CARMUS: Towards a General Framework for Continuous Activity Recognition with Missing Values on Smartphones

Tianheng Wu, Liang Wang, Zhiwen Zheng, Simeng Wu, Jun Ma, Xianping Tao, Jian Lu
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
Email: {wth, zwzheng, wsm}@smail.nju.edu.cn, {wl, majun, txp, lj}@nju.edu.cn

***Abstract*—This paper presents the CARMUS framework for continuous activity recognition with missing values on smartphones. Besides the power and resource constraints discussed in existing work, our framework is proposed to further tackle the critical issue of missing values during data collection. We demonstrate the issue's impact on continuous recognition through a motivating example, and specify two challenges—blackouts and resource constraints—with respect to smartphone-based sensing and processing platforms. To address the challenges, CARMUS provides a novel framework which involves a lightweight admission control unit and a data imputation unit intuited by the daily repeated pattern and temporal smoothness of human activity data. Based on extensive experiments conducted on a real-world data set with 37% of the data missing, we show that the CARMUS framework is effective for achieving an 85.5% recognition accuracy by adopting the state-of-the-art imputation algorithms.**

***Index Terms*—Continuous Activity Recognition, Missing Values Imputation, Smartphones**

I. Introduction

Continuous activity recognition on smartphones [1], [2] is an important enabling technology for smartphone-based *continuous sensing applications* [3] such as life-logging [4], energy expenditure monitoring [5], health and well-being monitoring [6]. With the rapid advances of smartphones' hardware including sensing, storage and computing, as well as robust feature extraction and recognition algorithms [7] that tackle various real-world issues such as smartphone orientation and placement issues, it is now promising to perform reliable daily activity recognition using the smartphones—when effective sensor data are available. However, in real-life, continuous activity sensing and recognition on smartphones still faces many obstacles.

One of the most frequently discussed obstacles in the existing work is the *resource and power efficiency* issue on smartphones [2], [3], [8], [9]. For example, the Jigsaw continuous sensing engine proposed in [3] adopts admission control and duty cycling technologies to reduce the resource overheads of data collection. In [8], Razzaque et al. propose a compression-based approach to support energy-efficient data gathering. PowerForecaster is proposed in [9] to predict the power consumptions of continuous sensing applications at pre-install time. Yan et al. [2] propose an activity-adaptive approach for energy-efficient continuous activity recognition on smartphones. In summary, as discussed in [10], resource limitation is a major challenge for smartphone-based continuous sensing.

A natural consequence of the resource constraints is that the data will be partially missing if the collection services are not kept alive constantly. Making the situation even more critical, to perform continuous activity sensing and recognition, we need to further assume that the smartphone is always carried by its owner [11]. Through extensive empirical studies, Dey et al. find that the participants keep their phones within arm's reach for only 53% of the time on average [11]. To make sure the users always carry their phones, one solution is to ask the participants to keep the smartphones in their pockets during data collection [12]. However, it is not practical to make such requirements in real-life scenarios.

Summarizing the above discussions, we find that the *missing values issue*—either caused by resource constraints or ineffective data collected when the user is not carrying the phone[1]—is a critical issue for continuous activity sensing and recognition on smartphones. To address this issue, in this work, we propose the **CARMUS** framework for **C**ontinuous **A**ctivity **R**ecognition with **M**issing val**U**es on **S**martphones that detects and imputes the ineffective and missing values. To understand the impact of missing values on continuous activity recognition and the challenges faced by the CARMUS framework, we start our work with a motivating example as follows.

A. Motivating Example

In this section, we present a motivating example to understand the missing values issue and its impact on the performance of real-life smartphone-based continuous activity recognition systems.

1) Data Collection: One male subject is asked to carry two Android-based smartphones with data collection service installed—*Phone G* and *Phone D*, where *Phone G* is for ground-truth collection which is required to be always carried by the subject during the day; and *Phone D* is for real-life data collection following the subject's daily usage patterns[2]. During data collection, we focus on the missing values caused by the

[1] In this work, we are more focused on the latter cause while our framework is also capable to handle missing values with different causes.

[2] We use the notations *Phone G* and *Phone D* as explained here for the rest of the paper.

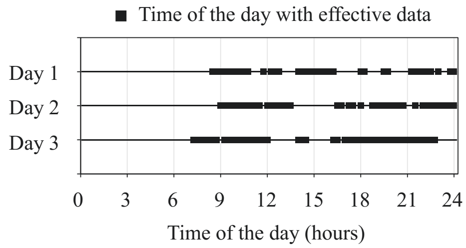

Fig. 1. Distribution of effective data in three days.

subject not carrying the phone and omit the power constraint by using additional power sources.

Fig. 1 illustrates the distribution of effective data collected by *Phone D* over three days. The effectiveness of the data is determined by the sensor data admission control unit introduced later in Sec. III-B. As shown in the figure, despite that the collection service is always kept alive, we still experience an average 55.9% of data loss over time because the subject is not carrying the phone, which is consistent with the results presented in [11]. Note the above experiments are conducted with the subject aware of the collection process, and the results are not intended to represent a general pattern of missing values for daily smartphone usage for different subjects. Instead, we focus on studying the impact of missing values on the performance of continuous activity recognition by using the subject's data as a case study.

2) Preliminary Results: To understand the issue's impact on continuous activity recognition performance, we evaluate the recognition accuracy using *Phone D*'s data by comparing the recognized activities against the results obtained by *Phone G* which is always kept on the subject's body. We consider three activities including *idle*, *walking*, and *biking*, recognized by a classification model trained using labeled data collected earlier by the subject. Details on the recognition process is introduced later in Sec. IV. Table I lists the recognition accuracies with three trivial solutions: 1) **Effective Only**: only use *Phone D*'s effective data (determined as the user is carrying the phone) for recognition, and mark the activity as *lost* for periods without effective data; 2) **Ignore**: ignore the data effectiveness issue and use all the data collected by *Phone D* for recognition; and 3) **Replace**: replace the ineffective data with the last effective data collected by *Phone D* before activity recognition. The **baseline** accuracy is obtained by only using *Phone D*'s effective data for recognition and compare the results with the corresponding ground-truth from *Phone G*.

It is clear from the results that the missing values issue, caused by the subject not carrying the phone, has a significant impact on the system's performance which cannot be addressed by trivial solutions such as **Ignore** or **Replace** introduced above. While the data collection process has potentially altered the usage pattern of *Phone D*, it is still feasible to demonstrate the issue's impact on continuous activity recognition on smartphones.

3) Challenges: By the experience gained in this preliminary experiment, we summarize the challenges faced by the

TABLE I
IMPACT OF THE MISSING VALUES.

Trivial Solutions	Recognition Accuracy over 24 hours $\times$ 3 days
Effective Only	40.5%
Ignore	46.8%
Replace	49.7%
Baseline	84.2%

CARMUS framework as follows.

1) **Blackouts**. In our scenario, the smartphone is the sole sensing and processing device to recognize its owner's activities[3]. As a result, all the sensor data will be missing together during periods that the smartphone fails to capture the user's activity.
2) **Resource Constraints**. It is prohibitive to process or transmit all the collected and history data on the smartphones due to their high volume. As a result, the CARMUS framework must select the most relevant data to be processed locally at the smartphones with stringent computing and storage resource constraints.

To address the missing values issue with the above challenges, we propose the CARMUS framework in this paper. In the next section, we briefly introduce our approach and the contributions made.

B. Proposed Approach & Contributions

Motivated by the above example, we propose the CARMUS framework that is designed following the idea to first identify the ineffective data to ensure the sensing quality (Sec. III-B), and then complete the sensor data stream under a novel imputation framework. The framework is designed to be general to incorporate the state-of-the-art missing value imputation algorithms for time series [13], [14], and efficient with respect to the unique challenges for smartphone-based sensing scenarios (Sec. III-C) as introduced next.

The proposed CARMUS framework tackles the **blackouts** issue by exploring the **daily repeated pattern** of activity data inspired by the daily routine of human activities [15] and the circadian rhythms [16]. More specifically, we obtain a data matrix by aligning different days' data with the same time of the day, and estimate the missing values by referring to data collected in previous days in this matrix. Through this technique, imputation algorithms designed for multi-dimensional time series [13], [14] can naturally fit into our framework to process data collected in multiple days. Experiment results suggest by increasing the number of days from 1 to 4, the activity recognition accuracy increases from 83.9% to 90.6% on the benchmarking data set, showing the effectiveness of our design.

Further, to tackle the **resource constraints** issue, we explore the **temporal smoothness** of human activity data [13] which means the similarity of sensing values over time [17]. Instead

[3] In this study, we omit the cases that users may share their activity data for privacy, communication overhead, and service availability concerns.

of using all the history data for imputation, we only include data in the near past into the matrix to estimate the current missing values. Through this technique, we limit the amount of input data to the imputation algorithm which is the bottleneck of the framework's data processing pipeline with respect to time and resource costs. Experiment results suggest that the system achieves a stable recognition accuracy of 85.5% on the complete real-world data set of 176 hours even by limiting the amount of history data to 15 minutes (30 minute segmentation with 50% overlap).

Finally, while the framework is evaluated mostly using the DynaMMo algorithm proposed by Li et al. in [13], we show by the experiments that CARMUS is a general framework that can also incorporate other state-of-the-art algorithms including Dynamic Contextual Matrix Factorization (DCMF) [14] and Non-negative Matrix Factorization (NMF) [18]. In summary, this paper makes the following contributions.

- We propose the CARMUS framework which is designed specifically to address the challenges of continuous activity recognition on smartphones, which is also general and efficient to adapt different state-of-the-art missing value imputation algorithms;
- We study the impact of missing values on the performance of smartphone-based continuous activity recognition, and specify the blackouts and resource constraints challenges faced;
- A prototype system is implemented to study the effectiveness and efficiency of the proposed approach;
- We conduct extensive experiments on a real-world data set to evaluate the performance and the resource overheads of the proposed system.

The rest of the paper is organized as follows. We introduce the related work in Sec. II. Detailed system design is presented in Sec. III. We introduce the implementation of our prototype in Sec. IV, and evaluate its performance through extensive experiments in Sec. V. Finally, Sec. VI concludes the paper.

II. Related Work

We provide a brief summary of the work closely related to this work in this section.

A. Smartphone Continuous Sensing Engines

Much work has been conducted to support continuous sensing on smartphones. Jigsaw [3] is a continuous sensing engine for mobile phone applications which require continuous monitoring of human activities and context. The authors develop a reusable sensing engine and propose application agnostic techniques that allow Jigsaw to be both resilient and energy efficient. StudentLife is proposed in [19] that integrates a flexible ecological momentary assessment (EMA) component so it could automatically infer human behavior in an energy-efficient manner. The energy cost of sensor data sampling is carefully managed in their framework. Besides, in [8], Razzaque et al. propose a compression-based approach to support energy-efficient data gathering, which balances the data quality and transmission overheads. Moreover, Yan et al. [2] propose an activity-adaptive approach for energy-efficient continuous activity recognition on smartphones. And they focus their discussions on the trade-off between classification accuracy and energy overheads.

While the energy constraints is an important issue for smartphone-based continuous sensing applications, in this paper, we focus on the issue of missing values caused by ineffective data collected. As shown by our motivating example, missing values are unavoidable even if we omit the energy constraints by adopting additional power sources.

B. Missing Data Imputation

Another series of work closely related to this work is missing data imputation technologies. In [18], Meng et al. solve the problem of estimating the missing values in crowd sensing applications through matrix factorization. They propose two novel regularizations in their objective function to model the similarities between the entities and the virtual users to improve the estimation accuracy.

For time series imputation, in [13], Li et al. propose DynaMMo which can estimate the missing values by learning their hidden patterns. By exploiting the temporal smoothness and spatial correlations, DynaMMo is shown to be promising in estimating the missing values in high-dimensional time series. In [14], Cai et al. propose the Dynamic Contextual Matrix Factorization (DCMF) based on joint matrix factorization and linear dynamic systems which models the contextual information using a contextual matrix. Extending the framework of DCMF, Facets [20] is proposed for fast comprehensive mining of co-evolving high-order time series. It formulates high-order time series as tensors and adopts the tensor decomposition model to find the latent factors of time series data.

While the above work studies the missing value problem for various time series, it is often assumed that data are collected by a network of distributed sensors [13], [14], [20]. As a result, the challenges of blackouts and resource constraints, though discussed in [13], is less sever compared to the smartphone-based sensing scenario where the phone is the sole sensing and processing device. To address this issue, we propose the CARMUS framework that is specifically designed to tackle these challenges, which is also a general framework to adopt the above state-of-the-art algorithms. Additionally, the above work focuses on the imputation task alone and mainly evaluates their approaches by estimation errors against the true values. Different from this work, our work focuses on the task of activity recognition which is essentially a classification task [21].

For classification tasks, Farhangfar et al. study the effect of missing data imputation on classification accuracy using six different imputation methods with six popular classifiers on 15 datasets [22]. Liu et al. [23] propose an adaptive imputation algorithm for incomplete pattern classification. Different from their work, our framework is designed to perform activity classification with time series. Moreover, our framework is designed with respect to the unique challenges of smartphone-based sensing as discussed above.

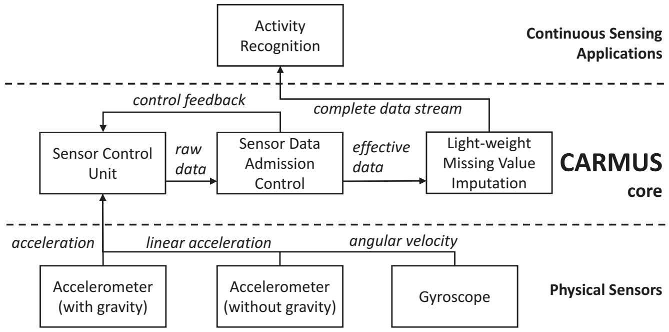

Fig. 2. Overview of the CARMUS framework.

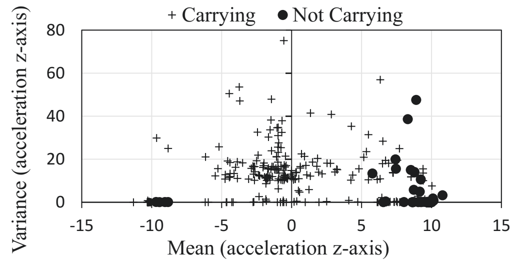

Fig. 3. Distribution of instances with user carrying / not carrying the phone in the feature space.

For activity recognition with missing values, Savvaki et al. [24] propose to organize the available data streams into low-rank 2-D and 3-D Hankel structures, and address the missing values issue using matrix and tensor completion technologies. Different from their approach, we propose to obtain a data matrix by including data in the near past and previous days intuited by the nature of human activities.

C. Sensor Data Admission Control

Finally, our work is also related to admission control [25] and anomaly detection [26] in time series. In [25], the authors perform admission control to reduce the systems overheads on CPU, memory and power consumptions. Different from this work, we consider sensor data admission control with respect to the data effectiveness issue. In [11], the authors investigate the proximity of users to their smartphones to be within the arm's reach or in the same room. And different from their work, we require the smartphone to be carried by the user to make the collected data effective. Our work is also different from [26] that discusses dirty data detection and repairing for we focus on the scenario of smartphone-based activity recognition, and design our framework based on the nature of human activity data.

III. CARMUS Design

In this section, we introduce the design of the proposed CARMUS framework for continuous activity sensing and recognition on smartphones.

A. Overview

Fig. 2 illustrates an overview of the CARMUS framework. To perform activity recognition on the smartphones, CARMUS collects raw data from three inertial sensors: *acceleration* (with gravity), *linear acceleration* (without gravity), and *angular velocity* from the gyroscope. The activation and sampling rate of the inertial sensors are controlled by the **Sensor Control Unit** according to the control feedback from the sensor data admission control unit.

The **Sensor Data Admission Control Unit**, inside the core of CARMUS, takes the raw *acceleration* data as input and determines whether the sampled data are effective in representing the user's current activity. If the result is positive (data are effective), it asks the sensor control unit to activate all the three sensors for data collection, and simply forwards the data to the later units. On the contrary, if the result is negative, it reduces the resource overheads of data collection by commanding the sensor control unit to only collect the *acceleration* data.

The sensor data filtered by the admission control unit only contains effective data, leading to a large portion of missing values in the data stream over time as shown earlier by the motivating example in Sec. I-A, which is infeasible for continuous activity recognition. To address this issue, another core component, the **Missing Value Imputation Unit** is designed to estimate the missing values and generate a complete sensor data stream with respect to the three types of sensors to support continuous activity recognition. As introduced later in Sec. III-C, there is a trade-off between the imputation algorithm's performance and resource overheads given the restrictions of the smartphone platform.

In the following sections, we introduce the design details of the core system components and benchmarking results to establish the key parameters.

B. Sensor Data Admission Control

We first introduce the design of the sensor data admission control unit which is responsible for identifying ineffective data and preventing them from polluting the data stream fed to the activity recognition component.

1) Problem Description and Proposed Solution: For activity recognition tasks discussed in this paper, the inertial sensor data are effective only when the user is carrying the phone. As a result, we model the problem of determining the effectiveness of sensor data as a binary classification problem with the two classes corresponding to the user *carrying* and *not carrying* the phone, respectively. More specifically, we model the problem as follows.

Given the sensor data collected in a time window of l samples starting at time t, $\boldsymbol{D}_{t,l} =< \boldsymbol{d}_t, \boldsymbol{d}_{t+1}, ..., \boldsymbol{d}_{t+l-1} >$, assign a label to $\boldsymbol{D}_{t,l}$ to be carried *or* not carried *by the user, where $\boldsymbol{d}_t$ is the sensor data collected at time t.*

To solve the above problem, we propose the solution based on two observations: 1) the phone is often left unmoved when the user is not carrying it; 2) in most cases, the phone is left on a surface such as a table with its screen facing up or down. The first observation indicates that the *variance* of acceleration readings is an effective indicator. And the second observation leads us to focus on the z-axis to compare its readings with $1G \approx 9.8m/s^2$. As a result, we propose to use the *mean* and *variance* of the z-axis readings in $D_{t,l}$ to represent the features of readings contained in the window.

Fig. 3 illustrates the distribution of instances with the user *carrying* / *not carrying* the phone in the 2D feature space of the the z-axis's *mean* and *variance* with a window size of $l = 1s$ and sampling rate of $50Hz$. As shown by the figure, instances belong to different classes discriminates well with each other in the feature space as expected. We then conduct an experiment by applying a J48 decision tree classifier. The resulting detection accuracy with ten-fold-cross-validation is 97.9%, suggesting the effectiveness of the proposed solution.

Since the ineffective data are not useful for the later tasks, it is desirable to keep the data collection and processing costs to a minimum when the user is not carrying the phone. More specifically, the admission control unit can duty-cycle the data effectiveness detection task by activating the sensors periodically and hibernate on detecting the data are ineffective. While the detailed scheduling strategy is application specific, one important question is *what is the minimum amount of data necessary to make an accurate detection for the effectiveness of the data?* The above proposed solution has already limited the sensor to be the z-axis of acceleration based on our observations, it is beneficial to further determine the minimal sampling rate r and window size l for lower costs, which we obtain in the next section through benchmarking.

2) Parameter Benchmarking: To find the minimum amount of data necessary to make an accurate data effectiveness detection, we study the admission control unit's detection accuracy under different combinations of sampling rate r and window size l for the acceleration's z-axis.

The experiment is conducted by first asking two subjects (one male and one female) to collect labeled (*carrying/not carrying* the phone) acceleration data sampled at 50Hz for 67 days. We then synthesize the data for different sampling rates r by resampling the original data set, and segment the resampled data with different window sizes l. A J48 decision tree-based classification model is built to classify the data segments to be the subject *carrying/not carrying* the phone. Detection accuracy is evaluated by ten-fold-cross-validation.

Fig. 4 illustrates the detection accuracies with different sampling rates r ($5Hz$ to $50Hz$ with a step of $5Hz$) and window sizes l ($1s$ to $10s$ with a step of $1s$). A general observation made from the figure is that the unit's performance is insensitive to the parameters. The detection accuracy is 97.7% for $r = 5Hz, l = 1s$, and 98.1% for $r = 50Hz, l = 10s$, suggesting that an accurate detection can be made with a few data samples at a low cost. As a result, we use $r = 5Hz, l = 1s$ for data effectiveness detection if not specified otherwise.

A natural result of eliminating the ineffective data is the missing values contained in the sensor data stream. The missing value imputation unit introduced next is then designed to address this issue.

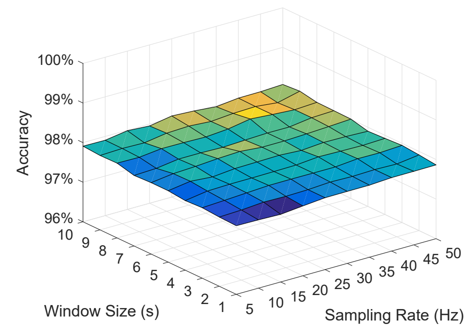

Fig. 4. Data effectiveness detection accuracy under combinations of window size and sampling rate.

C. Missing Value Imputation

The raw data stream passed the above sensor data admission control unit only contains effective data that represent the user's activity. However, as shown by the motivating example in Sec. I-A, it is infeasible to obtain reliable continuous recognition results by only using the effective data. As a result, it is important to complete the data stream to obtain reasonable estimations of the missing values.

1) Problem Description: The problem of missing data imputation can be presented informally as *estimating the missing values contained in a continuous sensor data stream*. While the problem appears to be similar to those discussed in existing work [13], [14], [18], there are some unique challenges with our application scenario including **blackouts** and **resource constraints** as discussed in Sec. I-A3. Following the above discussions, we formulate the problem as follows.

Given the temporal data sequence collected up to time T, $\boldsymbol{D}_T =< \boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_t, ..., \boldsymbol{d}_T >$ and an indicating vector $\boldsymbol{W}$ of size T, $\boldsymbol{W} =< w_1, w_2, ..., w_t, ..., w_T >$, estimate the values of all sensors for $\boldsymbol{d}_t$ with $w_t = 0$, $\forall t \in [1, T]$, where $\boldsymbol{d}_t$ is the sensor data collected at time t, and scalar $w_t = 0$ means all data for $\boldsymbol{d}_t$ are missing, $w_t = 1$ for otherwise.

A key difference of our problem against those discussed in existing work lies in that the indicator at time t, w_t, is a scalar instead of a vector [14]. This is because of the blackouts issue discussed in Sec. I-A3 that all the sensor data will be missing together if the smartphone fails to sense the user's activity at time t.

2) Proposed Solution: To address the above problem, we design the imputation framework in CARMUS following the nature of daily human activities:

1) **Daily Repeated Patterns**. Based on existing evidences of daily routine of human activities [15] and human circadian rhythms [16], we propose to tackle the **blackouts** by correlating the current missing values with non-missing values collected (or estimated) at the same time of the day in previous days.
2) **Temporal Smoothness**. Existing work on sensor-based activity recognition suggests the similarity of sensing values over time [17] for human activities. Following

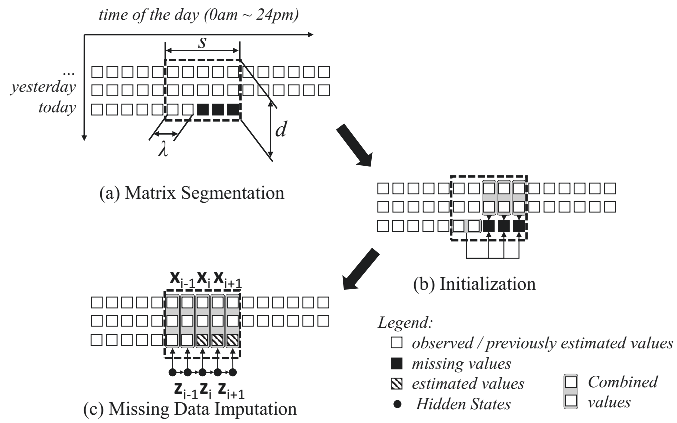

Fig. 5. Design of the missing data imputation unit, illustrated with the DynaMMo [13] algorithm.

this observation, data collected in the near past are more correlated/similar to the current values. As a result, we propose to address the **resource constraints** issue by limiting the volume of data processed by the imputation algorithm to only include the data in the closest past.

Following the above analysis, we design the missing value imputation unit in our CARMUS framework as illustrated in Fig. 5. More specifically, the missing data imputation process is composed of three steps.

Matrix Segmentation. Given the current data involving missing values, we first obtain a data matrix by segmenting the current and history data buffered in the system as shown in the dashed box in Fig. 5(a). Data collected in different days are aligned by the *time of the day* according to the daily repeated activity pattern discussed above. To explore the temporal smoothness among the data, we also include part of the last observed data into the segment.

As shown by the dashed box in Fig. 5(a), the size and position of the data matrix are determined by three parameters—the number of days d, the segmentation size s, and the overlapping rate λ which is the portion of effective data (or estimated previously) in the current segment. Note the CARMUS framework is designed to perform online imputation, which means only the observed data will be included in the current segment. For applications that do not have an online requirement, it is trivial to extend the current framework to include both the observed and future data into the current segment.

Initialization. Before applying the missing value imputation algorithm introduced next, we first initialize the missing values by averaging the last observed data and data with the same time of the day in previous days as shown in Fig. 5(b). In [13], the authors use linear interpolation to initialize the missing values. In this paper, we show by our experiment results that the proposed initialization approach improves the system's performance compared to linear interpolation in Sec. V-D2.

Missing Value Imputation. As shown in Fig. 5(c), after the above preprocessing steps, the last step is to complete the data segment by missing value imputation. In this work, we adapt the DynaMMo [13] algorithm for this task because it explores both the temporal smoothness and spatial correlations among different sensors, which well fits into our scenario. It is important to note that the main focus of this paper is to propose a general framework for missing value-tolerant activity recognition which can integrate different imputation algorithms such as DCMF, Non-negative Matrix Factorization, etc. And we leave the topic of choosing and designing more effective algorithms for our future work.

To be more specific, we adapt DynaMMo [13] into our framework with the following steps. Data collected with the same time of the day i across d days form the observation at time i, $\mathbf{x}_i$ (data at time i for each day contain readings from the three sensors). Each $\mathbf{x}_i$ corresponds to a hidden state $\mathbf{z}_i$ through a projection matrix $\mathbf{G}$ with an additional Gaussian noise term $\epsilon_i \sim \mathcal{N}(0, \Sigma)$, i.e.,

$$\mathbf{x}_i = \mathbf{G}\mathbf{z}_i + \epsilon_i \tag{1}$$

where the projection matrix $\mathbf{G}$ models the spatial correlation among different axes across different days.

The hidden state at the next time step is temporally correlated to the hidden state at the current step through a transition matrix $\mathbf{F}$ and an additional Gaussian noise term $\omega_i \sim \mathcal{N}(0, \Lambda)$, i.e.,

$$\mathbf{z}_{i+1} = \mathbf{F}\mathbf{z}_i + \omega_i \tag{2}$$

where the transition matrix $\mathbf{F}$ models the temporal smoothness among the data [13].

The model parameters are estimated by maximizing the log-likelihood of the joint distribution of hidden states and the observations through Expectation-Maximization. And the missing values are estimated using the resulting series of hidden states. We omit further details in this paper because we focus on the design of the framework instead of the imputation algorithms. Readers interested in imputation algorithms can refer to [13], [14], [18] for details.

IV. Continuous Activity Recognition Prototype

In this section, we introduce the implementation of the prototype for continuous activity recognition for the CARMUS framework on Andriod smartphones.

The prototype is implemented following the typical activity recognition chain introduced in [21]. The complete raw sensor data stream preprocessed by the CARMUS framework contains 3-axis data of the *acceleration* with gravity and *linear acceleration* without gravity from the accelerometer, and the *angular velocity* from the gyroscope. The sampling rate is set to $5Hz$ to balance the processing overheads on smartphones while maintaining a high recognition accuracy.

The data stream is first segmented using a sliding window of 30s with 50% overlap. For each window, we extract nine features for each axis of the above three sensors. The features include *mean*, *variance*, *median*, *max*, *min*, *1st quarter*, *3rd quarter*, *energy*, and *zero crossing rate*. For activity recognition, we model the problem as a multi-class classification

TABLE II
OVERVIEW OF THE DATA COLLECTED.

	Subject 1	Subject 2
Hours of Ground-truth Data	101.1	74.9
Hours of Effective Data	59.9	51.0
Percentage of Effective Data over Time*	58.9%	68.1%

*To better interpret the results, the percentage is calculated with respect to the hours of ground-truth data collected rather than 24 hours a day.

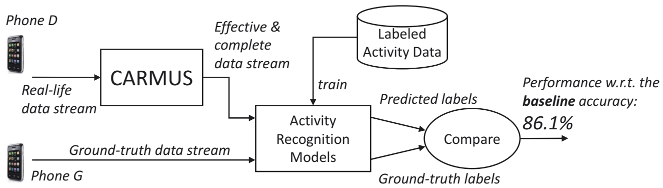

Fig. 6. Experiment methodology.

problem and use the Random Forest classifier provided by Weka [27] to implement the prototype. We set the number of trees to ten to achieve a balance between the recognition performance and resource overheads.

The above parameters are determined by a preliminary study on a benchmarking data set collected in our previous work. Details on parameter tuning are omitted in this paper for page limits.

V. EMPIRICAL STUDIES

We evaluate the performance of the proposed continuous sensing engine in this section through empirical studies.

A. Data Collection and Experiment Methodology

We introduce the data collection process, and the experiment methodology in this section.

1) Data Collection: With IRB approval, data collection is performed by two young male subjects for one week. Each subject is asked to carry two smartphones—*Phone G* and *Phone D* as described in Sec. I-A1. The power constraints are omitted in this study by connecting the smartphones to external batteries when necessary. Table II lists an overview of the data collected, which suggests the overall missing rate of *Phone D* is 37% compared to the amount of ground-truth data collected by *Phone G*. With respect to the above data collection process, the possible threats to validity include:

- The subjects are aware of the data collection process which may potentially alter their normal daily usage patterns of the smartphones;
- We only include two young male subjects and do not include data from females and elderlies.

Fortunately, the main focus of this paper is to study the effectiveness of the proposed CARMUS framework on addressing the missing values issue, rather than studying the missing patterns of different subjects. Because the above data are collected in a real-world manner, and contain missing values close to their natural patterns, we argue that the data collected are feasible for the following experiments.

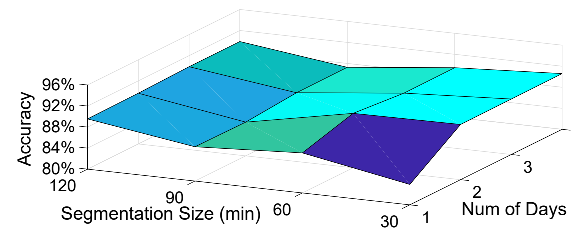

Fig. 7. Performance impact of segmentation size and number of days.

2) Experiment Methodology: Fig. 6 illustrates our experiment methodology. Data collected by *Phone D* are filtered and imputed by the proposed CARMUS framework to generate an effective and complete data stream. Both the data streams generated by CARMUS and collected by *Phone G* are fed to the activity recognition models trained using 45 hours of labeled data from the same subjects. Three activities are selected to demonstrate the effectiveness of the system, which include: *idle*, *walking*, and *biking*. Finally, the system's performance is evaluated by comparing the predicted and ground-truth activity label sequences. Hours without ground-truth data are omitted during the evaluation to make the results comprehensible. Additionally, we study the performance of the system under different configurations with respect to the **baseline** accuracy of 86.1% which is obtained by only using the effective data and their corresponding ground-truth for evaluation (as did in Sec. I-A). We evaluate the system's performance by comparing to the baseline accuracy instead of 100% accuracy to eliminate the impact of differences in data collected by the two phones.

In the following sections, we present the experiment results with respect to the system's performance and resource overheads. While most of the results are obtained by replaying the data on a PC, the resource overheads are evaluated using a benchmarking smartphone.

B. Impact of System Parameters

In this section, we evaluate the system's performance using the DynaMMo [13] algorithm with respect to system parameters including the segmentation size s, number of days d, and overlapping rate λ following our design in Fig. 5. Due to the large size of the parameter space, we start our evaluation by fixing the overlapping rate $\lambda = 0.5$ and test the system's performance with different combinations of segmentation size s and number of days d. Additionally, the time consumption is prohibitive if we test all the combinations of parameters on the complete data set of 176 hours. As a result, we select the data from 8pm to 11pm collected by subject 2 to benchmark the system's performance because the portion of missing values and distribution of activity classes during this period is close to the complete data set.

1) Segmentation Size s & Number of Days d: In this experiment, we evaluate the impact of segmentation size s and number of days d on system's performance with overlapping rate $\lambda = 0.5$ fixed. We empirically choose four candidate val-

TABLE III
IMPACT OF OVERLAPPING RATE λ.

λ	0.3	0.4	0.5	0.6
Accuracy	89.2%	89.7%	90.5%	90.3%

ues for s and d, respectively, including $s = \{30, 60, 90, 120\}$ minutes and $d = \{1, 2, 3, 4\}$ days.

Fig. 7 plots the system's recognition accuracy with different segmentation sizes and number of days. From the figure, it can be observed that when $s = 30$ minutes, the recognition accuracy increases sharply from 83.9% to 90.5% by increasing d from 1 to 2 days. Similar trends can also be observed with different segmentation sizes. This result suggests that the system's performance is improved by including data collected at the same time of the day in previous days, which validates our design with daily repeated patterns. On the other hand, merely increasing the segmentation size does not have a clear effect on the recognition accuracy. While the accuracies are generally close to each other when $s = 30$, 60, and 120 minutes, there is an average drop of 2% in accuracy when $s = 90$ minutes. In summary, with respect to segmentation sizes and number of days, there are three combinations that show high recognition accuracies, i.e., with s all set to 30 minutes and $d = 2$, 3, and 4 days. The corresponding recognition accuracies are 90.5%, 90.5%, and 90.6%, respectively.

While the highest recognition accuracy is achieved with $s = 30$ minutes and $d = 4$ days, the resource consumption is prohibitive as reported later in Sec. V-E. As a result, we choose the optimal combination to be $s = 30$ minutes and $d = 2$ days for the rest of the experiments.

2) Overlapping Rate λ: Based on the above results, further evaluate the impact of the overlapping rate λ on the system's performance. Empirically, we test the system's performance under four settings including $\lambda = \{0.3, 0.4, 0.5, 0.6\}$.

Table III summarizes the system's performance under different λ with $s = 30$ minutes and $d = 2$ days. Though the highest accuracy is achieved by $\lambda = 0.5$, the above result suggests different overlapping rates achieve comparable performances. As a result, we follow the settings used in the above experiment and determine the optimal λ to be 0.5.

C. System Performance with Optimal Parameters

Based on the above experiment results, we choose the optimal parameters to be $s = 30$ minutes, $d = 2$ days, and $\lambda = 0.5$. We then evaluate the system's performance on the complete data set of 176 hours from the two subjects.

The overall recognition accuracy is 85.5%, comparable to the baseline accuracy of 86.1% introduced in Sec. V-A2. Fig. 8 plots the confusion matrix together with the precision and recall for different classes. We make the following observations from the confusion matrix.

First, the data set is biased to the *idle* class which represents cases the subjects are sitting or standing still. This is natural because the subjects spend most of their time in the laboratory. As a result, the precision and recall for the *idle* class are both above 0.9, which are comparable to the results obtained in the baseline study.

Ground-truth \ Predicted	*idle*	*walk*	*bike*	recall	cmp. to baseline
idle	15370	1305	107	0.92	⇧ 0.01
walk	1478	2644	69	0.63	⬇ 0.09
bike	54	41	50	0.34	⬇ 0.06
precision	0.91	0.66	0.22		
cmp. to baseline	⬇0.01	⬇0.06	⬇0.08		

Fig. 8. Confusion matrix.

Second, the system performs poorly on the *biking* class with precision and recall being 0.22 and 0.34, respectively, by mis-classifying 28.3% and 37.2% of the instances to *walking* and *idle*, respectively. Possible explanations include: 1) there are fewer instances for *biking* than the other two classes, leading to a biased recognition result; and 2) many instances are classified as *idle* because the estimated missing values do not reflect the true state of the subjects, suggesting further improvements of the imputation algorithms are needed. However, on the other hand, the result also suggests the drop in precision and recall are less significant (0.08 and 0.06, respectively) compared to the baseline results. This is because the recognition accuracy for *biking* is low even by omitting the missing values in the baseline study, which is possibly caused by the label bias issue in the original data set.

Similarly for the *walking* class, after applying the imputation algorithm in the CARMUS framework, the precision and recall decrease for 0.06 and 0.09, respectively. Further study reveals that the loss is mainly caused by the estimation errors for missing values, which makes 35.3% of the *walking* data to be mis-classified as *idle*.

Because the absolute recognition accuracy is not our main focus, in summary, given that 37% of the data are missing in the collected data set, we conclude that the proposed CARMUS framework powered by the DynaMMo algorithm is effective for keeping the system's performance from dropping significantly compared to the baseline.

D. System Components

In this experiment, we evaluate the performance impact of different components, including the admission control unit, and the algorithms adopted in the missing data imputation unit to validate our design.

1) Admission Control Unit: To study the effectiveness of the admission control unit, we compare the system's performance with the unit disabled against the above results obtained with the unit enabled. The resulting recognition accuracy drops significantly to 70.9% by disabling the admission control unit. Comparing to the recognition accuracy of 85.5% reported above, we conclude that the ineffective data have a significant impact on the system's performance and the admission control unit proposed in the CARMUS framework is effective and important in preventing the ineffective data from polluting the data stream fed to the activity recognition models.

TABLE IV
DIFFERENT IMPUTATION ALGORITHMS.

Algorithm	DynaMMo [13]	NMF [18]	DCMF [14]
Accuracy	85.5%	85.3%	75.8%

Time (sec.) [Memory (MB)]		Number of Days			
		1	2	3	4
Segmentation Size [min]	30	66 [125]	254 [185] *	N/A [>275]	N/A [>325]
	60	140 [160]	521 [270]	N/A [>307]	N/A [>307]
	90	208 [200]	N/A [>297]	N/A [>307]	N/A [>338]
	120	293 [225]	N/A [>317]	N/A [>307]	N/A [>338]

Fig. 9. Time (above, in seconds) and memory (below, in MB) consumptions of processing single segments with different sizes, time consumption of N/A represents cases that the process is killed for out-of-memory error.

2) Initialization Approaches: Different from the linear interpolation-based initialization approach proposed in DynaMMo [13], in this paper, we propose to initialize the missing values using history data from both the current and previous days determined by the segment. By comparing the results, we discover that the detection accuracy increases by 1.2% on average when using the proposed initialization approach than initializing with linear interpolation. Moreover, the DynaMMo algorithm also converges 9.3% faster on average when the missing values are initialized by the proposed approach. This result suggests that the system's performance is generally improved with better initial values.

3) Adopting other Imputation Algorithms: In this experiment, we demonstrate that CARMUS is a general framework to incorporate different state-of-the-art imputation algorithms by adopting the NMF [18] and DCMF [14] algorithms. Table IV lists the system's recognition accuracy with different imputation algorithms. While the DynaMMo and NMF algorithms achieve comparable results, the DCMF algorithm is shown to be less effective on our data set. However, this result does not suggest DCMF is less effective than the other algorithms because it is designed to estimate the missing values in the data collected by a network of sensors instead of a single smartphone [14]. And its performance can be further improved by tuning the parameters with respect to our application scenario. We leave the topic of selecting and designing more optimized imputation algorithms for our future work.

E. Resource Overheads

In this experiment, we evaluate the resource overheads of the proposed CARMUS framework with respect to time, memory, power, and CPU consumptions with a benchmark smartphone equipped with an 8-core 2.45GHz CPU and 6GB RAM powered by the Android OS. We focus on the missing data imputation unit because the costs for the other units are relatively insignificant. Fig. 9 lists the time and memory consumptions of processing a single segment on the smartphone by varying its size. It is observed that the smartphone is unable to complete the processing for more than half of the sizes (marked with N/A). For the optimal parameter selected in the above experiments (marked with $*$), it takes 254 seconds to process data of 30 minutes with a memory overhead of 185MB. Additionally, the CPU overhead under this settings is approximately 15%. And the battery power consumption is 174.2mA.

The above results suggest that by properly selecting the parameters, the proposed CARMUS framework is able to process the data in real-time[4] with acceptable memory, CPU, and battery power overheads on modern smartphones. On the other hand, the results also raise the need for more light-weight imputation algorithms suitable for smartphone platforms, which will be our research interest in the future.

VI. CONCLUSION

In this paper, we propose the CARMUS framework designed to tackle the missing values issue caused by ineffective data collected to support continuous activity sensing and recognition applications on smartphones. Besides the power and resource constraints discussed in existing work, we further study the problem of missing values which is shown to be critical for our applications through a motivating example. Two challenges, namely the blackouts and resource constraints issues, are specified by the nature of smartphone-based sensing and processing platforms. With respect to the challenges, the CARMUS framework is designed following observations on the daily repeated pattern and temporal smoothness of human activity data. More specifically, CARMUS involves in its core a light-weight data admission control unit that eliminates the ineffective data, and adopts the state-of-the-art imputation algorithms to estimate the missing values under a novel framework.

Extensive experiments are conducted using a real-world data set with 37% of the data missing. The results suggest the framework, by adopting the DynaMMo [13] imputation algorithm, is effective by maintaining the system's recognition accuracy to be 85.5%, comparable to the baseline which only uses the effective data for recognition. By carefully selecting the parameters, the framework is shown to be able to perform real-time processing with resource overheads acceptable to modern smartphones. Moreover, we demonstrate that CARMUS is a general framework to incorporate different imputation algorithms.

For our future work, we plan to: 1) design more effective imputation algorithms that tackle the unique challenges raised by the smartphone platform and explore the nature of human activity data; 2) design light-weight and adaptive algorithms that can identify and estimate the missing values in various sensor data streams; 3) conducting experiments to involve more subjects and activities.

[4] Because the processing time of 254 seconds is significantly less than the data collection time of 30 minutes.

Acknowledgment

This work is supported by the National Key R&D Program of China under Grant No. 2017YFB1001801, NSFC No. 61502225, and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] F. Kawsar, M. K. Hasan, R. Love, and S. I. Ahamed, "A novel activity detection system using plantar pressure sensors and smartphone," in *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, vol. 1. IEEE, 2015, pp. 44–49.

[2] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer, "Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach," in *16th International Symposium on Wearable Computers (ISWC'12)*. Ieee, 2012, pp. 17–24.

[3] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The jigsaw continuous sensing engine for mobile phone applications," in *Proceedings of the 8th ACM conference on embedded networked sensor systems*. ACM, 2010, pp. 71–84.

[4] R. Rawassizadeh, M. Tomitsch, K. Wac, and A. M. Tjoa, "Ubiqlog: a generic mobile phone-based life-log framework," *Personal and ubiquitous computing*, vol. 17, no. 4, pp. 621–637, 2013.

[5] A. Pande, Y. Zeng, A. K. Das, P. Mohapatra, S. Miyamoto, E. Seto, E. K. Henricson, and J. J. Han, "Energy expenditure estimation with smartphone body sensors," in *Proceedings of the 8th International Conference on Body Area Networks*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013, pp. 8–14.

[6] N. D. Lane, M. Mohammod, M. Lin, X. Yang, H. Lu, S. Ali, A. Doryab, E. Berke, T. Choudhury, and A. Campbell, "Bewell: A smartphone application to monitor, model and promote wellbeing," in *5th international ICST conference on pervasive computing technologies for healthcare*, 2011, pp. 23–26.

[7] H. Guo, L. Chen, G. Chen, and M. Lv, "Smartphone-based activity recognition independent of device orientation and placement," *International Journal of Communication Systems*, vol. 29, no. 16, pp. 2403–2415, 2016.

[8] M. A. Razzaque and S. Clarke, "Compression-based energy efficient sensor data gathering framework for smartphones," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International*. IEEE, 2016, pp. 126–132.

[9] C. Min, C. Yoo, J. Song, Y. Lee, S. Kang, and I. Hwang, "Powerforecaster: Predicting power impact of mobile sensing applications at pre-installation time," *GetMobile: Mobile Computing and Communications*, vol. 20, no. 1, pp. 30–33, 2016.

[10] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications magazine*, vol. 48, no. 9, 2010.

[11] A. K. Dey, K. Wac, D. Ferreira, K. Tassini, J.-H. Hong, and J. Ramos, "Getting closer: an empirical investigation of the proximity of user to their smart phones," in *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 2011, pp. 163–172.

[12] S. Inoue, N. Ueda, Y. Nohara, and N. Nakashima, "Recognizing and understanding nursing activities for a whole day with a big dataset," *Journal of Information Processing*, vol. 24, no. 6, pp. 853–866, 2016.

[13] L. Li, J. McCann, N. S. Pollard, and C. Faloutsos, "Dynammo: Mining and summarization of coevolving sequences with missing values," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 507–516.

[14] Y. Cai, H. Tong, W. Fan, and P. Ji, "Fast mining of a network of coevolving time series," in *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 2015, pp. 298–306.

[15] J. Yin, Q. Zhang, and M. Karunanithi, "Unsupervised daily routine and activity discovery in smart homes," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. IEEE, 2015, pp. 5497–5500.

[16] J. N. Mills, "Human circadian rhythms." *Physiological reviews*, vol. 46, no. 1, pp. 128–171, 1966.

[17] C.-H. Wu and Y.-C. Tseng, "Data compression by temporal and spatial correlations in a body-area sensor network: A case study in pilates motion recognition," *IEEE Transactions on Mobile Computing*, vol. 10, no. 10, pp. 1459–1472, 2011.

[18] C. Meng, H. Xiao, Y. Cheng, and Y. Cheng, "Tackling the redundancy and sparsity in crowd sensing applications," in *ACM Conference on Embedded Network Sensor Systems Cd-Rom*, 2016, pp. 150–163.

[19] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. T. Campbell, "Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014, pp. 3–14.

[20] Y. Cai, W. Fan, W. Fan, P. Ji, and Q. He, "Facets: Fast comprehensive mining of coevolving high-order time series," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 79–88.

[21] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, p. 33, 2014.

[22] A. Farhangfar, L. Kurgan, and J. Dy, "Impact of imputation of missing values on classification error for discrete data," *Pattern Recognition*, vol. 41, no. 12, pp. 3692–3705, Dec. 2008.

[23] Z.-g. Liu, Q. Pan, J. Dezert, and A. Martin, "Adaptive imputation of missing values for incomplete pattern classification," *Pattern Recognition*, vol. 52, pp. 85–95, 2016.

[24] S. Savvaki, G. Tsagkatakis, A. Panousopoulou, and P. Tsakalides, "Matrix and tensor completion on a human activity recognition framework," *IEEE journal of biomedical and health informatics*, vol. 21, no. 6, pp. 1554–1561, 2017.

[25] L. Wang, W. Cheng, L. Pan, T. Gu, T. Wu, X. Tao, and J. Lu, "Spiderwalk: Circumstance-aware transportation activity detection using a novel contact vibration sensor," in *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2018.

[26] A. Zhang, S. Song, J. Wang, and P. S. Yu, "Time series data cleaning: from anomaly detection to anomaly repairing," *Proceedings of the VLDB Endowment*, vol. 10, no. 10, pp. 1046–1057, 2017.

[27] T. C. Smith and E. Frank, *Statistical Genomics: Methods and Protocols*. New York, NY: Springer, 2016, ch. Introducing Machine Learning Concepts with WEKA, pp. 353–378. [Online]. Available: http://dx.doi.org/10.1007/978-1-4939-3578-9_17