1999.
- [13] Y. Cheng. "A Fall Detection System Based on SensorTag and Windows 10 IoT Core". 15th International Conference on Mechanical Science and Engineering, pp. 238-244, 2015.
- [14] O. Ojetola, E. I. Gaura and J. Brusey. "Fall Detection with Wearable Sensors Safe (Smart Fall Detection)". Intelligent Environments, 2011 7th International Conference on, pp. 318-321, 2011.
- [15] S. M. S. Forbes. "Fall prediction using behavioural modelling from sensor data insmarthomes". *Artificial Intelligence Review*, vol. 53, pp. 1071-1091, 2019.
- [16] S. K. Gharghan. "Accurate fall detection and localization for elderly people based on neural network and energy-efficient wireless sensor network". *Energies*, vol. 11, pp. 1-32, 2018.
- [17] D. Yacchirema. "Fall detection system for elderly people using IoT and big data". *Procedia Computer Science*, vol. 130, pp. 603-610, 2018.
- [18] C. C. H. Hsu. "FallCare+: An IoT Surveillance System for Fall Detection". Proceedings of the 2017 IEEE International Conference on Applied System Innovation IEEE-ICASI 2017 Meen, pp. 921-922, 2017.
- [19] Ahmad S. "Real-Time Anomaly Detection for Streaming Analytics". arXiv, 2016.
- [20] S. C. Tan. "Fast Anomaly Detection for Streaming Data". Proceedings of the 22nd International Joint Conference on Artificial Intelligence, pp. 1511-1516, 2011.
- [21] M. Gupta. "Outlier detection for temporal data: A survey". *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250-2267, 2014.
- [22] S. Sadik and L. Gruenwald. "Research Issues in Outlier Detection for Data Streams". Vol. 15. SIGKDD Explorations, pp. 33-40, 2013.
- [23] T. Özdemir and B. Barshan. "Detecting falls with wearable sensors using machine learning techniques". *Sensors*, vol. 14, no. 6, pp. 10691-10708, 2014.
- [24] H. Kerdegari, K. Samsudin, A. R. Ramli and S. Mokaram. "Development of wearable human fall detection system using multilayer perceptron neural network". *International Journal of Computational Intelligence Systems*, vol. 6, no. 1, pp. 127-136, 2013.
- [25] T. Nukala, N. Shibuya, A. Rodriguez and J. Tsay. "An efficient and robust fall detection system using wireless gait analysis sensor with artificial neural network (ANN) and support vector machine (SVM) algorithms". *Open Journal of Applied Biosensor*, vol. 3, pp. 29-39, 2014.
- [26] T. Theodoridis, V. Solachidis, N. Vretos and P. Daras. "Human fall detection from acceleration measurements using a recurrent neural network". In: *Precision Medicine Powered by pHealth and Connected Health*. Springer, Berlin, Germany, pp. 145-149, 2018.
- [27] M. Musci, D. De Martini, N. Blago, T. Facchinetti and M. Piastra. "Online Fall Detection using Recurrent Neural Networks". arXiv, 2018.
- [28] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Padiaditis and M. Tsiknakis. "The MobiAct Dataset: Recognition of Activities of Daily Living Using Smartphones". ICT4AgeingWell, pp. 143-151, 2016.
- [29] D. Dharmitha, M. Sazia and Z. Farhana. "Fall Detection from Physical Activity Monitoring Data". Presented at the International SIGKDD workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications (BigMine), London, 2018.
- [30] D. Ajerla, S. Mahfuz and F. H. Zulkernine. "A Real-Time Patient Monitoring Framework for Fall Detection". Queen's University, Kingston, Ontario, Canada, 2018.
- [31] D. Ajerla. "Fall Detection from Physical Activity Monitoring Data". BIGMINE, London, UK, 2018.
- [32] T. Xu, Y. Zhou and J. Zhu. "New advances and challenges of fall detection systems: A survey". *Applied Science*, vol. 8, no. 3, p. 418, 2018.
- [33] H. Chen, (Eds.). "Ngu3Fall Detection Using Smartwatch Sensor Data with Accessor Architecture". Springer International Publishing, Berlin, Germany, pp. 81-93, 2017.
- [34] E. C. J. Santoyo-Ramón. "UMAFall: Fall Detection Dataset (Universidad deMalaga)". Available from: https://www.figshare.com/articles/UMA_ADL_FALL_Dataset_zip/4214283. [Last accessed on 2018 Jun 04].
- [35] S. Ahmad, A. Lavin, S. Purdy and Z. Agha. "Unsupervised real-time anomaly detection for streaming data". *Neurocomputing*, vol. 262, pp. 134-147, 2017.
- [36] M. Ahmed, A. N. Mahmood and M. R. Islam. "A survey of anomaly detection techniques in financial domain". *Future Generation Computer Systems*, vol. 55, pp. 278-288, 2016.
- [37] S. Ahmad and S. Purdy. "Real-time Anomaly Detection for Streaming Analytics". arXiv, 2016.
- [38] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep Learning for IoT Big Data and Streaming Analytics: A Survey," *IEEE Communications Surveys & Tutorials*, 2018.
- [39] J. Meehan, C. Aslantas, S. Zdonik, N. Tatbul and J. Du. "Data Ingestion for the Connected World". Classless Inter-Domain Routing, 2017.
- [40] S. Sadik and L. Gruenwald. "Research issues in outlier detection for data streams". *ACM SIGKDD Explorations Newsletter*, vol. 15, no. 1, pp. 33-40, 2014.
- [41] M. Gupta, J. Gao, C. C. Aggarwal and J. Han. "Outlier detection for temporal data: A survey". *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250-2267, 2014.
- [42] E. Casilari, J. A. Santoyo-Ramón and J. M. Cano-García. "Analysis of public datasets for wearable fall detection systems". *Sensors*, vol. 17, no. 7, p. 1513, 2017.
- [43] J. Meehan. "Data Ingestion for the Connected World". Creative Commons Attribution License, 2017.
- [44] M. P. G. Vavoulas. "The mobiFall dataset: Fall detection and classification with a smartphone". *International Journal of Monitoring and Surveillance Technologies Research*, vol. 2, no. 1, p. 13, 2014.
- [45] O. Aziz, M. Musngi, E. J. Park, G. Mori and S. N. Robinovitch. "A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials". *International Federation for Medical and Biological Engineering*, vol. 55, no. 1, pp. 45-55, 2017.

- [46] G. Vavoulas¹. "The MobiAct Dataset: Recognition of Activities of Daily Living using Smartphones". In" International Conference on Information and Communication Technologies for Ageing Well and e-Health, 2016.
- [47] J. R. Kwapisz, G. M. Weiss and S. A. Moore. "Activity recognition using cell phone accelerometers". *SIGKDD Explorations*, vol. 12, no. 2, pp. 74-82, 2010.
- [48] Y. Z. Z. Wang. "Application of Relieff Algorithm to Selecting Feature Sets for Classification of High Resolution Remote Sensing Image". 2016 IEEE International Geoscience and Remote Sensing Symposium, pp. 755-758, 2016.
- [49] MQTT. Available from: <https://www.mqtt.org/getting-started>. [Last accessed on 2020 Jan 01].
- [50] A. Kafka. Available from: <https://www.kafka.apache.org/intro>. [Last accessed on 2020 Jan 05].
- [51] IBM. "IBM Watson Studio IBM Watson and Cloud Platform Learning Center", 2020. Available from: <https://www.developer.ibm.com/technologies/data-science>.
- [52] IBM. "IBM Streaming Analytics IBM Watson and Cloud Platform Learning Center". 2016-07-18 2016
- [53] IBM. "IBM Cloudant-IBM Watson and Cloud Platform Learning Center", 2020. Available from: <https://www.developer.ibm.com/components/cloud-ibm>.