A Hidden Markov Model Based Dynamic Scheduling Approach for Mobile Cloud Telemonitoring

Xiaoliang Wang*, Wenyao Xu† and Zhanpeng Jin*
*Department of Electrical and Computer Engineering, State University of New York at Binghamton,
Binghamton, NY 13902-6000, USA, Email: {xwang90, zjin}@binghamton.edu
†Department of Computer Science and Engineering, State University of New York at Buffalo,
Buffalo, NY 14260-2500, USA, Email: wenyaoxu@buffalo.edu

***Abstract*—Recent advances in mobile and cloud technologies have been proved to be a promising way to provide healthcare, particularly health monitoring, to individuals in a cost-effective, user-friendly, and pervasive way. However, in practical use, multiple objectives usually need to be considered and fulfilled when deploying such a mobile-cloud-based telemonitoring platform, such as processing latency, energy consumption, and diagnosis accuracy. Given the ever-changing clinical priorities, personal demands, and environmental conditions, it is imperative to explore a smart scheduling and management approach capable of dynamically adjusting the offloading strategy on this mobile-cloud infrastructure. In this study, we propose a new Hidden Markov Model (HMM) based dynamic scheduling approach to allow the system to adapt to the changing requirements.**

I. Introduction

Mobile health (also known as "mHealth"), which utilizes the advanced mobile technologies together with wireless body sensors for healthcare monitoring and diagnosis, has been significantly advocated in the recent years. Mobile devices, however, due to their limited storage space and computational capability, might not be able to perform tasks which require intensive computing and extensive storage resources [1]. On the other side, cloud computing, because of its unique features like elasticity and scalability in both computation ability and storage space [2], when combined with mobile computing, would be able to provide ubiquitous and personalized healthcare to patients [3].

Mobile cloud computing has been proved to be a promising way for next-generation, individual-centered, pervasive healthcare. Through offloading part of tasks from the mobile to the cloud, a significant portion of energy can be saved to extend the mobile battery life. Meanwhile, the diagnostic capability could be improved by executing more sophisticated algorithms on the cloud. In this way, the personal mobile device could sustain much longer to meet the increasing demands for day-long, continuous health monitoring. Recently, many studies present different strategies to achieve more energy saving and performance improvements through mobile cloud offloading, in an online dynamic manner [4], [5], [6], [7]. Barbera et al. [8] gave an evaluation towards the impact of network bandwidth on the cost of both mobile computation offloading and mobile software/data backups in terms of traffic overhead from mobile to cloud and mobile energy consumption. Cheng et al. [9] developed a genetic algorithm based strategy to seek the optimal solution for code offloading with a just-in-time objective. Ragona et al. [10] established a mathematic model of energy consumption and time cost for wearable and mobile devices in different offloading scenarios. Chen [11] proposed a game theory based decentralized dynamic offloading strategy to generate optimized schemes for multi-user offloading. The tradeoff among multiple optimization goal in mobile cloud offloading was considered in some other studies [12], [13].

For health telemonitoring, superior processing performance (i.e., low latency), high accuracy and low power consumption (i.e., longer battery life of mobile devices) are the common requirements in practical use. Unfortunately, all those goals can hardly be achieved simultaneously. For example, machine learning algorithms, through its well-established supervised training procedures, have demonstrated their superiority in providing more accurate diagonosis results over simple rule-based algorithms. Nevertheless, due to its non-trivial computational complexity, when real-time processing and lower latency are demanded for those urgent medical conditions, machine learning based algorithms may no longer be the best choice, compared with more computationally efficient rule-based options. Similarly, when offloading computing tasks from the mobile device to the cloud for extending the battery life of the mobile, a situation-aware, dynamic, and online offloading strategy is demanded to adapt to the ever-changing network conditions and thus achieve an optimal trade-off between the performance latency and power consumption.

In this study, we propose a smart, dynamic scheduling approach using hidden markov model (HMM) for synergistically optimizing the mobile-cloud-based telemedicine applications towards multiple objectives: high accuracy, low latency, and long battery life. Specifically, by taking into the consideration of multiple factors including severity of clinical conditions, network conditions, and battery level of mobile devices, this HMM-based approach can dynamically prioritize those objectives and adjust the processing procedures.

II. Problem Formulation and Method

Based on our previously proposed offloading approach specifically designed for machine learning techniques which involve computation-intensive training processes [14], we

978-1-5090-4179-4/17/$31.00 ©2017 IEEE

would like to extend this offloading strategy into generalized processing algorithms and network conditions. We define a health monitoring algorithm as a set of basic functional blocks (BFBs). Each BFB consists of various inputs (required knowledge) and outputs (outcomes). Given a mobile application A, its call function graph is $G = \{V, E\}$, where each vertex $v \in V$ denotes a BFB in A and an invocation from u to v thereby is denoted by an edge $e = (u, v)$. We reconstruct a new graph $G' = \{V', E'\}$ by applying offloading methods onto V, where $v' \in V'$ represent BFBs being offloaded through $e' = (u, v')$. The execution time of each BFB can be annotated as T_v and $T_{v'}$ in the mobile and cloud respectively. The energy consumed by the mobile system is thus denoted as: P_c for computing, P_i while being idle, and P_{tr} for sending and receiving data. Furthermore, we will consider multiple wireless network interface scenarios (such as WiFi, WiMAX, and 2G/3G/4G LTE) where the mobile device is connected via the most fast and reliable channel when more than one is available. Though both the type of network connection and the level of signal strength will influence the data transmission bandwidth, we will only focus on the actual data transmission rate for the sake of simplicity and denote it as D_n. The size of data to be transferred for BFB v' is given by $n_{v'}$.

More specifically, for example, if we want the implemented system to have the optimal performance, we could choose solution to equations below as resolution strategy.

$$min_{v,v'} \sum_{v \in V} T_v + \sum_{v' \in V'} (\frac{n_{v'}}{D_n} + T_{v'}),$$
$$T_v \propto O(n), T_v \geq 0, T_{v'} \geq 0, n_{v'} \geq 0, D_n \geq 0 \quad (1)$$

The execution time of a certain BFB v is proportional to its algorithm complexity, which is denoted as $O(n)$.

According to the equation (1), we could find that the processing performance relies largely on the type of algorithms used by BFB, since different algorithms possess different $O(n)$. Also, the size of data to be transferred due to offloading for BFB v', which is given by $n_{v'}$ and different data transmission rate D_n have impacts on the processing performance.

If we want the implemented system to be most energy efficient, we would like to solve the equations below:

$$min_{v,v'} \sum_{v \in V} P_c \times v + \sum_{v' \in V'} P_{tr} \times \frac{n_{v'}}{D_n} + \sum_{v' \in V'} P_i \times v',$$
$$T_v \geq 0, T_{v'} \geq 0, n_{v'} \geq 0, D_n \geq 0 \quad (2)$$

where the optimization goal is to minimize the overall energy consumption of the mobile device, including the actual computing portion, the data transmission portion, and the idle portion. According to the equation, we could find that the energy saving can be achieved by decreasing the size of data to be transferred due to offloading for BFB v', which is given by $n_{v'}$. Alternatively, we could find that the energy cost is also influenced by different data transmission rate D_n.

We rely on the Hidden Markov Model (HMM) to perform optimal parameters tuning procedures. As explained in Figure 1, $S1, S2, S3$ represent the hidden states; $Y1, Y2, Y3, Y4$ represent observed outputs in each hidden state. a_{ij} represents

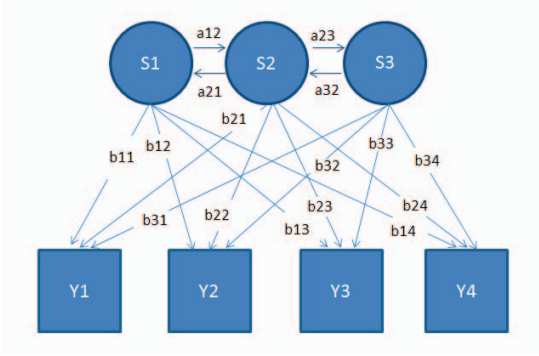

Fig. 1. Illustration of Hidden Markov Model

transition probabilities between hidden states and b_{ij} represents the output probabilities for each hidden state. We define different network situations, mobile device battery drain status, and CPU load states as the hidden states. The mobile energy efficiency, processing accuracy and execution performance could be observed as outputs from the hidden states. We used the Baum-Welch algorithm for HMM learning process. In our implementation, we would like to get a HMM for each operation setting (e.g., a fixed sampling rate, a selected mobile cloud offloading strategy, etc.). Through sorting output probability of hidden states, we could make optimal choice according to the statistical features.

$$P(\boldsymbol{S}, \boldsymbol{Y}) = P(S_1)P(Y_1|S_1)\prod_{t=2}^{T} P(S_t|S_{t-1})P(Y_t|S_t) \quad (3)$$

III. Mobile Cloud Based Health Telemonitoring

Electrocardiogram (ECG) monitoring represents an important, useful but usually expensive component of current cardiac disease detection and elderly care [15]. ECG processing could be generally divided into several stages: signal preprocessing, feature extraction, heartbeat classification, and diagnostic decision making. Similar to other physiological signal monitoring and analysis deployed on mobile cloud computing, ECG processing also have those requirements: higher diagnosis accuracy, lower processing latency and longer mobile battery life. Specifically, we may consider the following options to optimize the system: tuning the ECG sampling rate, employing different processing algorithms, and utilizing different offloading strategies. The external influence could be the changing network conditions (different transmission speeds).

a) Processing Algorithms: An artificial neural network (ANN), consisting of a hidden layer of 30 neurons, was developed as the heartbeat classifier. 41 raw ECG samples, including 14 points before the fiducial mark and 26 points after, were fed into the ANN as the inputs for each heartbeat [16]. To preserve generality, the training process categorized three types of heartbeats: normal, premature ventricular contractions, and other beats. Therefore, the implemented ANN classifier contains 41 inputs and 3 outputs. The other algorithm we adopted was a rule-based one, which relied on some heartbeat feature comparisons to classify heartbeats.

b) Offloading Strategies: In this study we consider three offloading strategies. The first one is to offload the entire

processing to the cloud, called "cloud-only". For machine learning method, both training and classification processes are offloaded from mobile to cloud. The second one is hybrid mobile-cloud processing, named "hybrid cloud," as investigated in our previous work [14]. The third one is no offloading at all, called "mobile-only," which means that all the ECG analysis procedures are executed on the mobile.

c) HMM Dynamic Scheduling: We applied the proposed HMM-based approach to dynamically seek the optimal options. In this mobile cloud ECG analysis scenario, choosing the heartbeat classification algorithms, offloading strategies and ECG sampling rates could be the action set used in HMM learning procedures, as shown in Algorithm 1.

Algorithm 1 Optimal action selection algorithm

```
for each action k do
    HMMLearning()        ▷ get a new Hidden Markov Model
    for each state i do
        for each optimal output j do
            probability(i,k)+=b(i,j)
for each state i do
    sorting probability(i,k) over k
    select action k' if probability(i,k') is maximal
```

IV. Experimental Results

To provide a comprehensive evaluation towards the efficacy and efficiency of our approach, we chose twelve representative patient records from the MIT-BIH database [17]. We used a Google Nexus 4 smartphone (equipped with Android 5.0.1 Lollipop OS and 2,100 mAh battery) as the mobile client in the experiment and built our test cloud infrastructure based on a Dell PowerEdge M620 server (equipped with twelve Xeon 2.5 GHz cores and 64 GB memory).

1) Optimal Energy Efficiency Driven Offloading Strategy: We implemented an HMM-based dynamic scheduling mechanism, in which the ANN-based classification (including both training and classification) would be performed either on the mobile locally, on the cloud remotely, or the hybrid cloud, according to the changing network conditions. We defined 6 network transmission rate ranges. For each patient record, two thirds of patient data were processed repetitively until 6000 heart beats were classified. We ran the entire patients data repetitively until the mobile phone battery ran out. For comparison, we also used the Linear Regression model to work as another dynamic scheduler to determine different offloading strategies under different network transmission rates. The energy consumption results (i.e., battery level drop) for those five configurations are shown in Figure 2. It is seen that our proposed HMM-based dynamic offloading strategy is more power saving (i.e., sustains much longer) than all of the static and dynamic Linear Regression offloading strategies.

2) Optimal Energy Efficiency and Accuracy Driven Offloading Strategy: In this setting, we seek to optimize the system towards the optimal energy saving while keeping a high level

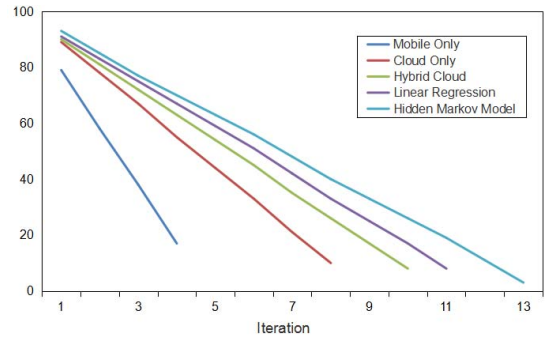

Fig. 2. Comparison of battery level drop among different approaches

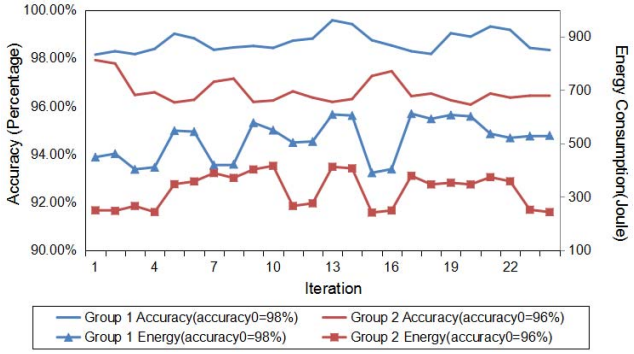

Fig. 3. Comparison of accuracy and energy consumption between two group of settings.

diagnosis accuracy. A multi-objective optimization problem is formulated as follows:

$$energy_{opt} = f(algorithm\ type, sampling\ rate)$$
$$(accuracy - accuracy_0) > \varepsilon \quad (4)$$

We implemented the HMM dynamic scheduler and ran two benchmarks together with our application simultaneously. The two benchmarks are qsort[1] and linpack[2]. In order to achieve the multi-objective optimization goal, the observation output for each hidden state is calculated as:

$$output = EnergyEfficiency * a$$
$$a = (accuracy - accuracy_0) > \varepsilon ?\ 1 : 0 \quad (5)$$

For each learning step, energy efficiency would be regarded as an effective output only if the diagnosis accuracy is maintained at a high level; otherwise, output would return to zero.

We used the "mobile only" option for both ANN and rule-based algorithms and set up 2 groups of experiment for comparison to see the effectiveness of this dynamic automatic control process. For both groups, 4000 heart beats of each patient are used for recording experiment result. For Group 1, we set the bar of accuracy as 98%, i.e., $accuracy_0 = 98\%$ in equation (4); For Group 2, we set the bar of accuracy as 96%, i.e., $accuracy_0 = 96\%$. The accuracy and energy consumption variation over time is illustrated in Figure 3. Each iteration step contains 2000 heart beats of a patient. Therefore, there are total of 24 iterations for 12 patients data. When processing

[1]http://code.google.com/p/android-benchmarks
[2]http://www.netlib.org/benchmark/linpackjava

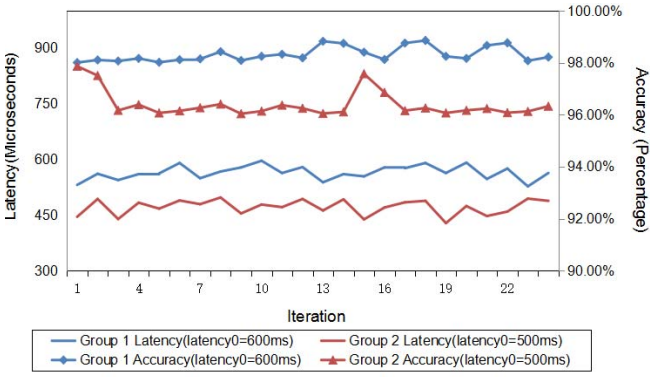

Fig. 4. Latency and Accuracy comparison between two group of settings

the total of 12 patients' data, we could find that the processing accuracy of Group 2 is lower than the processing accuracy of Group 1; however, Group 2 could achieve more energy saving than Group 1.

3) Optimal Accuracy and Performance Driven Offloading Strategy: In this setting, we seek to optimize the system towards optimal diagnosis accuracy while keeping a low processing latency level. A multi-objective optimization problem is formulated as follows:

$$accuracy_{opt} = f(algorithm\ type, sampling\ rate)$$
$$(latency - latency_0) < \varepsilon \quad (6)$$

Similar to the scenario IV-2 above, we implemented the HMM dynamic scheduler and the observation output for each hidden state is calculated as:

$$output = DiagnosisAccuracy * a$$
$$a = (latency - latency_0) < \varepsilon ?\ 1 : 0 \quad (7)$$

where, during each learning step, diagnosis accuracy would only be regarded as effective output if the processing latency could be maintained at a rather low level; otherwise, output would return zero.

As the scenario IV-2, we still used the "mobile only" option for both ANN and rule-based algorithms and set up 2 groups of experiment: we set the bar of latency as 600 millisecond ($latency_0 = 600ms$) for Group 1 and the bar of latency as 500 millisecond ($latency_0 = 500ms$) for Group 2. The accuracy and latency variations over time is illustrated in Figure 4. It can be observed that the processing accuracy of Group 2 is lower than the one of Group 1; however, Group 2 could achieve faster processing speed than Group 1.

V. Conclusion

Within this study, we present a new Hidden Markov Model (HMM) based dynamic scheduling approach to allow the system to adapt to the changing requirements while optimizing the operation towards multiple performance metrics, including processing latency, diagnosis accuracy, and energy consumption. We examined several simulated scenarios and evaluated multiple distinct performance requirements. The experimental results show that, our proposed approach can successfully identify the optimal system configuration for different requirements. Our investigation here provides insights on potential solutions for addressing challenges present in future personalized and context-aware healthcare.

References

[1] X. Wang, Q. Gui, B. Liu, Y. Chen, and Z. Jin, "Leveraging mobile cloud for telemedicine: A performance study in medical monitoring," in *NEBEC*, 2013, pp. 49–50.

[2] L. Badger, T. Grance, R. Patt-Corner, and J. Voas, "Cloud computing synopsis and recommendations (draft), NIST special publication 800-146," *Recommendations of the NIST, Tech. Rep*, 2011.

[3] Z. Jin and Y. Chen, "Telemedicine in the cloud era: Prospects and challenges," *IEEE Pervasive Computing*, vol. 14, no. 1, pp. 54–61, 2015.

[4] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *INFOCOM*, 2012, pp. 945–953.

[5] J. Kwak, Y. Kim, J. Lee, and S. Chong, "Dream: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Selected Areas in Communications*, vol. 33, no. 12, pp. 2510–2523, 2015.

[6] M. Shiraz, A. Gani, A. Shamim, S. Khan, and R. W. Ahmad, "Energy efficient computational offloading framework for mobile cloud computing," *Journal of Grid Computing*, vol. 13, no. 1, pp. 1–18, 2015.

[7] X. Lin, Y. Wang, Q. Xie, and M. Pedram, "Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment," *IEEE Trans. Services Computing*, vol. 8, no. 2, pp. 175–186, 2015.

[8] M. Barbera, S. Kosta, A. Mei, and J. Stefa, "To offload or not to offload? the bandwidth and energy costs of mobile cloud computing," in *INFOCOM*. IEEE, 2013, pp. 1285–1293.

[9] Z. Cheng, P. Li, J. Wang, and S. Guo, "Just-in-time code offloading for wearable computing," *IEEE Trans. Emerging Topics in Computing*, vol. 3, no. 1, pp. 74–83, 2015.

[10] C. Ragona, C. Fiandrino, D. Kliazovich, F. Granelli, and P. Bouvry, "Energy-efficient computation offloading for wearable devices and smartphones in mobile cloud computing," in *GLOBECOM*, 2015.

[11] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, 2015.

[12] Z. Jiang and S. Mao, "Energy delay tradeoff in cloud offloading for multi-core mobile devices," *Access, IEEE*, vol. 3, pp. 2306–2316, 2015.

[13] H. Wu, W. Knottenbelt, and K. Wolter, "Analysis of the energy-response time tradeoff for mobile cloud offloading using combined metrics," in *Teletraffic Congress (ITC 27), 2015 27th International*. IEEE, 2015, pp. 134–142.

[14] X. Wang, Q. Gui, B. Liu, Z. Jin, and Y. Chen, "Enabling smart personalized healthcare: a hybrid mobile-cloud approach for ecg telemonitoring," *IEEE J. Biomed. Health Inform.*, vol. 18, no. 3, pp. 739–745, 2014.

[15] J. J. Oresko, Z. Jin, J. Cheng, S. Huang, Y. Sun, H. Duschl, and A. C. Cheng, "A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing," *IEEE Trans. Inform. Technol. Biomed.*, vol. 14, no. 3, pp. 734–740, 2010.

[16] R. Ledesma and Z. Jin, "Resiliency analysis and modeling for real-time cardiovascular diagnostic devices," in *SPMB*. IEEE, 2012, pp. 1–6.

[17] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, 2001.