Sleep Apnea Detection in Fog Based Ambient Assisted Living System

Ace Dimitrievski[a], Natasa Koceska[b], Eftim Zdravevski[a], Petre Lameski[a], Betim Cico[c], Saso Koceski[b] and Vladimir Trajkovik[a]

[a] *Faculty of Computer Science and Engineering, Ss.Cyril and Methodius University, Skopje, N. Macedonia*
[b] *Faculty of Computer Science, University Goce Delcev Stip, N.Macedonia*
[c] *MetropolitanTirana University, Tirana, Albania*

Abstract

Ambient Assisted Living environments use different sensors and actuators to enable their end-users to live in their preferred environments. Unlike smart homes, where a target audience is usually a family unit, standard Ambient Assisted Living end users are care receivers and care providers. This article describes an approach based on the fog computing paradigm to detect sleep apnea in an Ambient Assisted Living context unobtrusively. The edge nodes process and detect local activities of daily living events and have direct control of the local environment. The fog nodes are used to further process and transmit data. The cloud is used for more complex and anonymous data computation. This research shows that sensors, which are unobtrusive and do not interfere with users' daily routines, can be successfully used for pattern observation.

Keywords

Ambient Assisted Living (AAL), Fog computing, Cloud computing, Personal health care.

1. Introduction

Advancements in cloud computing and the Internet of Things (IoT) have had a positive impact on pervasive computing and can improve Ambient Assisted Living (AAL) solutions. Fog computing is a newer discipline that brings an opportunity to fill in some gaps and improve many aspects of cloud-based AAL systems, mainly by increasing user privacy if used correctly [1], [2].

Technology for monitoring, assisting, and improving personal health has improved considerably with affordable wearable and unobtrusive sensors, cloud computing, and improved Internet connectivity. The presence and rapid growth of the Internet of Things (IoT) paradigm has also impacted how people monitor their health. Most current wearable devices can monitor heart rate and physical activity. More appliances come with Internet connection capability, and smart sensors are becoming increasingly common. The data obtained with unobtrusive sensing can give a more detailed picture of the care receivers' health and personal habits. In that way, technology directly impacts elderly and disabled people’s ability to remain at home and live more independent lives [3][4]. AAL is also addressing the growing cost of traditional health care. Advances in AAL's research provide tools and methods for improving the health of the elderly and people with disabilities. On the other hand, Enhanced Living Environment (ELE) is a field that provides resources for personal health for the general population. Although AAL and ELE address different target audiences, both fields benefit from similar technology [5].

A typical AAL goal is to enable care providers to have technology-enabled continuous monitoring of care receivers. It reduces care costs on the one hand and increases care efficiency on the other hand [6][7]. Cloud paradigm fits well for this scenario as data can be aggregated and analyzed in a centralized location. An interface for care

Proceedings of RTA-CSIT 2021, May 2021, Tirana, Albania
EMAIL: ace.dimitrievski@gmail.com (A. 1);
natasa.koceska@ugd.edu.mk (A. 2);
eftim.zdravevski@finki.ukim.mk (A. 3)
petre.lameski@finki.ukim.mk (A. 4) bcico@umt.edu.al (A. 5)
saso.koceski@ugd.edu.mk (A. 6) trvlado@finki.ukim.mk (A. 7

providers can be provided from the cloud using the web and mobile devices. Network reliability and demand for real-time processing of risk factors and different privacy concerns require some local data processing. Fog computing addresses these problems by its definition.

A single device, individual, or group of unobtrusive sensors present in the ELE can provide input on a limited set of health aspects. Smartwatches and health trackers can track body temperature, heart rate, walking or running; environmental sensors can detect temperature, humidity, fall detection, and movement within the home. A more holistic picture of these devices and sensors can be provided if connected to the cloud, where all the data is analyzed for more robust data processing techniques. By using data from many users, machine learning (ML) algorithms can learn and predict health hazards and find correlations between the environment and human health [8][9].

While IoT cloud-based computing benefits are visible both in research and daily use, there are many drawbacks when it comes to personal health care data clouds. The most significant ones are the following:

- The lack of security of IoT devices and companies' un-proper practices that gather and abuse personal data have made consumers more proactive in protecting their data [10]. There is a potential of targeted advertisement to identify personal health details, with a possibility that future employers could refuse potential employees because of their health risks or personal habits. Insurance companies can purchase personal data and use it to deny coverage or increase premiums. Protections against these practices vary and can be loose in some jurisdictions. Even when such protections exist, the legal expenses can be high, and the case can be challenging to prove. Fog computing can have a role in data protection by moving some data analysis to the edge nodes and anonymizing the cloud's data.
- Personal healthcare and AAL systems can generate a significant quantity of data [11]. Some ALE scenarios, such as fall detection, require having an immediate reaction of the system by triggering an alarm to the care provider. Data pre-processing on edge nodes can significantly reduce the bandwidth requirement and the need for real-time cloud communication.
- Cloud downtime or connectivity issues can be a problem in the case of AAL [12]. While many large cloud providers have multiple availability zones, the cost of having high availability of the cloud is higher. Edge nodes can more easily be clustered, allowing for the high availability of fog computing.

In this paper, we identify multiple benefits of fog computing in the typical AAL scenario and propose an architecture that would make them possible. The AAL fog-based architecture is described in Section 2 of this paper. The proposed architecture benefits are illustrated with the experiment presented in Section 3, and Section 4 concludes the paper.

2. Fog based AAL architecture

Fog computing adds a layer to the cloud computing architecture. However, it should not be interpreted as an extension of the cloud. Fog computing spans to adjacent physical locations. It supports online analytics and various communications networks in performing distributed computing [13]. There are four logical layers of Fog computing.

Data is generated by sensors that can be wearable or body sensors and peripheral or environmental sensors on the first layer. Data can also be generated from external sources such are: social networks, clinical center information systems, or medical databases. Data collected by the sensors can include vital signs, personal habits, or environmental factors. External data sources can provide different information, including medical check results, medical databases for diagnostics, and similar.

The fog layer gathers the sensor data, processes them, and passes either processed or portions of raw data to the cloud. The devices directly connected to the sensors are called edge nodes. Aside from collecting data, they can take action with the user. Each LAN environment can have one or more edge nodes, depending on the application requirements and scale. In elderly care facilities, data for multiple tenants could be processed on the same edge nodes. These actions can include providing feedback to the person to take their medicine or to start exercising. They can directly interact with the environment, such as: activating the humidifier or regulating the room temperature, controlling electrical appliances, and cutting-off for water, gas, and electricity in case of an emergency. The fog network usually has a more limited capacity than the cloud for data computation and cannot do complex machine learning and feature extraction.

However, fog nodes could be able to run algorithms developed by machine learning. As the machine learning system improved and evolved, regular updates could be pushed down to the fog network to improve sensor data patterns. Using this methodology, ADL detecting ML could receive continuous data and improve the detection rate. Events that take a brief time, such as when a person falls, can be detected by the fog nodes using the latest ML model improved in the cloud.

The cloud layer assembles and processes the data from multiple sources and creates machine learning models. The feature extraction is done at this layer as well. Data from the fog and external sources is collected and processed by the data fusion component [14]. The output is an improved knowledge base. The service layer uses this knowledge base in turn.

The service layer is the product of the system. Knowledge obtained by analyzing the data is used for services, including creating customized recommendations for diet and exercise, improving diagnostics systems, providing updates to the health providers, and adding additional information in medical databases.

The critical features that should be satisfied by the system include security, privacy, high availability, and interoperability. Security and privacy [15] can be addressed by implementing best practices to protect the network and the data. Redundancy and automatic fail-over are needed to provide high availability, primarily when the health care recipient's life depends on the assisted living system. The increased complexity requires ensuring connected and inter-operable components by using frameworks intended to ensure mutual compatibility [16].

In fog computing, the nodes nearest to the devices are named edge nodes. In healthcare systems, these nodes represent smart e-health gateways. They act as a bridge for medical sensors to cloud computing platforms. The main requirement of a gateway is to support various wireless protocols and inter-device communication. Its role can be extended to support several features such as acting as a repository to temporarily store sensors' and users' information and bring intelligence by enhancing data fusion, aggregation, and interpretation techniques. It is essential to provide preliminary local processing of sensors' data, which is the primary role of a smart e-health gateway. Smart e-health gateway can tackle many challenges in ubiquitous healthcare systems such as energy efficiency, scalability, interoperability, and reliability issues [17].

Due to the privacy concerns and the technical aspects for scalability and interoperability, it is crucial to identify and trace the data flow in the system. Sensor data originates when sensors acquire measurement from the physical world. This measurement is represented by an electrical signal transferred to a controller that would interpret the signal. Some sensors are manufactured to include the electronic circuits to digitalize the reading, and some are even Internet-connected, enabling them to upload the data to a remote system directly. The sensor data is then passed to the local processing nodes. These nodes are part of the fog and can communicate to other layers of the fog. The data on these edge nodes is processed for local events detection.

Only the edge nodes or smart e-health gateways should be able to get unfiltered raw sensor data. The data that is passed on to other layers of the fog is pre-processed [18]. From this point, the data can be split into multiple processing paths depending on the desired function. Data with person-identifying properties can only be passed to the fog areas used for healthcare provider usage in a compliant way with local regulations for handling medical data. Data used for science research can also contain medical data, but personal identifiers should be stripped or hashed. Other service types might require aggregated data that does not expose the user's medical conditions. It, for example, can include the average time spent outdoors. Such data can be correlated with local weather to determine the best time to organize group activities for the community's senior members. Some data might be of the type that the person would like to share on social media or other platforms. It might include exercise data such as walking, hiking, or riding a bike.

Each of the services dealing with user data is logically independent and can be hosted on separate cloud platforms. The health provider service is independent of social media or medical research databases. The separation of the cloud can be implemented by separation on any level in the fog network. As the data is passed between layers of the fog network, several processing types can occur. Data processing tasks mostly would take place on the edge nodes or smart e-health gateways as the gateways would directly interface with the sensor network and receive raw sensor data. At this layer, we can identify the following types of tasks:

Data filtering is used to filter noise, invalid sensor readings, and redundant information that does not contribute to the desired information the system should induce. Sensor data contains valuable information. However, they also carry non-deterministic errors such as motion artifacts, data corruption issues, and unwanted signals that are also significantly uploaded to increasing storage requirements and power consumption. Fog computing could play an essential role in increasing efficiency and reduce storage requirements for medical big data solutions [19].

Anonymizing of data strips or replacing person-identifying information from data packets. When there is a requirement to separate patient/customer data, personal information is replaced with unique identifiers. This data can be passed on to the fog nodes for added security using an enterprise service bus (ESB).

Data fusion automatically transforms information from different sources and points into a representation that provides practical support for automated decision-making. Applying data fusion in gateways provides several advantages: reduced data ambiguity, extended coverage in space and time, robustness and reliability, and increased data quality. After data is fused, only final results are transmitted through the network so the network bandwidth can be efficiently utilized, and the system can be more energy-efficient [15].

Data processing that can be done on any layer of the fog network includes:

- Data compression is used to reduce the amount of bandwidth required to transmit the sensor network's information. Compression can be lossy or lossless. Lossy compression can be acceptable in many cases, especially if the sensor data's resolution is too high. Besides, the extra information will not cause significant improvements in the algorithms [20].
- Data encryption is used to protect data as it passes through the network. Data encryption can be full or partial. For example, a gateway node would encrypt sensor readings and meta-data of the person. However, the personal information would be encrypted so that only the healthcare provider's network would have the decryption key. The sensor data would be encrypted so the fog nodes would decrypt and, without person-identifying meta-data, pass it onto cloud instances to do statistical analysis or machine learning. This method will reduce data duplication in the network as the same information will not have to be transmitted twice from the gateway to different fog nodes.
- Error code correction can be used to ensure validity during transmission. The fog network can rely on various data transmission techniques and technologies to pass on the information; sometimes, the network protocol would have a built-in feature to ensure valid transmission. When this is not the case, the fog nodes would have to ensure the data's validity by identifying and correcting transmission errors. The same applies to the data traveling from the sensors to the gateway, as many sensors do not have a buffer memory and cannot re-transmit data. Error code correction will be used to identify faulty readings and discard them (because having gaps in the data is often better than having inaccurate data).

3. Experiment

A common usage of sensor networks is to train machine learning models and enable different end-user actions. Depending on the number of sensors used, the number of features extracted from the sensor data, and the data generation rate, generating the model will most likely be done in the cloud due to the resources demand and a potential need to use data from other locations. On the other hand, the implementation of the model can and should be done on edge. As an example, we will consider a data flow model to detect sleep apnea using noninvasive sensors, illustrated in Figure 1.

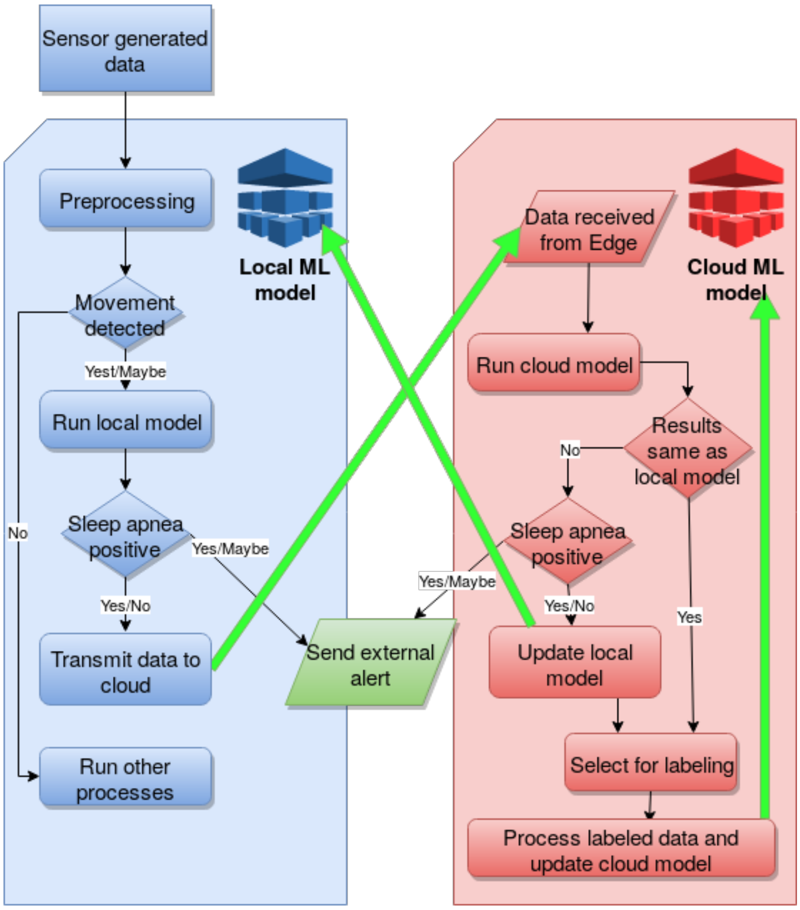

Figure 1: Data processing for unobtrusive sleep apnea detection

The sensor readings from multiple care recipients are collected. In scenarios of multiple occupants, such as in a hospice, edge nodes retain personal or identifiable information, which is then stripped by the edge.

The sensor readings from multiple care recipients are collected. In scenarios of multiple occupants, such as in a hospice, edge nodes retain personal or identifiable information, which is then stripped by the edge.

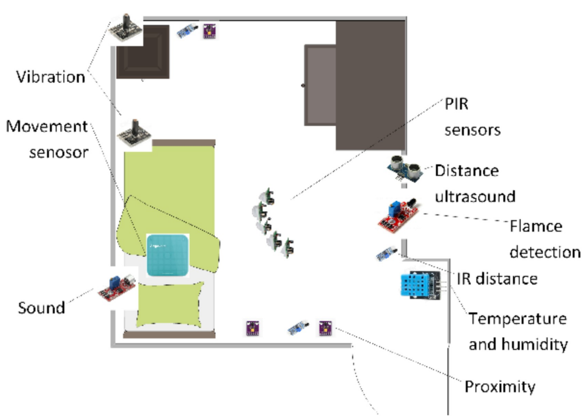

Figure 2: Floor plan and sensor layout

The first phase is to pre-process the data by identifying body movements. As described in [21], sleep apnea is accompanied by body or leg movement, which noninvasive sensors can detect. We have used multiple PIR sensors and piezoelectric-based sensors placed under the mattress (see Figures 2 and 3).

Figure3: Sensor for movement detection on the bed under the mattress

The strong correlation between the two sensor types, shown in the diagram of recorded sensor data over 8 hours, is presented in Figure 4.

When motion is detected, the data from multiple noninvasive sensors is processed on the edge node. The local machine learning model is run, and the possible occurrence of sleep apnea is diagnosed. Periodical sessions with invasive sensors or medical professionals' observations can be carried out to label the data set [22].

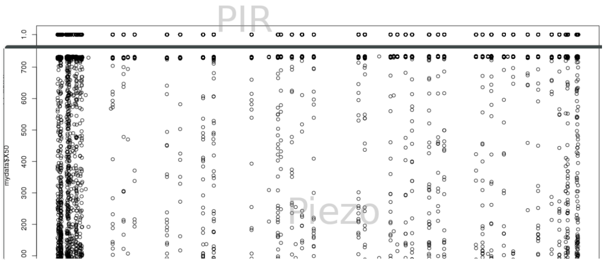

Figure 4: Movement in bed over 8 hours of continuous sleeping

After anonymization of the data, it is packaged and sent to the cloud for additional processing. The data model on the cloud side is run to verify the outcome for the received data. If the model present in the cloud makes positive detection for the received data and if the data was previously labeled with a negative result by the edge node, then the updated model is sent back to the edge node, which in turn processes the data against the updated model.

Sensor data that does not suggest strong negative results are marked for further labeling if additional data such as monitoring from medical equipment or video that can be analyzed by a trained professional is available. Such feedback is periodically included to build the cloud model continuously.

3.1. Classification algorithms

This section explains the classification algorithms used for feature ranking and construction classification models. The accuracy was used for the comparison of various classification models throughout the system. One of the classification algorithms used in our experiments is logistic regression [23]. For small datasets, it is straightforward and provides easily interpretable models. Moreover, it is a lightweight algorithm, which can be useful if the system is deployed on hardware with limited resources.

Random Forest (RF) [24] is an effective algorithm that creates an ensemble of decision trees [25] by randomly sampling training instances from the dataset. The sampling is random but consistent while growing a single tree. The multiple decision trees are trained on the training data independently.

The tree branching is performed by finding the best split from the features on each node. During classification, trees vote for the class, and the majority class is eventually predicted. Like RF, the Extremely Randomized Trees (ERT) algorithm [26] also generates trees' ensembles. ERT chooses the split from the attributes randomly, unlike RF. As a result, the number of calculations per node is decreased, thus increasing the training speed. Both algorithms provide excellent classification performance and can train models on extensive datasets very fast.

Both ERT and RF provide feature importance estimates, a property used for feature ranking and discarding of low-importance features during the feature selection phase. We have used the feature importance estimates when training an ERT classifier due to its better speed than RF.

Additionally, we have also used the Support Vector Machines (SVM) classifier [27] with Gaussian kernel. Even though SVMs are much slower algorithms as the dimensionality of data increases, they are compelling, especially after parameter tuning [28]. Whenever we used SVMs, the datasets were normalized so that the training dataset will have a mean and standard deviation of 0 and 1, respectively. The RF and ERT parameters were the default per their implementation in the [29] library. We did not notice any significant gain by tuning their parameters (i.e., number of features per tree). Both ERT and RF classifiers were trained using 100 trees, which was appropriate for this size dataset. Using fewer trees improved the speed while offering slightly worse classification performance. This library was used for the other classification algorithms as well.

3.2. Feature extraction

The measurements from sensors can detect atomic actions or states. More complex actions are depending on the context, which recent measurements can determine. Therefore, the data needs to be first adequately segmented, and then feature extraction performed [30]. This study additionally discusses the window size impact on activity recognition. Generally, lower sensor frequencies entail longer windows. It is considered during our experiments by using different window lengths and analyzing the accuracy depending on them. The segmentation into windows, step 1 on Figure 5, was performed, thus excluding the border intervals when the activity changes from one activity to another.

Segmenting of streaming data into windows is performed in step 2 in Figure 5. Step 2 extracts the following types of features (the number of measurements within one window is denoted by n.):

- Basic statistics results in 14 features per time series.
- Equal-width histogram calculated with $[\log_2 n+1]$ intervals, based on the Sturges rule [31]. It results in 5 to 8 features when the window length varies from 5s (25 measurements) to 20s (100 measurements).
- Quantile-based features: first quartile, median, third quartile, interquartile ranges, and other percentiles (5, 10, 20, 30, 40, 60, 70, 80, 90, 95), also used in [32]. From one-time series, it generates 14 features.
- Auto-correlation of the measurements within one sliding window [33]. Let τ denote the amount of shift, and its domain is defined as $\tau \in [1, \lfloor\sqrt{n}\rfloor]$
- For exponentially increasing values of τ in that range, classical autocorrelation and Pearson correlation are calculated. Additionally, it calculates both correlations using the first and second half of measurements within one sliding window. This results in 3 to 4 τ values when the window length varies from 5s (25 measurements) to 20s (100 measurements).
- Pearson correlations between pairs of time series; For five-time series, this results in 9 features.
- Linear and quadratic fit coefficients; There are two linear fit and three quadratic fit coefficients, yielding five features in total per time series.
- As a result of step 2, 250 to 270 features are generated depending on the window length. In step 3 performs feature importance and drift sensitivity estimation is done. Next, step 4 performs coarse-grained feature selection, which tests a set of thresholds used to discard features with low importance or high drift sensitivity.

The system evaluates different feature sets by building classification models using the training dataset and evaluating them with the validation dataset. The test set is not utilized at this stage at all. Thus, only the feature set that results in the best classification accuracy is retained. To summarize, the purpose of this step is to significantly reduce the feature set size by discarding features with low importance or high data drift sensitivity.

The system evaluates different feature sets by building classification models using the training dataset and evaluating them with the validation dataset. The test set is not utilized at this stage at all. Thus, only the feature set that results in the best classification accuracy is retained. To summarize, the purpose of this step is to significantly reduce the feature set size by discarding features with low importance or high data drift sensitivity.

After the feature set is reduced, step 5 uses the training and validation sets to perform parameter tuning for the SVM.

Finally, step 6 evaluates different classifiers by building classification models with the training and validation dataset's union and evaluating it using the independent test set.

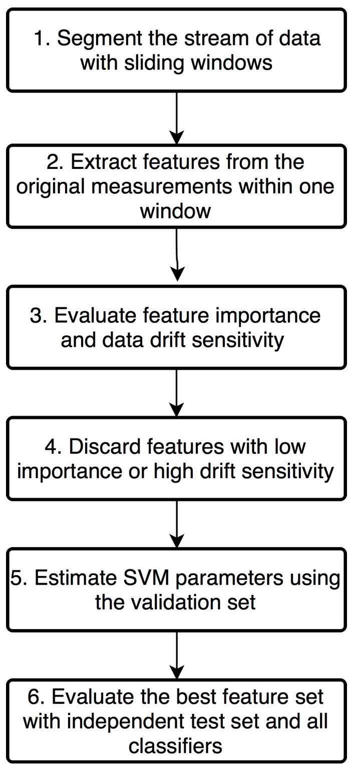

Figure 5: Feature extraction, selection, and classification flow

4. Results and evaluation

The duration of our experiment was 8 hours. The sampling rate was set to 10Hz, thereby measuring ten values from each sensor every second. We divided the dataset into three different subsets: training, validation, and testing. The training subset consisted of the first 45% records for each action, and the validation subset consisted of the next 25% records. The remaining 30% of records belonged to the test subset. When performing parameter tuning for SVMs and making feature selection, the training set was used to build models, and the validation set was used to evaluate their performance. Once this phase was completed, the final evaluation was performed only with the best feature set decided after the feature screening and using the most optimal parameters. The union of the training and validation sets was used to build classification models for making final predictions. The test set was used for building predictions and the performance evaluation.

5. Conclusion

As personal health becomes pervasive and the data generated by it increases in volume, fog computing offers a solution for many critical challenges. The added flexibility of the fog architecture enables better placement of computing and network resources. Smarter data flow could protect personal data, bandwidth cost could be reduced, and more scalable, secure, and interoperable systems can be designed. This paper illustrates those benefits by providing an experimental illustration of typical AAL service provided by fog-based health care ELE.

By using simple hardware, the AAL data was streamed to a cloud-based system, where it was fused. Using a systematic and automated feature extraction and selection process, we could extract robust and reliable features that facilitated building powerful classification models.

6. Acknowledgements

This work was partially financed by the Faculty of Computer Science and Engineering at the Ss. Cyril and Methodius University, Skopje, Macedonia and is supported by the networking activities provided by the ICT COST Actions IC1303 AAPELE and CA16226 SHELD-ON.

We also acknowledge Microsoft Azure's support for research through a grant providing computational resources for this work.

7. References

[1] M. Aazam and E.-N. Huh, “Fog computing and smart gateway based communication for cloud of things,” in *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*. IEEE, 2014, pp. 464–470.

[2] T. N. Gia, et al., “Fog computing in health- care internet of things: A case study on ECG feature extraction,” in *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*. IEEE, 2015, pp. 356–363.

[3] V. Stantchev, et al., “Smart items, fog and cloud computing as enablers of servitization in healthcare,” *Sensors & Transducers*, vol. 185, no. 2, p. 121, 2015.

[4] P. Maresova, et al., "Technological Solutions for Older People with Alzheimer's Disease" *Current Alzheimer Research*, Volume 15, No. 9 April 2018, DOI:10.2174/1567205015666180427124547, pp. 975-983

[5] R. Goleva, et al.,, “AAL and ELE Platform Architecture” in book: Ambient Assisted Living and Enhanced Living Environments: Principles, Technologies and Control, , Elsevier, 2016.

[6] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, “Cost-efficient resource management in fog computing supported medical cyber-physical system,” *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, pp. 108–119, 2017.

[7] O. Kotevska, et al. "Towards a Patient-Centered Collaborative Health Care System Model," *International Journal of Computer Theory and Engineering* , vol. 4, no. 6, November 2012, DOI:

10.7763/IJCTE.2012.V4.631, pp. 1025-1029.

[8] L. Xu, et al."What Clinics Are Expecting From Data Scientists? A Review on Data-Oriented Studies Through Qualitative and Quantitative Approaches." In *IEEE Access,* Volume 7, January2019,DOI:10.1109/ACCESS.2018.2885586, pp. 641-654.

[9] I. Kulev, et al. "Recommendation Algorithm Based on Collaborative Filtering and its Application in Healthcare", *The 10th Conference for Informatics and Information Technology* (CIIT 2013), Bitola, R. Macedonia, April 18-21, 2013.

[10] M. Barhamgi, et al., "Enabling end-users to protect their privacy," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 2017, pp. 905–907.

[11] E. Vlahu-Gjorgievska, et al., "Connected-Health Algorithm: Development and Evaluation." *Journal of Medical Systems*, vol. 40, no. 4, 2016, DOI: 10.1007/s10916-016-0466-9, pp. 1-7.

[12] A. M. Rahmani, et al., "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.

[13] Y. Shi, et al., "The fog computing service for healthcare," in Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech), 2015 2nd International Symposium on. IEEE, 2015, pp. 1–5.

[14] A. Zdravevska, et al, "Cloud-based recognition of complex activities for ambient assisted living in smart homes with noninvasive sensors," in *Smart Technologies, IEEE EUROCON 2017-17th International Conference on*. IEEE, 2017, pp. 769–774.

[15] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the internet of things: Security and privacy issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, 2017.

[16] M. Memon, et al, "Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes," *Sensors*, vol. 14, no. 3, pp. 4312–4341, 2014.

[17] A.M. Rahmani, et al, "Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems," in *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*. IEEE, 2015, pp. 826–834.

[18] A. Atanasov, et al. "Testbed Environment for Wireless Sensor and Actuator Network", in Proc of the *5th International Conference on Systems and Network Communications* – ICSNC 2010, CD publication, Nice, France, August 2010.

[19] H. Dubey, et al, "Fog data: Enhancing telehealth big data through fog computing," in *Proceedings of the ASE BigData & SocialInformatics 2015*. ACM, 2015, p. 14.

[20] E. Zdravevski, et al. Improving activity recognition accuracy in ambient-assisted living systems by automated feature engineering. *IEEE Access,* 5,2017, pp. 5262-5280.

[21] V. K. Somers, et al.,"Sympathetic neural mechanisms in obstructive sleep apnea." *Journal of Clinical Investigation*, vol. 96, no. 4, 1995, p. 1897.

[22] J.-C. Vazquez, et al., "Automated analysis of digital oximetry in the diagnosis of obstructive sleep apnoea," *Thorax*, vol. 55, no. 4, 2000, pp. 302–307.

[23] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, Applied logistic regression. John Wiley & Sons, 2013, vol. 398.

[24] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, 2001, DOI: 10.1023/A:1010933404324, pp. 5–32.

[25] J. R. Quinlan, "Induction of decision trees," Machine learning, vol. 1, no. 1, 1986, pp. 81–106.

[26] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," Machine learning, vol. 63, no. 1, 2006, pp. 3–42.

[27] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, 1995. DOI: 10.1007/BF00994018, pp. 273–297.

[28] P. Lameski, E. Zdravevski, R. Mingo, and A. Kulakov, "Svm parameter tuning with grid search and its impact on reduction of model overfitting," in Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. Springer, 2015, pp. 464–474.

[29] F. Pedregosa, et al., "Scikit-learn: Machine learning in Python," Journal of Machine

Learning Research, vol. 12, 2011, pp. 2825–2830.
[30] O. Banos, et al., “Window size impact in human activity recognition,” Sensors, vol. 14, no. 4, 2014. DOI: 10.3390/s140406474, pp. 6474–6499.
[31] H. A. Sturges, “The choice of a class interval,” Journal of the American Statistical Association, vol. 21, no. 153, 1926, pp. 65–66.
[32] P. Siirtola and J. R¨oning, “Recognizing human activities user independently on smartphones based on accelerometer data,” International Journal of Artificial Intelligence and Interactive Multimedia, vol. 1, no. 5, 2012, pp. 38–45.
[33] H. Mart´ın, A. M. Bernardos, J. Iglesias, and J. R. Casar, “Activity logging using lightweight classification techniques in mobile devices,” Personal and Ubiquitous Computing, vol. 17, no. 4, 2013. doi: 10.1007/s00779-012-0515-4, pp. 675–695.