

Empowering Healthcare IoT Systems with Hierarchical Edge-based Deep Learning

Iman Azimi
University of Turku
Turku, Finland
imaazi@utu.fi

Janne Takalo-Mattila
VTT Technical Research
Centre of Finland
Finland
janne.takalo-mattila@vtt.fi

Arman Anzanpour
University of Turku
Turku, Finland
armanz@utu.fi

Amir M. Rahmani
University of California Irvine
Irvine, USA
TU Wien
Vienna, Austria
amirr1@uci.edu

Juha-Pekka Soininen
VTT Technical Research
Centre of Finland
Finland
juha-pekka.soininen@vtt.fi

Pasi Liljeberg
University of Turku
Turku, Finland
pakrli@utu.fi

ABSTRACT

Remote health monitoring is a powerful tool to provide preventive care and early intervention for populations-at-risk. Such monitoring systems are becoming available nowadays due to recent advancements in Internet-of-Things (IoT) paradigms, enabling ubiquitous monitoring. These systems require a high level of quality in attributes such as availability and accuracy due to patients critical conditions in the monitoring. Deep learning methods are very promising in such health applications to obtain a satisfactory performance, where a considerable amount of data is available. These methods are perfectly positioned in the cloud servers in a centralized cloud-based IoT system. However, the response time and availability of these systems highly depend on the quality of Internet connection. On the other hand, smart gateway devices are unable to implement deep learning methods (such as training models) due to their limited computational capacities. In our previous work, we proposed a hierarchical computing architecture (HiCH), where both edge and cloud computing resources were efficiently exploited, allocating heavy tasks of a conventional machine learning method to the cloud servers and outsourcing the hypothesis function to the edge. Due to this local decision making, the availability of the system was highly improved. In this paper, we investigate the feasibility of deploying the Convolutional Neural Network (CNN) based classification model as an example of deep learning methods in this architecture. Therefore, the system benefits from the features of the HiCH and the CNN, ensuring a high-level availability and accuracy. We demonstrate a real-time health monitoring for a case study on ECG classifications and evaluate the performance of the system in terms of response time and accuracy.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://www.acm.org/permissions).

CHASE '18, September 26–28, 2018, Washington, DC, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5958-0/18/09...\$15.00

<https://doi.org/10.1145/3278576.3278597>

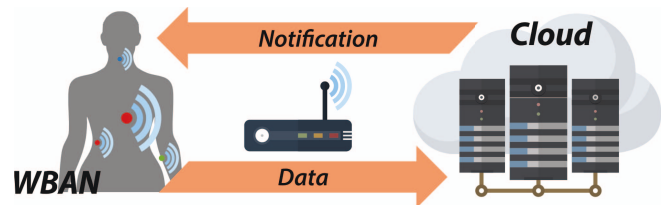


Figure 1: A three-tier IoT-based health monitoring system

KEYWORDS

Internet of Things, Hierarchical Computing, Health Monitoring, Deep Learning, Convolutional Neural Networks, Electrocardiogram

1 INTRODUCTION

Internet of Things (IoT) is increasingly growing in healthcare systems, where patients with acute diseases and at-risk populations such as senior adults and pregnant women can be continuously monitored. Such IoT-based applications are promising alternatives to traditional health services, extending the boundaries of healthcare outside of hospital settings [3, 15]. They mostly target early-detection and prevention of patients' health deterioration as well as allowing independent living of the patients [5].

These systems can be conventionally partitioned into three main tiers in the context of IoT to deliver health monitoring applications [2]. The three tiers are illustrated in Figure 1. First, a wireless body area network (WBAN) including wearable bio-sensors acquires health data. In real-world applications, such data acquisition is mostly performed 24/7 via heterogeneous sensors by which a massive volume of data (i.e., big data) [6, 21] is generated over time. Second, continuous connectivity is enabled via a gateway device located in the vicinity of the WBAN (i.e., edge). The gateway device traditionally operates as a relay between the WBAN and servers although supplementary services can be allocated to the edge. Third, a cloud server is responsible for continuous data analysis methods, enabling real-time decision making.

A wide range of machine learning algorithms is utilized for decision makings in healthcare applications [1, 25]. However, most

of the conventional methods such as traditional neural networks and k-nearest neighbors are inapplicable when the scale of data increases over time, and large amounts of data as big data are generated [32]. In contrast, deep learning methods are promising alternatives in this regard, using strategies in deep architectures to learn hierarchical representations [27, 33]. Such methods can manage large amounts of data while the accuracy improves with the increase of training datasets. Convolutional neural networks (CNN) is one example of the deep learning methods, considered in this work for the IoT-based health monitoring.

In a cloud-based IoT architecture (Figure 1) [11, 12, 23], deep learning methods are perfectly positioned in the cloud servers to take advantages of high-end machines. These machines provide a satisfactory performance with considerably low execution time. However, the response time of the system heavily depends on the availability and quality of Internet connection. Obviously, these systems are unable to satisfy latency-critical applications (e.g. health monitoring), as they have serious consequences for the patients in emergency situations, to the detriment of a delay in establishing a connection.

Moreover, exploiting smart gateway devices at the edge is recently proposed for health monitoring [7, 28, 29]. In this regard, the roles of the gateway devices are extended to implement data processing, through which the collected data is analyzed locally [8]. The gateway devices are equipped with limited computational resources, so a smart task allocation is required to fulfill health monitoring requirements. However, deep learning methods cannot be fully performed on the edge devices, as they are highly expensive in terms of computation time.

Another alternative is a hierarchical computing architecture, in which both local and remote computing resources of the IoT-based system are efficiently exploited. In our previous work [4], we proposed such a hierarchical architecture for a health monitoring system named as HiCH, partitioning a linear machine learning method (i.e., support vector machine with a linear kernel) into different computing components distributed in the three-layers IoT system. The HiCH architecture could utilize the benefits of both edge and cloud computing, where a high level of availability was obtained due to local decision making as well as preserving the performance of the learning algorithm.

In this paper, which is an extension of our previous work presented in [4], we investigate the feasibility of deploying deep learning as a nonlinear machine learning algorithm in the HiCH architecture. The successful integration of deep learning in the HiCH architecture enables health monitoring systems to offer a high level of availability and accuracy. In summary, our main contributions in this work are as follow.

- We present that HiCH is capable of fully employing the Convolutional Neural Networks (CNN)-based machine learning model [31] to perform a real-time heart-related disease detection.
- We demonstrate a real-time health monitoring system implementation for a case study and evaluate the response time of the system.

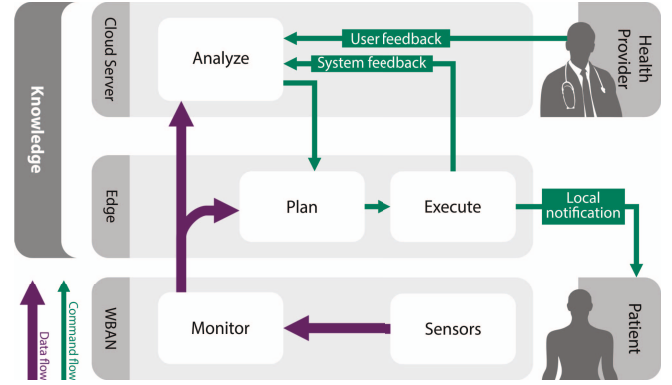


Figure 2: The hierarchical IoT-based architecture

- We evaluate the accuracy of the classification and indicate how the collected data throughout the monitoring can influence the accuracy.

The rest of the paper is organized as follow. In Section 2, we outline a short background of deep learning. Section 3 presents the hierarchical architecture. The demonstration and performance of the proposed system are indicated in Section 4. Finally, Section 5 concludes the paper.

2 DEEP LEARNING

Deep learning is one subset of machine learning algorithms that are being used recently in various fields. It has been demonstrated to outperform traditional methods in speech recognition, visual object recognition, and object detection. Deep learning models consist of multiple processing layers that are capable of learning meaningful features of the raw data without domain-level expertise. On the contrary, conventional machine learning methods typically require a considerable amount of domain-level expertise to first extract features and then perform the classification [22].

Convolutional neural networks (CNN) are a class of deep neural networks which are often used with two-dimensional signals such as videos and images. They can learn thousands of objects using millions of images as input datasets. Learning capacity of the CNN can be controlled by varying the depth and breadth of the model [20]. In addition to the two-dimensional signals, CNNs can be exploited with one-dimensional signals such as electrocardiography (ECG) or audio signals.

A typical architecture of CNN for image recognition is formed by stacking multiple layers of computing units with different roles [30]. The main unit in CNN architecture is the convolutional layer that contains learnable filter banks activating when specific features are detected. Max pooling layers leverage CNN architecture to reduce the amount of parameters and enable over-fitting. Fully connected layers typically follow the series of convolutional and max-pooling layers. Role of these layers acts as a classifier for the learned features.

3 HIERARCHICAL COMPUTING ARCHITECTURE

In this section, we outline the HiCH as a hierarchical computing architecture enabled by the Convolutional Neural Networks (CNN) to perform real-time heart-related diseases detection using ECG signals. The HiCH exploits the capabilities of edge and cloud computing paradigms, allocating heavy computation tasks of the classification algorithm to the cloud and outsourcing the decision making task (i.e., classifier) to the edge. Therefore, the availability of the IoT-based application is significantly improved, due to local decision making in the case of degraded Internet access or connection loss. Moreover, the performance (e.g., accuracy) of the learning algorithm is preserved in this hierarchical architecture as well as its performance in a fully-centralized computing core in the cloud.

The HiCH architecture employs an enhanced version of the MAPE-K model introduced by IBM [17], distributing the computations in the three-layers IoT system. The model includes 5 different computing components named as *Monitor*, *Analyze*, *Plan*, *Execute* and *System Management*. In this following, we only exploit the first 4 components of this model, as the *System Management* is responsible for managing the system configurations and is out of the scope of this paper. For more details see [4]. Figure 2 illustrates the proposed architecture enabled by the computing components, each of which shares the system knowledge.

3.1 Monitor

The *monitor* is a bridge between the sensors and other computing components, located in the WBAN. It includes a local processing unit for analog to digital conversion, pre-processing methods (e.g., data filtering and compression) and data aggregation in a local data storage. The stored data are periodically transmitted to the edge. The transmission time is determined according to the data and application, which is 10 seconds of the ECG signal in our case study.

3.2 Analyze

Heavy computation tasks including training the hypothesis function (i.e., classifier) are allocated to the *Analyze* that are fully positioned in the cloud machines. As indicated in Figure 3, the hypothesis function is generated using collected data and feedback. The *Training Data* performs required data processing methods before feeding the data to the *Classification Algorithm*.

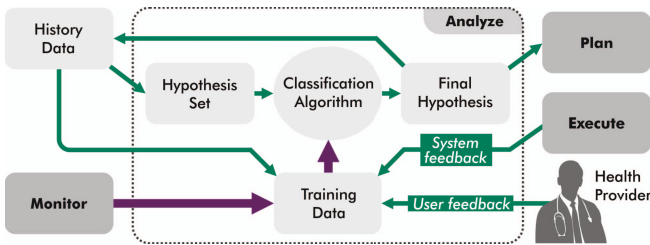


Figure 3: The *Analyze* component in the cloud

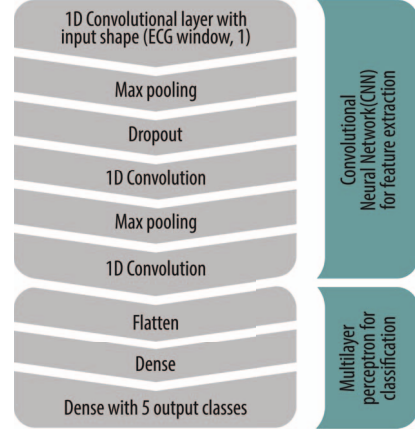


Figure 4: Overall structure of CNN and MLP

In our case study, the *Training Data* is responsible for ECG cycles (i.e., heartbeats) extraction from the incoming ECG signals. Moreover, a fully automatic deep neural network based classifier [31] is employed as the *Classification Algorithm* to detect and classify different abnormalities in ECG signals. In addition, a pre-trained model (i.e., *Hypothesis Set* in the Figure 3) is exploited from ECG datasets. At the beginning of the monitoring, this model acts as a baseline in clinical trials although it is periodically updated over time when new data and feedback are collected.

In the *Classification Algorithm*, first, the meaningful features are automatically determined by leveraging a three-layers CNN. In this method, 16, 32 and 64 neurons are selected as the first, second and third CNN layers, respectively. Moreover, pool size of max pooling is set as 4, and 20% dropout rate is determined to prevent overfitting. The rectified linear unit (ReLU) is used as an activation function [16] in convolutional layers.

Second, the Multilayer Perceptron (MLP) is utilized to implement the classification, using the extracted features from the CNN layers. In this method, one hidden layer with 128 neurons is selected with a learning rate equals to 0.001. Moreover, Tanh and Adam [19] functions are utilized as the activation function and optimization algorithm, respectively. A High-level structure of the classification algorithm including CNN and MLP is shown in Figure 4.

3.3 Plan

The classifier is periodically sent to the *Plan* located at the edge, providing local decision making. Such periodical updates of the classifier enable personalization in the decision making, considering

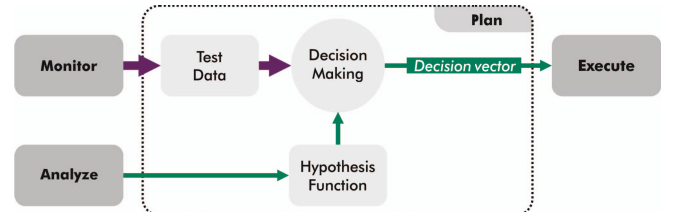


Figure 5: The *Plan* component at the edge

incoming data in re-training of the classifier (at the *Analyze* component) throughout the monitoring. As illustrated in Figure 5, the streaming data received from the *Monitor* component are classified, and a decision vector is generated. Note that similar to the *Analyze* component, required data processing methods (e.g., heartbeat extraction from ECG signals, filtering, and normalization) are fulfilled in the *Test Data*. The decision vector as the output contains the decision class (e.g., patient's health status). It is sent to the *Execute* component for further actuation.

3.4 Execute

Execute is the second computing component at the edge, implementing the actuation of the system. It sends notifications to the users when an abnormality is detected in the *Plan*. Moreover, it forwards system feedback to the *Analyze*, improving the classifier in terms of accuracy. For example, the model is improved over time by sending the estimated decision class and the true label of data reported by the patient and health provider.

4 IMPLEMENTATION AND EVALUATION

We demonstrate the proposed architecture empowered by the CNN via a health case study on ECG classification. In this regard, the decision making is implemented at the edge, sending notifications to the user in case of disease detection. We, first, evaluate response time and availability of the HiCH in comparison with a conventional IoT-based system where the computations are fully performed in the cloud server. Then, we assess the accuracy of the HiCH, indicating the accuracy of decision making at the beginning of the monitoring and its improvement throughout the monitoring.

4.1 Setup

We emulate a sensor node and use the MIT arrhythmia database available at [14, 24, 26] to train and test the classification algorithm. The sensor node emulator is an ESP8266-12E WiFi module which contains an 80MHz 32-Bits RISC microprocessor with 96KB RAM and 4MB QSPI flash memory. The Wifi module connects to a local WiFi network and the microprocessor is able to read a Micro SD card via SPI communication. The ECG data is stored on the Micro SD card. We program the sensor node to read 3600 ECG samples from a file on Micro SD card during a 10-second period and send them via an *upload POST* request to the edge device.

The MIT Arrhythmia database includes totally 48 separate ECG recordings, and the length of each recording is 30 minutes. ECG in this database is stored using a two-lead configuration using 360 Hz sampling rate and digitized with 11-bit resolution. Originally, heartbeats in the database are labeled by two cardiologists. 19 different labels have been used in classifying arrhythmias. However, AAMI [13] recommends that these classes can be divided into five super-classes, namely normal (N), supraventricular ectopic beat (SVEB), ventricular ectopic beat (VEB), fusion beat (F) and unknown beat (Q). These types of arrhythmias are not immediately life-threatening, but still may demand further investigation. Arrhythmias that belongs to this category can be detected from a single heartbeat, which means that shape and other morphological features define the type of the arrhythmia [18].

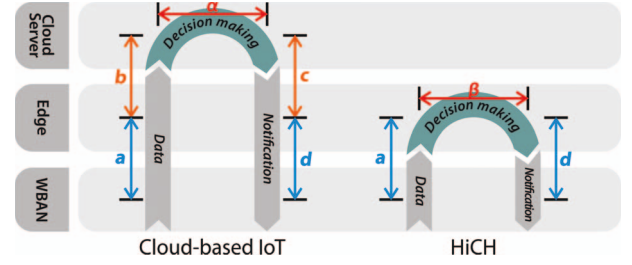


Figure 6: Response time in the conventional IoT-based system and the HiCH

The edge device is an Ubuntu Linux machine running Apache web server, PHP, and Python interpreter services. A PHP script receives the samples file from the sensor node and calls the Python codes for signal processing and decision making. The reply to the sensor node *upload POST* request contains the result of decision making.

To perform the comparison between the HiCH and the conventional IoT-based system, we implement a similar procedure on the cloud machine which is a virtual private server (VPS) with the same OS and services. The VPS runs on two E5-2680 v3 Intel Xeon CPUs at 2.50GHz with 4GB RAM and 40Gbps network.

4.2 Response Time

In this section, we assess the performance of the HiCH focusing on the system's response time which is the time period between recording data and notifying the patient in case of emergency. The response time can be divided into different intervals as:

- a = Data transmission time, WBAN to edge
- b = Data transmission time, edge to cloud
- c = Notification transmission time, cloud to edge
- d = Notification transmission time, edge to WBAN (patient)
- α = Execution time, in the cloud
- β = Execution time, at the edge

Table 1: Data transmission time using HiCH and the cloud-based IoT with different networks

		Data trans. rate (Kbps)	Trans. time between WBAN and edge (a+d) (ms)	Trans. time between edge and cloud (b+c) (ms)
Cloud-based IoT	HiCH Local Network	30000	10	-
	4G	4000	10	41
	Fast 3G	1500	10	151
	3G	750	10	450
	Fast 2G	450	10	753
	2G	250	10	1490
	GPRS	50	10	5803

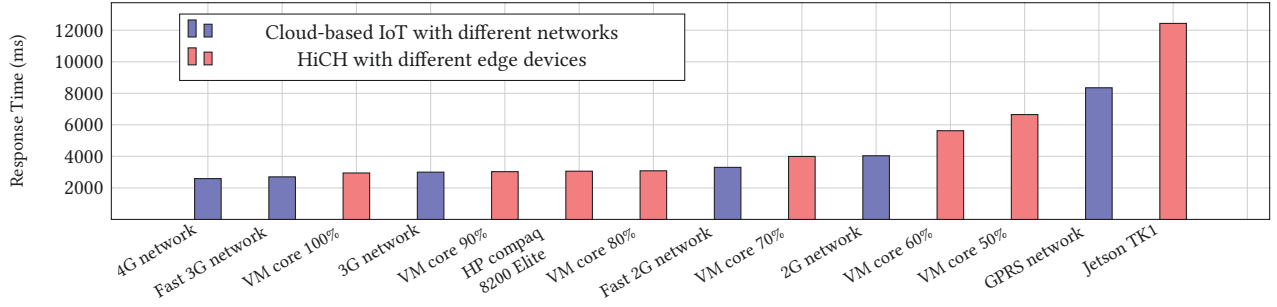


Figure 7: Response time for different approaches using the Cloud-based IoT and HiCH architectures

As indicated in Figure 6, the HiCH response time includes $a + \beta + d$ while the response time of the cloud-based IoT system is $a + b + \alpha + c + d$.

To validate the experiments, we measure the intervals for the two systems using different setups. In this regard, the transmission time is measured via 4G, Fast 3G, 3G, Fast 2G, 2G and GPRS networks. The average transmission time for the two system is represented in Table 1. $a + d$ as the local transmission equals to 10ms although $b + c$ varies from 41ms to 5803ms depending on the Internet network.

We measure the execution time of the decision-making process (i.e., α and β) using different edge devices with different CPU performance. In this regard, we utilize an HP Compaq 8200 Elite Linux machine with a quad-core Intel Core i3 CPU at 3.10 GHz and an NVIDIA Jetson-TK1 with a quad-core ARM Cortex A15 CPU at 2.33 GHz. Moreover, we use an Oracle Virtual Machine with a single-core Intel Core i7 CPU at 3.4 GHz and allocate 100%, 90%, 80%, 70%, 60%, 50% of its execution capacity to the computation in each experiment. As the decision-making algorithm in this research is in Python, we measure the CPU performance by counting the number of floating point operations performed per second by Python interpreter (FLOPS). Table 2 indicates the Python FLOPS and the average execution time for the two systems.

In conclusion, the response time of the two systems with different setups is illustrated in Figure 7. The response time of the cloud-based IoT system highly depends on the Internet network.

Table 2: Execution time of the decision making process using HiCH and the cloud-based IoT with different devices

		Execution time (ms)	Python FLOPS
HiCH	VM* Core, 100% Exe. Capacity	2936	17.5
	VM Core, 90% Exe. Capacity	3020	13.8
	HP Compaq 8200 Elite	3049	13.4
	VM Core, 80% Exe. Capacity	3074	13.1
	VM Core, 70% Exe. Capacity	3985	11.1
	VM Core, 60% Exe. Capacity	5617	7.9
	VM Core, 50% Exe. Capacity	6643	7
	Jetson TK1	12425	4.3
Cloud-based IoT	Cloud Server	2539	13

*Virtual Machine

As indicated, the response time of this system with the 4G network is the lowest although it increases when the Internet connection is poor. On the other hand, the response time of the HiCH is determined by the processing power of the edge device. Therefore, by selecting an appropriate edge device, HiCH ensures an acceptable response time.

4.3 Accuracy Assessment

We evaluate the accuracy of the ECG arrhythmia classification using the MIT Arrhythmia dataset. In this regard, we divide the dataset into two different datasets (DS1 and DS2) using the division method presented in [9]. Dataset division, where training and testing dataset are generated from separate patients, is called inter-patient paradigm. On the contrary, dataset division where testing and training phase data contains heartbeats from the same patients is called intra-patient paradigm. We ensure unbiased classification accuracy by the inter-patient paradigm, considering patient-specific variances in the data.

In the first step, the ECG classifier is trained by the DS1 dataset, which contains 51020 ECG samples from different patients (i.e., inter-patient). We validate the performance of the classifier using the DS2 dataset. The confusion matrix is specified as Table 3, using the estimated decision for ECG samples and the true labels.

The correct estimates, highlighted in the confusion matrix, are notably high in this initial phase. However, the accuracy might be insufficient particularly for clinical applications as the classifier is trained via general data and inter-patient variation of ECG morphologies is considerably large [10]. Therefore, the model is not specifically trained for the monitored patient.

To address this issue, the accuracy of the classifier is improved over time in our proposed architecture by re-training the classifier via incoming ECG samples from the patient along with labels from

Table 3: Confusion matrix of the classification algorithm

		Estimated Decision					
		Normal	SVEB	VEB	F	Q	
True Decision	Normal	40671	905	2615	68	0	
	SVEB	642	1148	47	0	0	
	VEB	339	2	2874	6	0	
	F	275	0	111	2	0	
	Q	2	0	5	0	0	

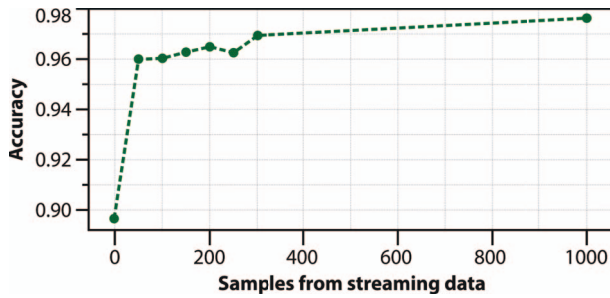


Figure 8: Accuracy improvement by retraining the model with new samples

the health provider (i.e., patient-specific information from ECG morphology). We test the performance of the classifier by randomly selecting 50, 100, 150, 200, 250, 300 and 1000 samples from the DS2 dataset and re-training the initial classification model.

Figure 8 illustrates how the accuracy of the model improves when the model is re-trained. The starting point represents the initial pre-trained model with no knowledge about patient-specific data. As indicated, the accuracy of the classifier at the starting point is less than 0.9. In contrast, the accuracy significantly increases to over 0.96 even if the classifier is re-trained with a small portion (e.g., 50 samples) of the patient data (i.e., intra-patient) throughout the monitoring.

5 CONCLUSION

IoT-based health monitoring systems enable at-risk patients to be monitored outside of conventional clinical settings. Such systems are demanded to deliver a high quality of experience as a defect in the service may lead to fatal consequences for the patients. In terms of decision making, deep learning can provide a satisfactory performance as a massive amount of data can be fed to the classification algorithm. In the conventional cloud-based IoT systems, these methods can be fully implemented in the cloud machines. However, these systems are insufficient for a time-sensitive health application due to the dependency of the service to the quality of the Internet connection. Fully distributed edge-based systems are other alternatives although they are incapable of implementing deep learning methods due to the restricted processing power. Another alternative, proposed in our previous work, is the hierarchical computing architecture to partition the learning method in the cloud and edge. In this paper, we investigated the feasibility of empowering the HiCH architecture with the CNN algorithm. We compared the response time of the HiCH with a conventional cloud-based IoT system and indicated that HiCH could ensure an acceptable response time and improve the availability particularly when the connection is poor. In addition, we assessed the accuracy of the system and showed that the accuracy was improved throughout the monitoring, feeding the streaming data to the classification algorithm.

ACKNOWLEDGMENT

This work was partially supported by the US National Science Foundation (NSF) WiFiUS grant CNS-1702950 and Academy of Finland grants 311764 and 311765.

REFERENCES

- [1] G. Acampora et al. 2013. A Survey on Ambient Intelligence in Healthcare. *Proc. IEEE* 101, 12 (2013), 2470–94.
- [2] A. Al-Fuqaha et al. 2015. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surveys & Tuts* 17, 4 (2015), 2347–76.
- [3] L. Atzori et al. 2010. The Internet of Things: A survey. *Computer Networks* 54, 15 (2010), 2787–805.
- [4] I. Azimi et al. 2017. HiCH: Hierarchical Fog-Assisted Computing Architecture for Healthcare IoT. *ACM Trans. Embed. Comput. Syst.* 16, 5s (2017), 174:1–174:20.
- [5] I. Azimi et al. 2017. Internet of things for remote elderly monitoring: a study from user-centered perspective. *Journal of Ambient Intelligence and Humanized Computing* 8, 2 (2017), 273–89.
- [6] M. Beyer. 2017. Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data. www.gartner.com/newsroom/id/1731916.
- [7] F. Bonomi et al. 2012. Fog computing and its role in the internet of things. *MCC'12* (2012), 13–16.
- [8] D. Borthakur et al. 2017. Smart fog: Fog computing framework for unsupervised clustering analytics in wearable internet of things. In *GlobalSIP*.
- [9] P. de Chazal et al. 2004. Automatic classification of heartbeats using ECG morphology and heartbeat interval features. *IEEE Transactions on Biomedical Engineering* 51, 7 (2004), 1196–1206. <https://doi.org/10.1109/TBME.2004.827359>
- [10] P. de Chazal and R. B. Reilly. 2006. A patient-adapting heartbeat classifier using ECG morphology and heartbeat interval features. *IEEE Transactions on Biomedical Engineering* 53, 12 (2006), 2535–43.
- [11] A. Dohr et al. 2010. The Internet of Things for Ambient Assisted Living. In *ITNG*.
- [12] M. Fazio et al. 2015. Exploiting the FIWARE Cloud Platform to Develop a Remote Patient Monitoring System. In *ISCC*.
- [13] Association for the Advancement of Medical Instrumentation and American National Standards Institute. 1999. *Testing and Reporting Performance Results of Cardiac Rhythm and ST-segment Measurement Algorithms*. The Association. <https://books.google.fi/books?id=gzPdtgAACAAJ>
- [14] A.L. Goldberger et al. 2000. Components of a New Research Resource for Complex Physiologic Signals. *Circulation* 101, 23 (2000), e215–e220.
- [15] J. Gubbi et al. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 29, 7 (2013), 1645–60.
- [16] R. H. R. Hahnloser and H. S. Seung. 2001. Permitted and Forbidden Sets in Symmetric Threshold-Linear Networks. In *Advances in Neural Information Processing Systems*. MIT Press, 217–223.
- [17] IBM Corporation. 2006. An architectural blueprint for autonomic computing. *White paper* (2006).
- [18] S. Karimifard et al. 2006. Morphological Heart Arrhythmia Detection Using Hermitian Basis Functions and kNN Classifier. In *Int. Conf. of the IEEE Engineering in Medicine and Biology Society*. 1367–1370. <https://doi.org/10.1109/IEMBS.2006.260182>
- [19] D. P. Kingma and J. Ba. 2014. Adam: A method for stochastic optimization. *CoRR abs/1412.6980* (2014).
- [20] A. Krizhevsky et al. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *25th Int. Conf. on Neural Information Processing Systems*. Curran Associates, Inc., 1097–1105.
- [21] D. Laney. 2001. *3D Data Management: Controlling Data Volume, Velocity, and Variety*. Technical Report. META Group Inc.
- [22] Y. LeCun et al. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–44.
- [23] J. Mohammed et al. 2014. Internet of Things: Remote Patient Monitoring Using Web Services and Cloud Computing. In *CPSCom*.
- [24] G. B. Moody and R. G. Mark. 2001. The impact of the MIT-BIH Arrhythmia Database. *IEEE Eng in Med and Biol* 20, 3 (May 2001), 45–50. <https://doi.org/10.1109/51.932724>
- [25] K. P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.
- [26] PhysioNet. 2018. MIT-BIH Arrhythmia Database. <https://www.physionet.org/physiobank/database/mitdb/>.
- [27] J. Qiu et al. 2016. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing* 2016, 1 (2016), 67.
- [28] A. M. Rahmani et al. 2017. *Fog Computing in the Internet of Things Intelligence at the Edge*. Springer.
- [29] A. M. Rahmani et al. 2018. Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach. *Future Generation Computer Systems* 78 (2018), 641–658.
- [30] K. Simonyan and A. Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR abs/1409.1556* (2014).
- [31] J. Takalo-Mattila et al. 2018. Inter-patient ECG Classification Using Deep Convolutional Neural Networks. In *Euromicro Conference on Digital System Design, special track on Applications, Architectures, Methods and Tools for Machine- and Deep Learning*. in press.
- [32] O. Y. Al-Jarrah et al. 2015. Efficient Machine Learning for Big Data: A Review. *Big Data Research* 2, 3 (2015), 87–93.
- [33] Q. Zhang et al. 2018. A survey on deep learning for big data. *Information Fusion* 42 (2018), 146–157.