Fog and IoT-based Remote Patient Monitoring Architecture Using Speech Recognition

Marc Jayson Baucas and Petros Spachos
School of Engineering, University of Guelph, Guelph, ON, Canada

***Abstract*—Health care services have become a high demand due to the rise in medical technology. As a result, related resources are being depleted. Hospitals no longer have any space to accommodate for incoming patients. Remote Patient Monitoring (RPM) is a solution to this issue by creating a convenient and easy to access healthcare service. However, RPM systems are constrained by concerns on patient privacy, response time, and patient-service interaction. Patients emphasize their privacy, which requires health care services to maintain the confidentiality of their patient's information. Wearable health monitors continuously transmit data. This feature results in high volumes of data transmissions towards the servers. In the current state of wearable devices, there is a lack of giving the patient an integrated means of interacting with the healthcare centre and vice versa. In this paper, we propose an architecture that uses fog computing and Internet of Things (IoT) devices to an already existing RPM system and addresses these challenges. The introduced system enables the health care providers to verify any of their data through a local server before it is reported to the main server. Also, this design incorporates a data filter that controls the outgoing data to maintain patient privacy. Finally, the inclusion of a local server offloads the extra data processing that is required from the server for a better flow of data. Tests in latency were executed to investigate the feasibility of a scalable fog architecture against a standard cloud-device setup. The results show that the proposed fog setup yielded significantly lower latencies under an increasing number of RPM rooms compared to the cloud setup. Results further support the fog and IoT-based architecture as a potential option for a scalable RPM.**

I. Introduction

Health care services are always interested in using technology as the driving force for improvements and changes to their established systems [1]. To further supplement their need for fundamental change, service providers such as hospitals, individual physician practitioners, and pharmaceutical companies have all started to integrate information systems. Internet of Things (IoT) is one of the systems that is being used by the health care industry. This decision is due to its ability to improve the quality of their services [2]. As a result, this industry has started developing applications that revolve around IoT networks.

An application that uses IoT is Remote Patient Monitoring (RPM) [2]. An RPM is a telemonitoring system that patients are given wearable devices to monitor their health. Then, this information is sent to their health care providers for analysis. This integration of big data monitoring and health services brought rise to health-based smart spaces called smart healthcare [3]. The resulting ecosystem allows these patients to be observed for any anomalies in their health within a remote setting. RPM systems benefit the patients by no longer requiring them to travel to diagnose illnesses or go for check-ups. At the same time, health care services minimize the need for physical space since patients are no longer required to be within the hospital for a check-up. These benefits allow emergencies to be detected and handled earlier due to the data being transmitted in real-time. Patients have also ensured a better quality of care due to the low cost and enriched experience of opting in to this IoT-based service [2].

However, RPM systems have several open issues and challenges. One issue is in the privacy of the patient's information [4]. User information that is sent wirelessly can easily be compromised due to the open nature of wireless medium. As a result, patients are discouraged from using RPM systems. Another issue is the volume of data that needs to be managed by the server when receiving sensor data from the wearable devices of the system. These devices are expected to transmit data to health care providers continuously. This continuous transmission of data results in a large influx of unprocessed data, which results in an overloaded server. The third issue is in the system's ability to allow the patient and the service to interact, especially during emergencies. During these times, RPM systems lack the means of allowing a more reactive way of providing situational data to the healthcare centre [5]. Without a means of providing more dependable data to the healthcare centre, their ability to react to different situations and interact with the patient is kept to a minimum.

In this paper, we extend our previous approach that focuses solely on standard cloud-based IoT [6], and propose an improved framework. We incorporate a fog-based structure to address the privacy, data flow, and reliability issues that are affecting RPMs. Also, we conduct experiments to examine the ability of the introduced framework to properly address these issues.

The rest of this paper is as follows: A discussion on Fog-IoT-based RPM systems, the open issues, the related work, and the proposed solution is presented in Section II. Section III presents the details of the proposed framework including each component and the introduced design. Section IV reports the preliminary results of our proposal. Finally, Section V contains the conclusions we formulated based on the results of our implementation.

978-1-7281-8086-1/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: University of Guelph. Downloaded on February 02,2021 at 22:06:48 UTC from IEEE Xplore. Restrictions apply.

II. Fog-IoT in RPM

IoT-based RPM systems are usually wireless network services that allow a patient to be observed by a medical centre without having to be physically present in a clinic [7]. Distance is no longer a constraint for health care providers since IoT has made it possible for data to be obtained over a wireless network. The monitoring technology has even evolved to incorporate common devices such as smartphones [8] in their services. It is used to connect the devices needed for monitoring the patient within a wireless network [9]. Introducing fog devices allow the balancing of processing load across the network [10], [11]. As a result, data traffic can now be redistributed to a local server [12]. This feature allows Fog-IoT to enable RPM to be available to a larger demographic by making it more convenient not only for the patient but also for the health care providers. With the wider scope from the extended servers through the fog, patients are no longer required to travel to reach their family doctor. Also, medical centres no longer need to worry about resources and costs when checking on their patients [13].

A. Open issues with current RPM systems

1) Patient privacy: A challenging issue for patients who are presented with monitoring services is their privacy [4]. Patients find discomfort in disclosing their information. As a result, the likelihood of medical centres of implementing RPM systems is low [14]. Another aspect that contributes to dissuading patients from investing in the service is due to its wireless component. Wirelessly transmitting data over long distances could result in various security attacks [15]. Healthcare services are responsible for keeping the data that is obtained from a patient's private and confidential. A system that is left insecure can easily result in stolen patient data. RPM or wireless health monitoring systems need to maintain an adequate level of security towards their patient's data [16]. Proper data filtering would be able to prevent such attacks from being able to steal important data.

2) Data flow: Medical centres use wireless devices to monitor their patients for the long term [17]. These devices are programmed to transmit data continuously over the network. However, continuous transmission of data can lead to several complications with the server. The biggest contributor to server overload is from the continuous transmission of data of the end devices [8]. Health centres need to receive data in real-time, to be able to diagnose complications early, and to react to emergencies on time. Therefore, these wearable devices must stay on to make sure that a patient is fully monitored by their medical service providers. In terms of the wearable device, the capacity of the device to do calculations affects the flow of data. If this raw sensor data is pre-processed before it reaches the server, there will be better control over the network in terms of traffic. Reallocating processes away from the server can alleviate the data traffic. However, the amount of changes that can be done to the end devices is minimal. Therefore, in the proposed system, we chose to focus on including another device that is more capable of handling the pre-processing before it is sent to the server.

3) Patient-Service interaction: The interactability of RPM systems in case of emergencies is important for both the healthcare provider and the patient. An RPM system needs to be more interactive to provide more information about a patient's current situation. However, clinics in the current state of RPMs lack the means of establishing a more interactive way of checking on a patient [5]. Data from wearable devices come as raw and easy to process. However, it cannot fully give the healthcare centre the ability to gauge certain situations without asking for more data [18]. Therefore, adding a component that can further give medical centres the ability to react accordingly towards an emergency or false alarm is needed.

B. Proposed solutions

The introduced system proposes solutions to the privacy, data flow issues, and patient-service interaction that were cited through the inclusion of a fog device that can pre-process the raw data from wearable devices before it is sent to the server. Our proposed system uses a fog-variant of IoT by using fog technology to be an intermediate server that can offload the data strains away from the server to an equally capable device.

The fog device design has three parts that address each cited issue individually.

1) Data filter: Privacy is a big topic for any system that collects data from its customers, especially when monitoring cameras are used. Adding a camera feed might help for fast detection but it adds to the issue of privacy. Thus, we proposed an intermediary layer between the wearable device and the medical centre. This layer is the main component of our framework, which will filter any outgoing data. We propose an interactive framework that serves as this filter that will require consent from the patient before it will send any form of data to the medical centre. This framework will converse with the patient whenever it detects an anomaly from any of their wearable devices. Based on the patient's response, it will either notify the medical centre or not. This design will be programmed to be reactionary. It will account for severe emergencies where the patient is incapable of interacting with the framework. Therefore, we envision a smart interactive framework that can interact with the patient in a manner that allows private information to stay private while still maintaining a good quality of service.

2) Fog device and process reallocation: Data is being continuously transmitted from the devices worn by each patient. This transmission results in disruptions in the flow of data. Delaying the arrival time of data can also delay a health centre's response to an emergency. To avoid this, we propose the reallocation of processes from the main server to the fog. As a result, the overall size requirement of the transmitted data can be reduced to a more workable size for the server to manage.

3) Confirmation through surveillance: RPM systems need a more interactive way of bridging the patient and the healthcare centre. Therefore, we propose a way to send more

Authorized licensed use limited to: University of Guelph. Downloaded on February 02,2021 at 22:06:48 UTC from IEEE Xplore. Restrictions apply.

information about the current situation of the patient to the centre. By adding a camera to the system, a medical centre obtains a means of obtaining patient information that will be more descriptive since it's a video streamed. Also, it adds to their ability to react timely since the video is wirelessly sent to the cloud in real-time. As a result, health care providers are no longer constrained by relying on raw data to evaluate different situations. With live video feeds, a medical centre can now assess any emergencies without requiring an explanation from the patient, in case they are not able to provide any feedback. In serious emergencies where a patient is unable to call for medical assistance, a controlled video feed could make a faster response possible. Also, health care providers can detect any malfunctioning components in the system based on the camera feed. They can check if it coincides with the obtained data reports. However, adding this component can expose the privacy of patients in exchange for faster response times and more accurate data. We take advantage of the data filter that was proposed to protect the patient's data while improving the RPM system's performance accordingly.

III. Proposed Framework

We propose an architecture that pre-processes the sensor data before it is sent to the server. A data sensing model using a neural network was used to simulate the collection and processing of data to create a remote interactive response system for patients. With the given consent of the patient, our framework will record and transmit data to the medical centres during emergencies. The introduced system is modelled to have controlled surveillance, where only permitted data is transmitted from the end devices to the fog device, then to the cloud. Upon reception of a response from the user, the fog device will process the information and then filters the outgoing data accordingly. This step creates a more confidential system that will prioritize the well-being of the patient over the volume of the data transmitted. The data is then sent to the server for further processing. In terms of sound collection, a listening device is used to interact with the patient to record their responses. The proposed IoT-based RPM system is shown in Fig. 1.

The introduced framework is composed of three major sections:

i. Recording of sound,
ii. surveillance, and
iii. speech recognition.

The general flow of data within the proposed framework is shown in Fig. 2. It starts when the Pis detect an anomaly from the attached health monitor. Next, the Pi will prompt the patient to ask if she/ he needs assistance. Then, the Pi will start waiting for the patient's response. The response will be sent to the fog device for classification. If assistance is requested, it will immediately notify the medical centre along with a live feed of the room where the patient is located. If and only if an anomaly is detected, then this live capture will only be turned on. Any other time, the camera is turned off. This feature is implemented to make sure that privacy is

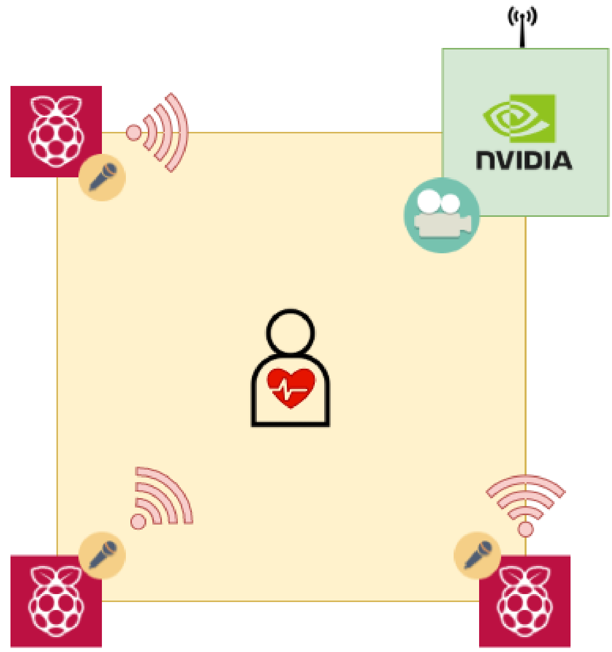

Fig. 1: Proposed IoT-based RPM system, consisting of three listening devices and one camera as the main processing unit.

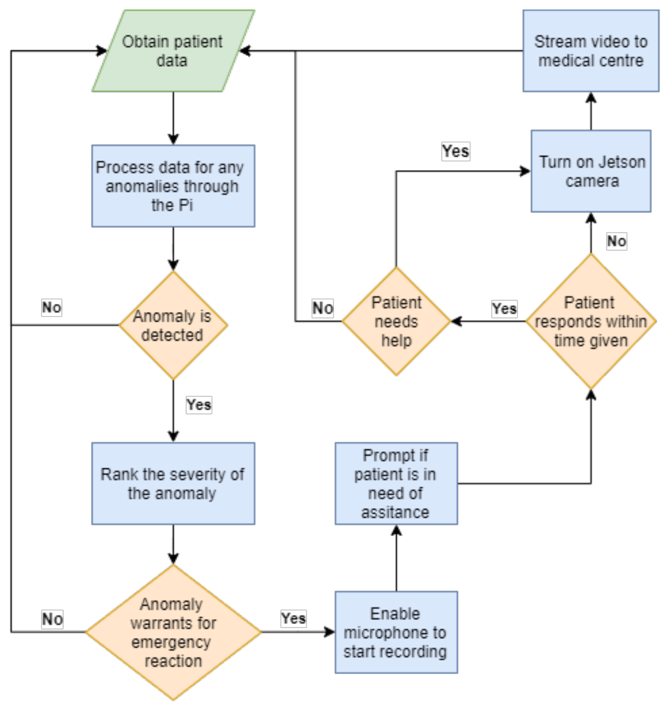

Fig. 2: General design flow of the proposed system.

retained. Based on this notification of emergency and live feed of the current room where the patient is located, the medical centre is expected to respond accordingly. Otherwise, if the centre deems that the patient does not need any assistance or if the detected data is a false alarm, the Pi will return into a low-powered sensing state.

We take an already existing health monitor system

Authorized licensed use limited to: University of Guelph. Downloaded on February 02,2021 at 22:06:48 UTC from IEEE Xplore. Restrictions apply.

(i.e. ECG) and use our proposed architecture as a data filter between patient and medical centre. As a filter, it will only transmit data that is required by the medical centres, which addresses privacy and flow, while providing interactability.

A. Components

Our framework has the following hardware and software components.

1) Monitoring devices: The framework uses 3 Raspberry Pis, an NVIDIA Jetson TX1 developer kit, and a personal computer (PC) as the RPM. Due to their programmability, the Raspberry Pi will be used to run the code that will be in charge of collecting the patient's response and data. Since the Pis are portable and modular, multiple Pis with the same function can be strategically placed around to room to maximize the effectiveness of the framework in detecting the response from the patient. These Pis are also capable of being used as smaller data filters by taking incoming data from the wearable health monitoring devices and pre-processing them for early detection of health anomalies. The Jetson board is a developer board used for prototyping computing tasks and other user-defined processes. It was selected due to its built-in GPU and camera modules, which provide a programmable unit that is convenient, compact, and optimized for image recording and complex data processing. In terms of operating systems, we loaded each Pi with a Raspbian operating system while the Jetson board was already pre-loaded with an Ubuntu 16.04 operating system.

2) Sound recording: Each Pi will be equipped with a sound recording device to listen to the response of the patient. The device that we used is an STM32 NUCLEO-64 board with an attached X-NUCLEO-CCA02M1 expansion board to serve as a digital Micro-Electrical-Mechanical System (MEMS) microphone. This microphone was chosen due to its low power and programmability for more optimal recording streams and formats that the Pi can handle. Next, we attached the microphone to the Pis via its USB hub and controlled it using a Python script. Our script makes use of a library called PyAudion. This library allows us to create a recording stream for our framework. Then, we configured our stream to match the default settings of the microphone, which was a sampling rate of 16 kHz and 16 bits of sound resolution. After recording the response from the patient, the Pis will transmit the sound data using a wireless socket to the fog device, which is Jetson board for further data processing. The logic flow of the sound recording mechanism using the Pi is shown in Fig. 3.

3) Data management: We take advantage of the built-in GPU of the Jetson board to handle all the more complex processes of our framework. It will be in charge of pre-processing the sound data before it can be used by our classifier. By having multiple sources of the patient's response, we programmed the Jetson to evaluate all three inputs for better coverage over the patient. The resulting feature maps will be classified by a model that uses a Convolutional Neural Network (CNN) as its base. We selected this neural network

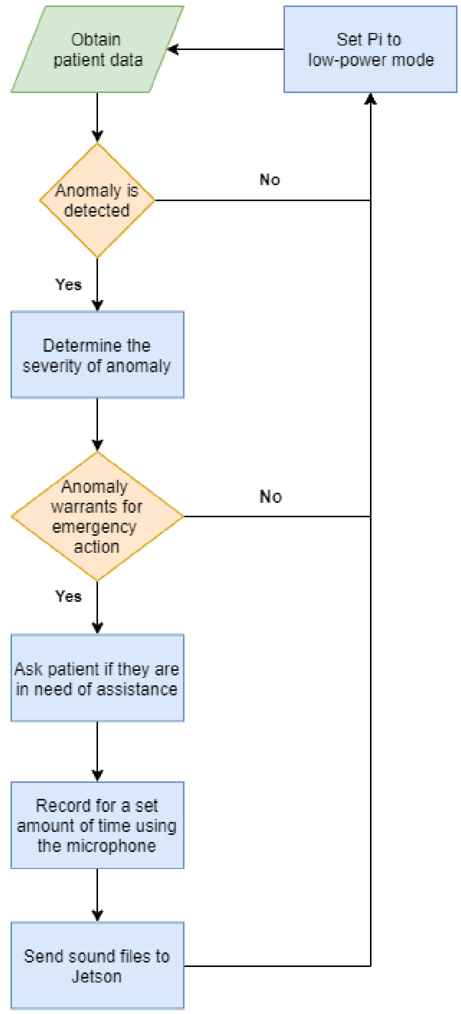

Fig. 3: Raspberry Pi operation flow.

due to its advantages in modelling acoustic data [19]. Also, CNN is known for its speech recognition toolkit [20].

4) Speech classification: We used Python to program the CNN that is used by this framework. We used Tensorflow and Keras as libraries to build the models. Also, we used Scipy and Numpy to process the sound and extract the features. We programmed our CNN using four types of layers: Convolutional Layer, Pooling layer, Normalization layer, and Full-Connected or Dense layer. A combination of layers is stored as blocks that complete the neural network sequence. Within each block are the convolutional layer, pooling layer, and the normalization layer. The flow of the network composes of three identical blocks where the input will pass through. The network will then flatten the output from these blocks before the dense layer decides the prediction of the model.

We have tested our model against two simple terms [6]. To improve its complexity, we decided to increase its scope by adding more terms that can help provide a better context. This addition allows the service to get a better understanding of the situation. As a result, we prepared a model that classified the following terms; “yes”, “no”, “on”, “off”, “bed”, “stop”,

Authorized licensed use limited to: University of Guelph. Downloaded on February 02,2021 at 22:06:48 UTC from IEEE Xplore. Restrictions apply.

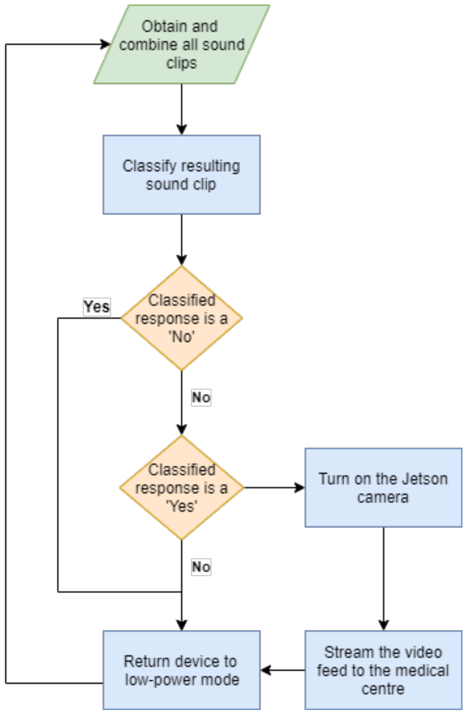

Fig. 4: Jetson operation flow.

"left", "right", "up", and "down". For future iterations, we plan to create a more flexible interface that can interact with the patient with phrases instead of single words.

5) Surveillance capture: To further monitor their patients in case of emergencies, we needed another physical layer that medical centres can use. Data is not enough to indicate if a patient needs assistance. Therefore, we added a surveillance aspect to our framework. This design was done to help our medical centres double-check on the patient in case of an anomaly. However, we still needed to pay attention to the patient's privacy. Therefore, we programmed the Jetson camera to turn on and stream a video feed to the medical centre only if the patient gives their consent. By taking advantage of the speech recognition feature of the framework, we can interact with the patient in case of anomalies. With this design, we create a more secure way of surveying the patients without crossing their privacy. A diagram of the logic behind how the Jetson board will process the sounds and how it will decide on either turning on the camera or keeping it off, is shown in Fig. 4.

6) Sending to the server: The final results of the pre-processing are then transmitted via WiFi to the PC that serves as the server. The PC will be continuously waiting for the data from the Jetson through a wireless socket. This device marks the end of the data flow, where the received data is stored.

IV. Results

A. Training the network

Using the model and dataset from [6], we customized our neural network to add more words to allow more context. In training, we split the dataset 70-30, hence, 70% of the dataset was used to train the model and the remaining 30% was used to verify its accuracy. In Fig. 5, the accuracy of the model as the verification accuracy approaches the intended accuracy as the number of epochs increased, is shown.

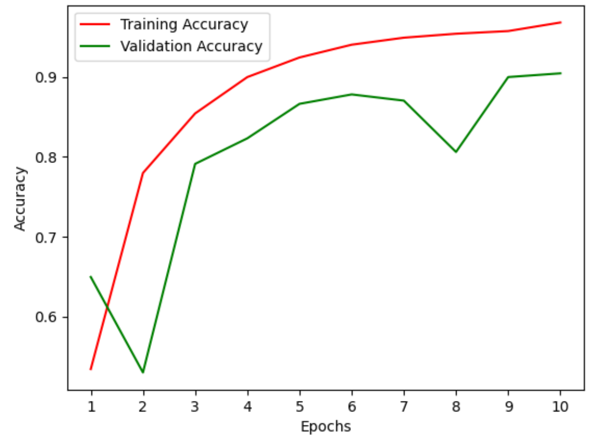

Fig. 5: Training and validation accuracy.

An epoch value of 10 was observed to be the cutoff point of the training model. This cutoff point means that it could no longer improve its accuracy and has reached a projected steady state. The resulting accuracy of our model was 96.8%. The training and testing result shows that the model was able to retain its accuracy even after increasing its complexity. This observation shows promise for future iterations and how we can move forward with more complex datasets with the same training approach.

B. Latency

To get a better understanding of the data flow of the proposed architecture, we also measured its latency. This metric was compared with that of a standard cloud network setup. This setup takes away the fog server, which makes the Pi send the data directly for the server. The server then evaluates the sound and sends instructions to the Jetson to either turn on the camera or stay on standby.

We specifically chose to measure the round trip latency to model the responsiveness of the server to incoming data. The experiment was carried out by increasing the number of room setups attempting to connect to the server. With the fog setup, the Jetson will be the only one that will communicate with the server from the room. For the cloud setup, each room will have three Pi's that collect the data and will attempt to communicate with the server. This measurement also points out that dealing with Pis in groups of three will be harder to coordinate for the cloud set up as the network grows. The average latency measurements with 1, 2, and 3 room setups at 40 iterations of the data transmission are shown in Fig. 6. The results show that the latency of the cloud setup is significantly greater than that of the fog. This difference could be attributed

Authorized licensed use limited to: University of Guelph. Downloaded on February 02,2021 at 22:06:48 UTC from IEEE Xplore. Restrictions apply.

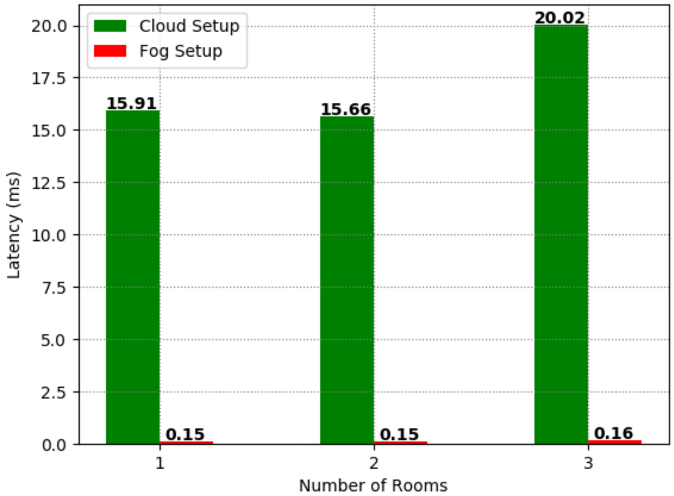

Fig. 6: Average latency of Fog and Cloud setups within 40 iterations.

to the size of the data being transmitted. The Jetson board only needs to transmit the results of the speech classification, while three Pis from each room setup will have to transmit the whole sound clip.

High latencies will keep a server from being able to respond to the reports on time. In this case, the network may still handle the latency. However, if the setup starts with such a high latency, then scaling up the architecture will be challenging. Thus, the fog setup shows better potential at handling the data flow in terms of latency due to the workable data size and more manageable arrangement of devices.

V. Conclusion

For most RPM systems that are being used by medical centres, user privacy, latency, and patient-service interactability are a rising concern. These concerns keep patients from opting in for this service. Our proposed framework aims to integrate itself as a data and information filter to an already existing health monitor system to minimize these concerns. We used speech recognition and proper surveillance to create a way to interact with the patient. The data that is sent to the medical centres are pre-filtered to prioritize the privacy of its patients. Our classifier builds on top of our previous iteration with added words for more interactivity and was trained for 10 epochs using the Google Speech Command dataset. The training resulted in an accuracy of 96.8% which shows the resiliency of the model and its potential for more complex scopes. Regarding the latency, the fog setup yielded lower values compared to the cloud setup. These results can be attributed to the size of the data being transmitted in combination with network traffic. To further improve our design, we plan to focus on other metrics such as runtime and data packet throughput for future iterations.

References

[1] K. Siau, "Health care informatics," *IEEE Transactions on Information Technology in Biomedicine*, vol. 7, no. 1, pp. 1–7, March 2003.

[2] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. Kwak, "The internet of things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[3] R. Dautov, S. Distefano, and R. Buyya, "Hierarchical data fusion for smart healthcare," *Journal of Big Data*, vol. 6, 12 2019.

[4] A. Abbas and S. U. Khan, "A review on the state-of-the-art privacy-preserving approaches in the e-health clouds," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 4, pp. 1431–1441, July 2014.

[5] A. Kara, "Protecting privacy in remote-patient monitoring," *Computer*, vol. 34, no. 5, pp. 24–27, May 2001.

[6] M. J. Baucas and P. Spachos, "Speech recognition driven assistive framework for remote patient monitoring," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2019, pp. 1–5.

[7] Y. Chen, W. Sun, N. Zhang, Q. Zheng, W. Lou, and Y. T. Hou, "Towards efficient fine-grained access control and trustworthy data processing for remote monitoring services in iot," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1830–1842, July 2019.

[8] N. Alshurafa, J. Eastwood, S. Nyamathi, J. J. Liu, W. Xu, H. Ghasemzadeh, M. Pourhomayoun, and M. Sarrafzadeh, "Improving compliance in remote healthcare systems through smartphone battery optimization," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 1, pp. 57–63, Jan 2015.

[9] P. Verma and S. K. Sood, "Fog assisted-iot enabled patient health monitoring in smart homes," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1789–1796, June 2018.

[10] M. J. Baucas and P. Spachos, "Using cloud and fog computing for large scale iot-based urban sound classification," *Simulation Modelling Practice and Theory*, vol. 101, pp. 102013, 2020, Modeling and Simulation of Fog Computing.

[11] M. J. Baucas and P. Spachos, "A scalable iot-fog framework for urban sound sensing," *Computer Communications*, vol. 153, pp. 302 – 310, 2020.

[12] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for iot systems: A computation offloading game," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3246–3257, 2018.

[13] M. Al Disi, H. Djelouat, C. Kotroni, E. Politis, A. Amira, F. Bensaali, G. Dimitrakopoulos, and G. Alinier, "Ecg signal reconstruction on the iot-gateway and efficacy of compressive sensing under real-time constraints," *IEEE Access*, vol. 6, pp. 69130–69140, 2018.

[14] A. Raman, "Enforcing privacy through security in remote patient monitoring ecosystems," in *2007 6th International Special Topic Conference on Information Technology Applications in Biomedicine*, Nov 2007, pp. 298–301.

[15] Q. Shen, X. Liang, X. Shen, X. Lin, and H. Y. Luo, "Exploiting geo-distributed clouds for a e-health monitoring system with minimum service delay and privacy preservation," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 2, pp. 430–439, March 2014.

[16] A. Zhang, L. Wang, X. Ye, and X. Lin, "Light-weight and robust security-aware d2d-assist data transmission protocol for mobile-health systems," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 3, pp. 662–675, March 2017.

[17] S. Majumder, L. Chen, O. Marinov, C. Chen, T. Mondal, and M. J. Deen, "Noncontact wearable wireless ecg systems for long-term monitoring," *IEEE Reviews in Biomedical Engineering*, vol. 11, pp. 306–321, 2018.

[18] A. Ara, M. Al-Rodhaan, Y. Tian, and A. Al-Dhelaan, "A secure privacy-preserving data aggregation scheme based on bilinear elgamal cryptosystem for remote health monitoring systems," *IEEE Access*, vol. 5, pp. 12601–12617, 2017.

[19] S. Park, Y. Jeong, M. S. Kim, and H. S. Kim, "Linear prediction-based dereverberation with very deep convolutional neural networks for reverberant speech recognition," in *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, Jan. 2018, pp. 1–2.

[20] A. Incze, H. Jancs, Z. Szilgyi, A. Farkas, and C. Sulyok, "Bird sound recognition using a convolutional neural network," in *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*, Sep. 2018, pp. 000295–000300.

Authorized licensed use limited to: University of Guelph. Downloaded on February 02,2021 at 22:06:48 UTC from IEEE Xplore. Restrictions apply.