OUKA
The University of Osaka
Institutional Knowledge Archive

Title	Research on Sensor Data Processing Methods for Recognizing and Understanding Factory Work Activities
Author(s)	Xia, Qingxin
Citation	大阪大学, 2021, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/85436
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

Research on Sensor Data Processing Methods for Recognizing and Understanding Factory Work Activities

Submitted to
Graduate School of Information Science and Technology
Osaka University

July 2021

Qingxin XIA

List of Publications

1. Journal Paper

1. Q. Xia, A. Wada, J. Korpela, T. Maekawa, and Y. Namioka. Unsupervised factory activity recognition with wearable sensors using process instruction information. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(2), June 2019.

2. Q. Xia, J. Korpela, Y. Namioka, and T. Maekawa. Robust unsupervised factory activity recognition with body-worn accelerometer using temporal structure of multiple sensor data motifs. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3), Sept. 2020.

2. Domestic Conference Paper (with peer-review)

1. Q. Xia, A. Wada, J. Korpela, T. Maekawa, and Y. Namioka: Preliminary investigation of assembly work activity recognition with wearable sensors via unsupervised learning, In *Proceedings of the Multimedia, Distributed, Cooperative, and Mobile Symposium*, *Dicomo'19*, NO.6E-1, June 2019.

2. Q. Xia, J. Korpela, Y. Namioka, and T. Maekawa: Preliminary investigation of unsupervised factory activity recognition with wearable sensors via temporal structure of multiple motifs, In *Proceedings of the Multimedia, Distributed, Cooperative, and Mobile Symposium*, *Dicomo'20*, NO.3B-4, June 2020.

3. Q. Xia, A. Wada, T. Yoshii, Y. Namioka, and T. Maekawa: A supporting technique for comparative analysis of factory work by skilled and unskilled workers using neural network with attention mechanism, In *Proceedings of the Multimedia, Distributed, Cooperative, and Mobile Symposium*, *Dicomo'21*, July 2020.(To appear).

3. Domestic Conference Paper

1. Q. Xia, J. Korpela, T. Maekawa, A. Wada, Y. Namioka: Preliminary investigation of an accelerated algorithm for unsupervised assembly-work activity recognition with acceleration sensors, In *Proceedings of the Institute of Electrical Engineers of Japan, Materials Information System Research*, IS-24, page 61-67, May 2019.

Abstract

Technologies of inertial sensors are developing fast, with those sensors being available in wearable devices, enabling various human activity recognition systems by using sensor data collected from humans. In the industrial domain, sensor data collected from individual workers will be crucial for smart manufacturing to achieve professional management, improving productivity from two aspects: (1) optimizing work processes and (2) optimizing skill training for workers.

In line-production systems, which are common in factories, each factory worker repetitively performs a predefined work process, with each period (one iteration of a complete work process) consisting of a sequence of operations. This thesis aims to recognize the starting times and duration (or end times) for individual operations in a work period by analyzing sensor data collected from the workers in an unsupervised manner to help factory managers optimize work processes.

To recognize the starting times and duration for individual operations, this study explores two approaches based on data characteristics. We propose the similarity-based approach, which leverages (i) semantic similarities between operations predicted from the process instruction document, and (ii) sensor-data similarities in an individual operation between consecutive periods, in addition to (iii) trend changes in the sensor data detected by a trend change detection algorithm, and (iv) frequent sensor-data patterns (motifs) discovered in the overall assembly work processes. We evaluated the proposed method using sensor data from six workers collected in actual factories, achieving a recognition accuracy of 80% (macro-averaged F-measure).

The other is the structure-based approach that focuses more on frequent sensor data motifs, which correspond to particular actions performed by the workers that appear in every iteration of the work periods. Specifically, we propose to track two types of motifs: period motifs and action motifs, during the unsupervised recognition process. A period motif is a unique data segment that occurs only once in each work period. An action motif is a data segment that occurs several times in each work period, corresponding to an action that is performed several times in each period. Tracking multiple period motifs enables us to roughly capture the temporal structure and duration of the work period. Action motifs, which are spread throughout the work period, permit us to

detect the start time of each operation precisely. We evaluated the proposed method using sensor data collected from workers in actual factories and achieved state-of-the-art performance.

In addition, to discover skill knowledge from workers, industrial engineers have manually identified activity differences between skilled and unskilled workers by watching video recordings or sensor data to obtain skill knowledge. However, a factory has many workers, and manual comparison between pairs of workers is time-consuming. This thesis aims to identify acceleration sensor data segments with activity differences to help industrial engineers find skill knowledge efficiently to optimize skill training.

To identify candidate segments with activity differences, we propose an attention based neural network to visualize the importance of segments within input acceleration data that contribute to classifying the input data into a skilled or unskilled class, which is useful to identify activity differences between skilled and unskilled workers. Our proposed method consists of three phases: (1) network training for the skilled/unskilled worker classification and for attention layers that emphasize the input segments with significant activity differences, (2) detecting activity differences, which uses attentions to find input segments to select candidates of segments containing activity differences, and (3) identifying corresponding activities of the candidates in input acceleration data from the other worker class. For instance, when an action of screwing by a skilled worker is identified by the attention mechanism, the corresponding sensor data segment of the screwing action by an unskilled is identified in his sensor data. Finally, the system outputs every candidate of activity differences and the corresponding segments of the other worker class. We evaluated our method using four workers' data in two types of works and ask an industrial engineer to assess the performance of identifying candidates of activity differences by the attention mechanism.

Contents

1 Introduction **1**

1.1 Research Target and Goal 3

1.2 Challenges 5

1.3 Approach 6

1.3.1 Unsupervised Factory Work Activity Recognition 6

1.3.2 Identifying Activity Differences in Skilled/Unskilled Workers . 8

1.4 Research Contribution 8

1.5 Structure of the Thesis 9

2 Similarity-based Unsupervised Factory Activity Recognition **13**

2.1 Introduction 13

2.2 Related Work 17

2.3 Methodology 19

2.3.1 Problem Set-Up 19

2.3.2 Assumed Environment 20

2.3.3 Overview 20

2.3.4 Work Model 22

2.3.5 Motif Detection and Tracking 25

2.3.6 Segmentation 28

2.3.7 Tracking with Particle Filtering 29

2.4 Evaluation 36

2.4.1 Dataset 36

2.4.2 Evaluation Methodology 37

2.4.3 Results 38

2.5 Conclusion 46

3 Structure-based Unsupervised Factory Activity Recognition 47
3.1 Introduction 47
3.2 Related Work 52
3.3 Methodology 53
3.3.1 Problem Set-Up 53
3.3.2 Preliminaries 54
3.3.3 Overview 55
3.3.4 Discovering Motifs 55
3.3.5 Building Motif Temporal Structure 60
3.3.6 Tracking Period Motifs 61
3.3.7 Tracking Operation Start Times 65
3.4 Evaluation 70
3.4.1 Dataset 70
3.4.2 Evaluation Methodology 72
3.4.3 Results 72
3.4.4 Discussion 82
3.5 Conclusion 83

4 A Supporting Technique for Identifying Skill Knowledge 85
4.1 Introduction 85
4.2 Related Work 88
4.3 Methodology 89
4.3.1 Preliminaries 89
4.3.2 Method Overview 90
4.3.3 Network Training 91
4.3.4 Detecting Activity Differences 93
4.3.5 Identifying Corresponding Activities 94
4.4 Evaluation 98
4.4.1 Dataset 98
4.4.2 Evaluation Methodology 98
4.4.3 Results of Identifying Corresponding Segments 99

4.4.4 Performance of Network Structure 100
4.4.5 Discussion 100
4.4.6 Qualitatively Analyze Performance of Identifying Candidates . 100
4.4.7 Selection of Data Sets 104
4.5 Conclusion 104

5 Conclusion and Future Work **107**
5.1 Thesis Summary 107
5.2 Future Work 109

Acknowledgment **111**

List of Figures

1.1 An overview of the smart factory 2

2.1 Example acceleration data collected from a worker. Red, green, and blue lines show x-, y-, and z-axis data, respectively. The rectangles below show the ground truth labels of the operations. The duration of each operation period is about 120 s. Each work period consists of eight operations, which are defined in the process instructions. Red brackets show the occurrences of a motif detected by the proposed method. . . . 15

2.2 Overview of the proposed method 20

2.3 An example of how paths of estimated start times for each operation are generated using a particle filter. We start with an initial particle at some trend change between the first and second motifs, which is a candidate start time for the first operation. We then sample several particles from this initial particle (three in this example), which are the initial estimates for the start time of the second operation. These sampled particles are then weighted and resampled, leaving us with our most likely estimates for the start time (one in this example). The most likely estimates are then sampled to generate the estimates for the start time of the third operation. The particle filter continues in this way, iteratively generating the start times for each operation based on its estimates for the start time of the previous operation. 22

2.4 Example flow of operations in the tree structure. (Due to the proprietary nature of these operations, only abstracted names are shown.) 23

2.5 Example semantic similarity matrix and sensor data similarity matrix of a work period. (Due to the proprietary nature of these operations, only abstracted names are shown.) 24
2.6 Example segmentation result for acceleration data 29
2.7 Sampling, weighting, and resampling procedures in the proposed method 30
2.8 Image from video captured in the factory 37
2.9 Accuracies (F-measures) of the five methods calculated using the path of the particle with the highest score at the end of the final work period . 39
2.10 Example output from Proposed for the 5th and 6th periods of work process A 39
2.11 Example output from Proposed for the 7th and 8th periods of work process B 40
2.12 Example output from Proposed for the 3th and 4th periods of work process E 41
2.13 Accuracies (F-measure) of the five methods calculated from the path of the particle with the highest score at the end of each work period 41
2.14 Transitions in the accuracies (F-measure) of the methods calculated from the path of the particle with the highest score at the end of each work period 42
2.15 Example output from “w/o Sens” for the 8th and 9th periods of work process C 44
2.16 Example output from “w/o Motif” for the 7th and 8th periods of work process B 45

3.1 Example acceleration data collected from a worker. The red, green, and blue lines show the x-, y-, and z-axis data, respectively. The rectangles below show the ground truth labels of the operations. The duration of each operation period is about 59 s. Each work period consists of six operations, which are defined in the process instructions. The red brackets show the occurrences of an example motif. “Optional 1” shows an optional operation. 49

3.2 Example acceleration data collected from a worker. The duration of each operation period is about 130 s. Each work period consists of eight operations, which are defined in the process instructions. Three period motifs are extracted in this example. m_{xy} shows the y-th occurrence of the x-th motif. 50

3.3 Overview of the proposed method 55

3.4 Calculating the similarity between segments of time series data. (a) Example original sensor data, (b) calculating the similarity between a motif and consecutive data segments using a sliding window, and (c) resulting output time series of similarity values. 57

3.5 Comparing the similarity time series for candidate period motifs. (a) The raw three-axis acceleration data. (b) The similarity time series for a good candidate for a period motif. (c) The similarity time series for a bad candidate for a period motif. The vertical dotted green lines show the start times for work periods, i.e., start time of the first operation. The vertical dotted red lines show the start time of each operation (except the first operation). The locations of the motif templates are indicated with stars. 58

3.6 Comparing the similarity time series for candidate action motifs. (a) The raw three-axis acceleration data. (b) The similarity time series for a good candidate for an action motif. (c) The similarity time series for a bad candidate for an action motif. The vertical dotted green lines show the start times for work periods, i.e., start time of the first operation. The vertical dotted red lines show the start time of each operation (except the first operation). The locations of the motif templates are indicated with stars. 58

3.7 Overview of the procedures used when tracking period motifs based on particle filters. (a) We start with an initial particle at the first occurrence of the period motif. We then sample several particles from this initial particle (three in this example), which are the initial estimates for the second occurrence of the motif. These sampled particles are then weighted and resampled (one in this example), leaving us with our most likely estimate(s) for this occurrence. The most likely estimates are then sampled to generate the estimates for the third occurrence of the motif. (b) During the sampling phase, new particles are generated at a distance of Δt from the previous particle, with Δt sampled from a distribution $\mathcal{G}$ that is based on the similarity time series $\mathbf{S}$ with the likelihood of sampling a particle at t proportional to the similarity value at t. (c) The $n+1$-th occurrence of a particle is weighted using a combination of a similarity score and a consistency score. The similarity score is the similarity value at the $n+1$-th particle. The consistency score measures the consistency of the temporal distance between the $n+1$-th occurrence of the current motif pf, which corresponds to a raw sensor data segment, and the $n+1$-th occurrence of the base motif pf_b when compared to their temporal distances in past work periods. 62

3.8 Overview of the procedures used when tracking operation start times based on particle filters. (a) Starting from an initial particle representing our estimated start time for the first operation, we sample several particles (three in this example), which are the initial estimates for the start time of the second operation. These sampled particles are then weighted and resampled, leaving us with our most likely estimate(s) for the start time of the second operation (one in this example). (b) During the sampling phase, the temporal distance between the particles for the n-th and $n+1$-th operations is $\Delta t + \Delta o$. Δt is sampled from a distribution of the duration of the n-th operation described in $\mathcal{M}$. Δo is sampled from a uniform distribution over $[0, t_{ol}]$. t_{ol} is an estimate of the duration of outliers, which is calculated as the extended time of the temporal distance between period motifs m_1 and m_2 compared to that in the temporal structure model $\mathcal{T}$. m_1 and m_2 are motifs just before/after the start time of the n-th operation. (c) Weighting phase of the particle filter for a particle corresponding to the start time of the $n+1$-th operation. The trend-based score for the particle is calculated based on the temporal distance from the closet trend change. The period-motif score for the particle is calculated based on the consistency of the temporal distances from adjacent period motifs. The action-motif score for the particle is calculated based on the consistency of statistical features extracted from the similarity time series corresponding to the estimated n-th operation. 66

3.9 Image from video captured in the factory 71

3.10 F-measures of the methods. Paired t-tests were used to confirm any significant differences between the results for each method. $*p < 0.05$. 73

3.11 Example output from "Proposed" for the third and fourth periods of work process F 73

3.12 Example output from "Proposed" for the third and fourth periods of work process D 74

3.13 Example output from "Xia2019" for the third and fourth periods of work process F 74

3.14 Transitions in F-measures of each method. The (*) in the horizontal axis indicates periods with outliers. 76
3.15 Example output from "W/o-Period" for the first and second periods of work process F 78
3.16 Example output from "W/o-Action" for the first and second periods of work process E 79
3.17 F-measures of the "Proposed" method for work process A when using best action motif (Proposed) and other action motifs (Motif_1, Motif_2 and Motif_3) that randomly selected. 80
3.18 Transitions in F-measures for each method for work process A when using best action motif (Proposed) and randomly selected action motifs (Motif_1, Motif_2 and Motif_3). 80
3.19 Example output from "Proposed" for the second and third periods of work process A. 81

4.1 An example of a series of operations collected from two workers' right wrists by using a 3-axes accelerometer. The rectangles indicate an activity difference between workers, which implies a skill knowledge about working posture to reduce body burden 86
4.2 Overview of the proposed method 90
4.3 Overview of the neural network 91
4.4 Procedure of identifying corresponding segments 95
4.5 MSE of the three methods on two datasets 99
4.6 Clustering results of latent representations for "Screwing" dataset, different color shows different clusters. (a), (c), (e), and (g) show latent representations visualized by PCA. (b), (d), (f), and (h) show clustering results visualized on raw acceleration data of right wrists. 101

List of Tables

2.1 Overview of our dataset 36

3.1 List of the symbols used in the following descriptions 56
3.2 Overview of our dataset 71
3.3 Computation time [s] of "Proposed" 84
3.4 Computation time [s] of Xia2019 84

4.1 Summary of hyper-parameter values 92
4.2 Overview of recorded datasets 98
4.3 Skill information for the dataset of Screwing 106
4.4 Skill information for the dataset of Final check 106

Chapter 1

Introduction

With the availability of inertial sensors embedded in commercial wearable devices, such as the accelerometer and gyroscope in Fitbit, Apple Watch, etc., various kinds of activities can be identified using the motion data collected from people. These motion data can be used not only to identify regular activities (e.g., step counting [13]), but also to identify relatively complicated activities, such as cooking [58] and factory work activities [41].

Besides the industrial production is in the midst of a digital transformation. To achieve greater efficiency in manufacturing, factory workers are expected to be equipped with appropriate sensors to collect various data of interest, enabling real-time interaction between the workers and factory managers through the analysis of data. In particular, the development of sensing technologies applied to mobile and wearable devices enables the factory managers to better monitor and visualize an entire production process and workers working in the process, making the manufacturing smarter [74, 62]. Figure 1.1 shows a prototype of a "Smart Factory" working in the future. In the physical site, the work machines are applied to make repeated productions that can analyze and diagnose issues without human intervention. The human workers are employed to perform flexible productions, whose motion data are collected by wearable sensors for analysis. In the cyber site, the manager can make management strategies based on the data collected from the physical site, which will benefit the physical site in return. There will be various approaches applied to processing the data from the physical site, and the result output by the approaches can be helpful for factory managers to make manage-

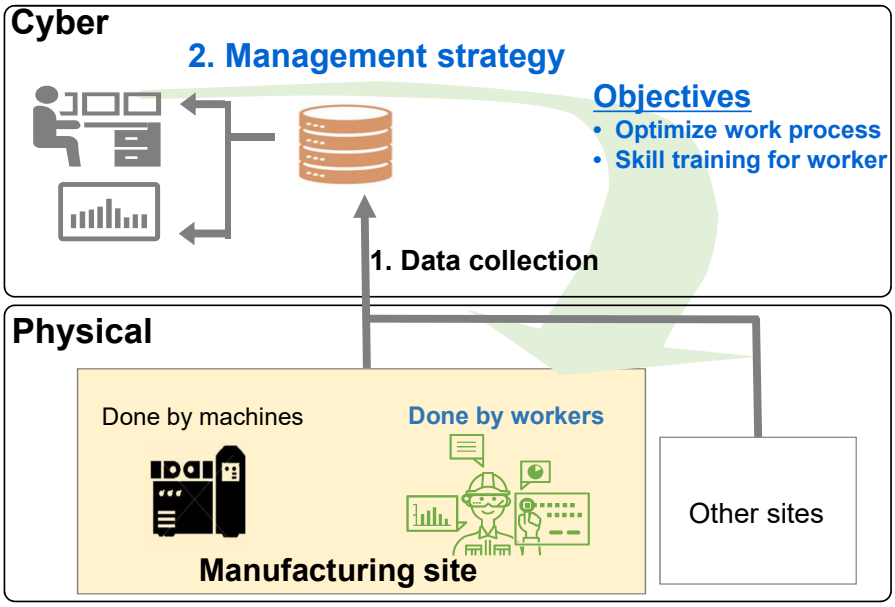


Figure 1.1: An overview of the smart factory

ment strategies from two aspects: (1) optimizing work process and (2) optimizing skill training for workers. With the guidance of the management strategy, the productivity of the manufacturing can be further promoted.

In this thesis, we develop several approaches to identify the workers' activities and their inherent skills to supplement the management strategy making.

To **optimize the work process**, it is important to know the starting times and duration of every activity performed by workers in the work process, which can support us to find the bottleneck of the process and make strategies to reduce the time spent on the bottleneck activities. To automatically detect the starting times and duration of every activity in all work processes, many of the earliest researches leverage supervised learning approaches [66] to recognize each activity with motion data collected from every worker, whereas it is costly to label data for each activity and individual. To address the problem of data labeling, this thesis proposes a novel way of recognizing activities in an unsupervised manner, which uses information of process instruction documents and features of similarity in sensor data as well as the similarity in data structures to recognize every activities performed by the workers.

To **design a better skill training**, it is important to efficiently extract skill knowl-

edge from workers. Since the skill level of workers in a factory are different, which will influence the work productivity, to help the unskilled workers develop their skills fast is an urgent need for factory managers. One of the most practical ways for the managers is to identify skill knowledge from the skilled workers, and then arrange skill training courses for the unskilled workers. However, efficiently extracting skill knowledge from skilled workers is a challenging task. Many of the earliest studies require skilled workers to do self-scaling or questionnaires to report their skills, while the results reported by workers are not accurate, and some unconscious skill knowledge hidden in workers' behaviors cannot be detected in such ways [21]. Besides, it is time-consuming to manually identify skill knowledge by factory managers, as there are many workers in a factory, and it is not easy to locate the specific timings that contain skill knowledge. This thesis contributes a supporting method to extract candidates of segments in acceleration data collected from factory workers, which is likely to contain skill knowledge. In particular, an attention-based neural network structure is applied to emphasize and map the input segments of workers' sensor data with significant differences between workers, enabling managers to efficiently extract skill knowledge from these selected segments.

1.1 Research Target and Goal

In this thesis, we focus on the issues of factory workers working in the assembly production line. First of all, a concrete description of information relates to the work is listed here.

- Operation: an operation is defined by the factory manager that consists of several activities. For example, in a "Setting" operation, the factory worker needs to move a box to the working table, switch a button and attach a label to a notebook.
- Work period: a work period is a complete iteration of a work process consisting of several operations with an order predefined by factory managers. If the first operation in a process is "Setting," a period starts from the "Setting" and ends before the occurrence of "Setting" in the following process.
- Process instruction: a process instruction is a document that lists the (1) descrip-

tion and standard duration of operations and (2) the order of operations in a period by words.

- Outlying operation: an outlying operation is a series of activities which is not described in the process instruction. It has less contribution to the work and should be suppressed. For example, if a worker made a mistake and then corrected the mistake, called an outlying operation.

- Work skill: the work skill is hidden in different activities among factory workers. The skill levels between workers are different.

The factory workers are iteratively performing work periods, and every individual is assumed to wear an inertial sensor device on each wrist, which is applied to acceleration data collection.

The **first research goal** of this thesis is to recognize the starting time and duration of every operation done by each worker using their acceleration data, which contributes to the improvements in the work process. For example, it is possible to know the bottleneck of the work process based on the average duration of operations. If the average duration of an operation is longer than expected, the manager can know if there is a problem in that operation. Or, the manager can identify that there is a room for improvement (e.g., shortening the duration) in an operation. Besides, if we can detect outlying operations in addition to standard operations, it is possible to reduce the occurrence of outlying operations by investigating the reason that caused the outlying operations in work periods. For example, if a worker usually waits for a while (an outlying operation) after he finished a period, the factory manager can assign more workers upstream of the production line to reduce the worker's waiting time.

The **second research goal** of this thesis is to detect candidate segments in work periods from skilled and unskilled workers, which are likely to contain skill information. This result contributes to the optimization of skill training. We believe that skill knowledge is more likely to exist in the significant activity differences between skilled and unskilled workers. Suppose the activity differences can be successfully detected by using the acceleration data of the workers. It will be more efficient for the manager to identify skill knowledge from the detected activity differences rather than searching around the whole data. For example, when a candidate segment corresponding to the

"Screwing" operation for a skilled worker is detected, by comparing it with the similar activities with unskilled workers, the factory manager quickly knows that the skilled worker can cooperate with both hands to control tools and bring new screws at the same time (one skill knowledge). In contrast, the unskilled worker separates these two activities, making the overall time consumption longer than the skilled worker. As a result, these identified skill knowledge can be added to the skill training courses to help the unskilled workers promote their skills faster.

1.2 Challenges

As mentioned above, this thesis uses an unsupervised approach to identify the starting times and duration of every operation done by factory workers using acceleration data, which reduces the time costs on data labeling. However, it is a challenging task to identify the starting times of operations in an unsupervised manner, which mainly has three challenges listed as follows.

- **Same operations in different periods**. The sensor data of the same operations in different periods performed by an individual worker is different since people cannot repeat exactly the same activity, which causes the duration and waveform of the same operations in different periods differently.

- **Optional operations in some periods**. There are some optional operations in work periods, resulting in the order of operations in some periods different from the other periods. For example, after the "Measuring" operation, it follows an optional operation of "Recording" in case that the size of the testing product is not qualified. The sensor data of the period with optional operations is different from other periods.

- **Occurrence of outlying operations**. The random occurrence of various types of outlying operations makes the problem challenging. An outlying operation could be a waiting status if a worker performs faster than the upstream workers, or a repeating action if a worker made a mistake. So, identifying the occurrences and duration of the outlying operation is difficult.

In general, the video data is not commonly used in factory work activity recognition as the data collected in real factories usually has many obstacles at the workers' body part. However, identifying activity differences by acceleration data is also very challenging for two reasons, which are listed as follows.

- **Unclear activity difference in sensor data**. It is challenging to identify activity differences by simply comparing sensor data differences. As even the same activities performed by an individual worker in different periods are different in sensor data
- **Individual differences for the same activities**. It is challenging to detect the corresponding activities on the other worker (e.g., if a screwing action that found in a skilled worker is identified as a candidate segment with significant activity difference, it is difficult to detect the segment of the corresponding screwing action on the unskilled worker), since workers with different skill-levels perform the target activities (i.e., the screwing action performed by both workers) in different patterns, making the sensor data different.

1.3 Approach

According to the two research goals and the corresponding challenges illustrated in Section 1.1 and Section 1.2, respectively, this thesis provides several approaches to address the problems, which are listed as follows:

- Unsupervised factory work activity recognition with acceleration data, which contributes to optimizing work process.
- A supporting technique for comparative analysis of factory work by skilled and unskilled workers using neural network with attention mechanism, which contributes to optimizing skill training.

1.3.1 Unsupervised Factory Work Activity Recognition

To help factory managers to optimize the work process, this thesis leverages an unsupervised learning approach to identify starting times and duration of every operation,

which reduces the effort on data collection, and is easy to implement on new tasks. In this thesis, we explore two different approaches to identify the starting times of every operation. Then the duration of each operation can be roughly calculated from its starting time to the starting time of the next operation. The main idea of the two approaches are described as follows.

Similarity-based approach: In Chapter 2, we address the problem of identifying starting times of operations by focusing on two similarities: i) sensor data similarity between different periods and ii) similarity between sensor data and word descriptions in instructions. The *sensor data similarity between different periods* is based on the idea that the sensor data of the same operations in different periods performed by an individual worker should be similar. As the work periods are repeatedly performed by the worker, the work pattern of the worker will stabilize, leading the activities of the same operations in different periods to become consistent. The *similarity between sensor data and word descriptions in instructions* illustrates that the operation similarities in sensor data and word descriptions have a similar trend. For example, one operation contains pushing red and blue buttons in the instruction, and another operation contains pushing red and green buttons. The semantic similarity between the descriptions of the operations is supposed to be high. Similarly, the sensor data of these two operations performed by the same worker are supposed to be similar. In the proposed method, we generate many candidates of segmentations for every period of data, with each segment corresponding to an estimate of the starting and ending times of an operation. The information regarding the expected sensor data similarity, which can be predicted from the semantic similarity in the instruction, is useful to evaluate the candidates. Chapter 2 contributes to the approach of generating segmentations of the data and evaluating them based on the two similarities.

Structure-based approach: In Chapter 3, we address the problem by using the consistent data structure of the work periods. We observe that some actions in a specific operation only perform once in every period. These characteristic actions can be applied to locate the operations in the period. By detecting several characteristic actions in a period, it is possible to sketch the structure of the period, including the temporal distance between these actions and the location of their corresponding operations in the period. Because workers iteratively perform work periods, the sketch corresponding to different periods done by the same workers should be similar. Therefore, We use the consistency

of sketch in every period to identify abnormal periods with the occurrence of outlying operations. Besides, the inner structure of a sketch for a work period can roughly locate the corresponding operations in the period. Finally, similar to the similarity-based approach, we generate many candidates of segmentations for the period data, and evaluate each segmentation according to the structure-based features. Chapter 3 promotes the approach of generating segmentations of the data by considering outlying operations, and improves the efficiency of scoring using the consistent structure of periods.

1.3.2 Identifying Activity Differences in Skilled/Unskilled Workers

Chapter 4 presents an attention-based neural network structure, which is an autoencoder combined with multiple Long Short-Term Memory (LSTM) layers, to address the above two challenges for the second research goal.

To identify significant activity differences between skilled and unskilled workers, a layer-wise attention component is applied on every LSTM layer to emphasize and map the most important timings as the candidates of segments, which is likely to contain significant activity differences. To detect corresponding activities on the other worker, a clustering approach is employed to cluster latent representations of the autoencoder. After training the neural network, the latent representation features become able to differentiating workers but will still be similar in similar activities. Therefore, we leverage the clustering approach to roughly eliminate the differences between workers and cluster similar activities into the same clusters.

1.4 Research Contribution

This thesis makes the following contributions.

In Chapter 2:

- We propose an unsupervised factory-activity recognition method for wearable sensors. To our knowledge, this is the first study on recognizing factory activities without the use of labeled training data.
- The similarity-based approach leverages information extracted from process instructions as well as sensor data analysis to precisely recognize activities in an

unsupervised manner.

- We evaluate the proposed method using sensor data collected from six workers in actual factories. Using this data, the proposed method achieved 80% recognition accuracy.

In Chapter 3:

- The structure-based method that makes use of the temporal structure of two types of motifs: period motifs and action motifs to robustly recognize factory activities in an unsupervised manner.
- Our method showed significant improvement in activity recognition accuracy over the similarity-based method when using real-world sensor data collected from factory workers, when outlying operations were included in the sensor data.
- As a side effect of the structure-based method for tracking multiple motifs, we can estimate the start times of operations with lower computation costs without the use of the dynamic time warping (DTW), with our method showing a significant improvement in computation speed when using real-life sensor data collected in a factory.

In Chapter 4:

- Our method provides an attention-based neural network to automatically detect candidates of activity differences between skilled/unskilled workers in a large amount of data, which reduces the time costs for managers spending on navigating key activity differences.
- Our method proposes a novel way of detecting corresponding activities between skilled/unskilled workers by using a clustering approach for the latent representations of the neural network to eliminate individual differences.

1.5 Structure of the Thesis

- Chapter 1 - Introduction
  We provide an introduction to this thesis. The questions that this thesis seeks to

answer, the challenges of the problems the core idea and major contributions are displayed in this chapter.

- Chapter 2 - Similarity-based Unsupervised Factory Activity Recognition
  We introduce a possible solution for recognizing factory activities in an unsupervised manner by using acceleration data and information of process instruction in Section 2.1. In Section 2.2, we first review activity recognition studies that use wearable sensors, including factory activity recognition studies. In Section 2.3 we present the proposed activity recognition method. We introduce the assumed experiment environment with sensor settings and information of process instructions. Next, an overview of the proposed method is displayed, consisting of three steps. (1) Detecting a characteristic sensor data motif that occurs only once in each work period to roughly identify the duration of work periods, (2) segmenting sensor data according to the state changes of the data to estimate starting times of operations, and (3) tracking with particle filtering to detect the starting times of operations in every period. In Section 2.4, we describe the testing environment and evaluate the performance of our proposed method by using acceleration data collected in real factories. Finally, we conclude our work and discuss future work in Section 2.5. The study in this chapter is based on our works published in [76].

- Chapter 3 - Structure-based Unsupervised Factory Activity Recognition
  We introduce a different direction of recognizing factory activities by using the structure of sensor data motifs in Section 3.1. Section 3.2 reviews the related work on time-series motifs applied on activity recognition. In Section 3.3, we introduce our proposed method, which consists of four sections, that is (1) detecting two kinds of motifs (i.e., period and action motifs) with an initial time of sensor data, (2) building motif temporal structure for period motifs, (3) tracking period motifs using particle filters, and (4) tracking the starting times of operations. In Section 3.4, we compare the proposed method in this chapter with the proposed method introduced in the previous chapter to discuss the performance of the similarity-based and structure-based approaches in different scenarios. Finally, we conclude our work and discuss future work in Section 3.5. The study in this chapter is based on our works published in [75].

- Chapter 4 - A Supporting Technique for Identifying Skill Knowledge
  We introduce our work in detecting activity differences between skilled and unskilled workers based on an attention mechanism in Section 4.1. Section 4.2 reviews the related work on extracting skill knowledge and identifying sensor data features using neural networks. In Section 4.3, we introduce the overview of our proposed method, which is composed of three phases: (1) training the network to generate features to differentiate the skilled and unskilled workers, (2) detecting candidates of segments with activity differences by using attention values of the network, and (3) identifying the corresponding segments (i.e., the corresponding activities in the same operation) on the other worker class by comparing the similarity of latent representations of the network. We evaluate the accuracy of identifying corresponding segments and qualitatively analyze the result of detecting candidates of segments in Section 4.4. Finally, we conclude our work and discuss future work in Section 4.5. The study in this chapter is based on our works published in [77].

- Chapter 5 - Conclusion and Future Work
  We conclude this thesis with a summary of its contributions and with a discussion of possible directions for future works.

Chapter 2

Similarity-based Unsupervised Factory Activity Recognition

In this Chapter, we describe our work on recognizing the starting times and duration for individual operations in work processes through analysis of i) sensor data similarity between different periods and ii) similarity between sensor data and word descriptions in instructions.

2.1 Introduction

Due to the recent proliferation of smart sensing devices such as smartphones, smart watches, and smart glasses, human activity recognition using sensor data obtained from body-worn smart devices has been actively studied in the ubicomp and wearable-computing research communities [28, 43, 72, 60]. Activity recognition using these wearable sensors can be applied to a variety of applications in both industrial settings, e.g., quality assurance for assembly workers, and home settings, e.g., analysis of daily living conditions of elderly patients. This chapter focuses on the application of activity recognition in an industrial setting, namely in the analysis of assembly work by factory workers using data captured by wrist-worn acceleration sensors.

Many factories employ a line-production system in which each product passes through the same sequence of work processes in series. Often included amongst these processes is assembly work conducted manually by factory workers, where each worker repeti-

tively performs a set of predefined processes with each process consisting of a sequence of operations, such as setting a board on a workbench and screwing parts onto the board. Such assembly work by factory workers still constitutes the core of line-production systems, making the improvement of assembly work one of the most important aspects of increasing productivity in these systems [2, 34].

In this and next chapters, we explore different approaches of recognizing the individual operations conducted during an overall production process using wearable sensor data. More specifically, we identify the starting time and duration of each operation by analyzing the different aspects of features of the sensor data. In particular, this chapter mainly focuses on the similarity of data in different work periods and the similarity between sensor data and word descriptions in instructions. Having detected the timing of the operations, we can then support many useful applications, such as monitoring the status of workers and detecting outlying operations. Such automated monitoring systems for assembly-work activities are in high demand by manufacturers, since line-production systems often employ large numbers of workers and it can be difficult for line managers to monitor all their workers simultaneously.

Previous work in the ubicomp community has applied supervised learning techniques to recognize operations using wearable sensor data. However, collecting labeled training data such as is required by these supervised approaches can be prohibitively costly when applied to assembly-work activities due to the following issues:

1. Each worker may perform a different operation process, necessitating training data collection for every worker.

2. The operations themselves may be modified frequently (e.g., weekly or monthly) due to frequent revisions of the production system, necessitating training data collection on a regular basis.

In this chapter, we address these shortfalls by recognizing operations in an unsupervised manner, which utilizes the similarity features of the line-production data to reduce the effort on data labeling.

Figure 2.1 shows example sensor data collected from a worker. As shown in the figure, a typical work process is iterated several times, with each iteration of the process (i.e., work period) comprised of a sequence of operations. In this example, we

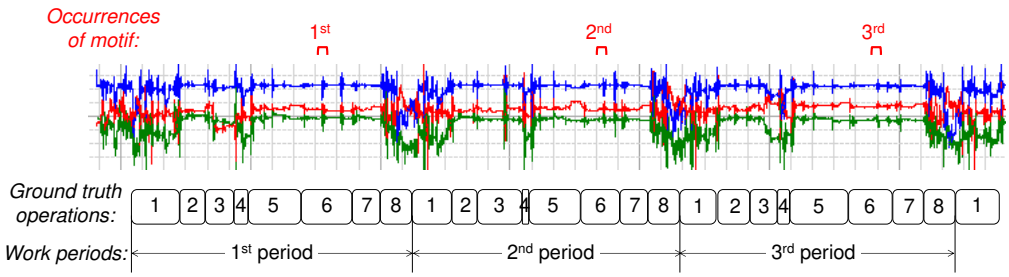


Figure 2.1: Example acceleration data collected from a worker. Red, green, and blue lines show x-, y-, and z-axis data, respectively. The rectangles below show the ground truth labels of the operations. The duration of each operation period is about 120 s. Each work period consists of eight operations, which are defined in the process instructions. Red brackets show the occurrences of a motif detected by the proposed method.

have eight operations per work period, iterated three times. The goal of this study is to identify the starting time and duration for each operation, as well as to estimate class labels for the operations. However, as can be seen in the figure, it is difficult to identify the sensor data segments corresponding to different operations solely using sensor data. The main feature of the proposed method is to leverage information included in process instructions in conjunction with the sensor data. Process instructions are documents that describe the flow of the operations included in each work process, as well as give detailed instructions for each operation. In this chapter, we assume that these instructions include information about the standard duration of each operation in a process, which we can use when identifying the starting time and duration of the individual operations. However, in practice the actual duration of each operation will differ from one work period to the next as shown in Figure 2.1, meaning that the process instructions alone cannot be used to accurately identify the boundaries between operations.

In this work, we employ a particle filter to robustly identify the boundaries of the operations with various durations, where the weight of each particle can be measured by the similarity features of the line-production data. A particle filter [54] is usually used to track an object in the computer vision research community. We employ a particle filter to robustly track the starting times of operations that appear non-linearly. We propose the following techniques to enable the precise unsupervised recognition of assembly-work operations based on a particle filter:

1. We note that the transition points between operations in a work period may contain noticeable trend changes in the sensor data, e.g., in Figure 2.1 we can see a dramatic change in the characteristics of the sensor data at each ending of the first operation. We can identify candidates for these transition points based on the trend changes detected by a nonparametric Bayesian hidden Markov model (HMM) and then determine the best sequence of transition points from these candidates by matching them to the standard operation durations using a particle filter.

2. We observe that operations with semantically similar descriptions in the working instructions may have similar sensor data. For example, description text of "screwing component A" operation can be semantically similar with that of "screwing component B" operation. And since both operations contain a screwing action, we expect that their sensor data will also be similar. Therefore, we can compare the similarities in the operations' sensor data for each candidate segmentation with their semantic similarities in order to identify the best candidate segments of the operations.

3. It is likely that the sensor data of an operation from one work period will be similar with data from the same operation in recent periods from the same worker. Thus, we can calculate the sensor data similarity for an operation across different work periods in order to identify the correct sensor data segments for the operation.

The research contributions in this chapter are summarized as follows.

- We propose an unsupervised factory-activity recognition method for wearable sensors. To our knowledge, this is the first study on recognizing factory activities without the use of labeled training data.

- The proposed method leverages information extracted from process instructions as well as sensor data analysis to precisely recognize activities in an unsupervised manner.

- We evaluate the proposed method using sensor data collected from six workers in actual factories. Using this data, the proposed method achieved 80% recognition accuracy (F-measure).

In the rest of this chapter, we first review activity recognition studies that use wearable sensors including factory-activity recognition studies. We then present the proposed activity recognition method and evaluate the method using sensor data collected in actual factories.

2.2 Related Work

Due to the recent advance in sensing technologies, a variety of sensors are used to recognize human activities such as inertial sensors [8, 61], microphones [15, 27], cameras [44, 40, 57], and magnetic sensors [39, 38]. In particular, human activity recognition using acceleration sensors is a core topic in the ubicomp community because accelerometers have a good trade-off between prediction accuracy and minimal device-maintenance overhead.

Many prior examples have applied supervised learning techniques on acceleration data in order to recognize daily activities [36, 37, 12, 32]. Several studies have dealt with the costs incurred when collecting the labeled training data needed for supervised learning. For example, Morales et al. [50] employed transfer learning to reuse kernels in convolutional neural networks (CNNs) across different domains. Rueda et al. [51] designed a CNN with parallel branches, with a branch processing data from each inertial sensor, and confirmed that the proposed network outperformed a baseline CNN when using a small amount of sensor data from the logistics domain. Maekawa et al. [43] reused labeled sensor data from source users with similar physical traits to those of a target user to train the target user's activity model.

Work has also been done on using unsupervised learning to accomplish activity recognition, such as in Huynh et al. [22] where they clustered activity data in an unsupervised manner using topic models. Ye et al. [80] transfered activity labels across different partially labeled datasets from smart home environments that had distributed sensors. They leveraged home ontologies that describe the semantic relationships between sensor-enabled objects/places to determine the semantic similarities between distributed sensors in different environments, which are used to find sensor data patterns in different environments representing the same activity. Sanabria et al. [65] also employed home ontologies and variational autoencoders to achieve unsupervised domain adaptation to

align the feature spaces of different environments. Krüger et al. [30] applied computational state space models (CSSMs), which have an ability to replace training data with prior knowledge, to human activity recognition. The authors confirmed that the CSSMs outperformed a training-based model (HMM) when the amount of information from training data is limited. Yordanova [81] focused on structured behavior such as cooking and automatically constructed a knowledge-based behavior model from instruction text that was used as a Planning Domain Definition Language (PDDL).

Due to the recent growing interest in smart manufacturing and Industry 4.0 [9, 35, 59], studies on recognizing and supporting factory work using sensor technologies [1, 3, 4, 6, 26] have been attracting attention. Of particular note are the previous studies done on monitoring and analyzing factory work using wearable sensors. One such study is Koskimäki et al. [29] in which they obtained acceleration and gyroscopic data from a wrist-worn inertial sensor device and analyzed operation processes in a line-production system to ensure that all necessary operations were performed. They recognized activities such as hammering and screwing using a kNN search. Another such study is Ward et al. [73] in which they obtained acceleration and audio data from a wrist-worn device to recognize woodworking activities by using HMMs and a linear discriminative classifier. Stiefmeier et al. [68] focused on assembly work on automobiles and used inertial sensors attached to several body parts, such as the upper and lower arms, to classify sensor data segments by computing the distance between the collected segments and sensor data templates prepared in advance. Examples of the activities used in their work include opening an engine hood and opening a trunk. Stiefmeier et al. [67] also studied the similar task of bicycle repair and used motion sensors and ultrasonic hand tracking to recognize maintenance activities using HMMs. Reining et al. [63] used CNNs to identify feature representations of activities performed during the order-picking process. They employed CNNs to process motion capture data to recognize specific motions that are considered as characteristic features of multiple activities in the order-picking system. Tao et al. [71] employed CNNs to recognize six different standard assembly operations performed consecutively; they converted multi-channel IMU data and electromyographic data into images and trained the CNNs on the images. All of these methods for analyzing factory work relied on supervised machine learning approaches and so required the creation of labeled training data.

The most relevant recent work is Maekawa et al. [41] in which they measured the

duration of each work period on a production line in an unsupervised manner. Their method tracked a motif (sensor data segment) that appears only once in each work period using a particle filter to estimate the duration of the periods. Our work uses this method to roughly capture the structure of work periods during our preprocessing stage.

2.3 Methodology

2.3.1 Problem Set-Up

The goal of this study is to find a set of timings, which is represented as $D_{T,estimation}$, where a timing $t(p_i, n, k) \in D_{T,estimation}$ represents an estimated starting time of the *n*-th operation in the *k*-th period, on a time-series that has a minimal overall error to their corresponding ground truth operation start times $D_{T,groundtruth}$.

This study first generates a set of candidate timings by using an HMM algorithm to reduce computation costs. Note that the number of candidates in this set is still larger than the number of ground truth operations. In order to find the best set $D_{T,estimation}$, with each timing in this set has the smallest temporal distance to its corresponding operation start time, this chapter introduces a particle filter to track with timings t in the candidate set to find the best set efficiently. We start from a candidate timing $t(p_i, 1, 1)$, generating a particle p_i to represent an estimation starting time of the first operation. Then, we generate several particles to represent the estimation starting time of the next operation, which are generated according to the standard duration of the first operation. After that, many sets were generated, with each set being evaluated by several criteria proposed in this chapter. This chapter focuses on criteria mainly designed based on the similarity between sensor data segments and the semantic similarity in the instructions. Because $D_{T,groundtruth}$ is unknown, we find $D_{T,estimation}$ that maximizes the criteria. Therefore, the performance of the method will be greatly affected by the criteria, and these criteria do not guarantee that the obtained $D_{T,estimation}$ yields the smallest error to $D_{T,groundtruth}$. Therefore, we evaluated the criteria by investigating if the proposed criteria achieve a small error using actual data.

In the following sections, we will introduce the proposed method of finding the best $D_{T,estimation}$.

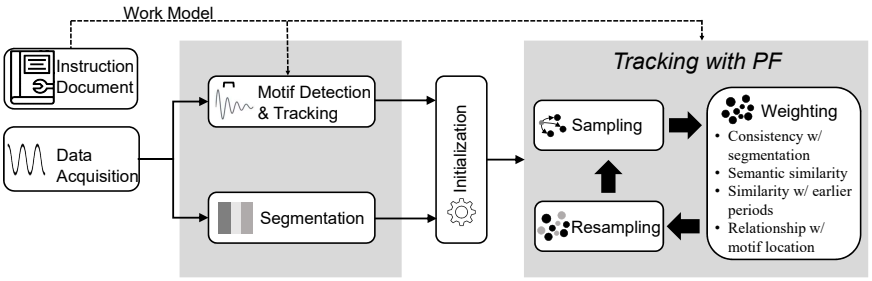


Figure 2.2: Overview of the proposed method

2.3.2 Assumed Environment

Sensor Setting

Our method assumes that workers will be wearing a body-worn inertial sensor, i.e., a three-axis accelerometer. In our experiment, the workers wore a smart watch on their right wrists, with the smart watches' accelerometers collecting data at a sampling rate of approximately 60 Hz.

Process Instructions

Process instructions are prepared in advance by a line manager for each work process to be done by a worker. They specify the flow of operations included in a work process, e.g., (1) place a board on the workbench, (2) check the test equipment mode, and (3) change the signal monitor mode, with some work processes containing optional operations, e.g., (4) pack accumulated boards into a box once 10 have been completed. These instructions also describe and specify the standard duration of each operation. For example, an operation named "preliminary preparation" had the following description: "Place the module in front of the body and fix it to the jig. Then connect a communication module and pick up the screwdriver." Note that this is translated from the original Japanese text used as input for our method.

2.3.3 Overview

Figure 2.2 shows an overview of the proposed method. We start with a time series of acceleration data that has been collected from a worker that corresponds to multiple

iterations of that worker's work period. We then identify and track a motif that occurs once per work period in this sensor data. This motif is a unique data segment that occurs only once in each work period, e.g., the data corresponding to a hammering action that is performed only once in each work period. Since this motif occurs only once per work period, the period of sensor data spanning between the first and second occurrences of this motif must contain the starting time of the second work period, as shown in Figure 2.1.

In addition to the large-scale segmentation done using motifs, we also segment the sensor data on a smaller scale using a nonparametric Bayesian HMM. As can be seen in Figure 2.1, visible trend changes are evident at the transition points between many of the operations. Such trend changes can be detected using HMMs and can be considered as possible start times for individual operations in the work period, including the start time of the first operation in the work period, i.e., the start time of the overall work period.

Having used the motifs to identify the region of sensor data containing the start time for the second work period and used the trend changes to detect possible start times for each operation, we then run our particle filter. First we initialize our particle filter by generating an initial particle at a discovered trend change, representing a likely start point for the second work period (i.e., the start point of the work period's first operation). Then, as shown in Figure 2.3, particles are generated (sampled) along possible paths from the work period's start point through all the individual operation start times by iteratively sampling, weighting, and resampling at each estimated operation start time to generate estimates for the next.

The particle filter begins with the sampling phase, in which it uses a work model that encapsulates data derived from the process instructions to generate candidate start times (particles) for the following operation in the work period from an initial particle. In the weighting phase, the candidate start times (particles) are assigned weights based on (i) trend changes in the sensor data detected by HMM, (ii) semantic similarities between operations based on our work model, (iii) data similarities between operations and corresponding operations in preceding work periods, and (iv) the consistency of the motif's location in the current work period with its location in the previous work period. In the resampling phase, the particle filter resamples the candidate particles based on their weights, leaving it with the most likely start times for that operation. Each

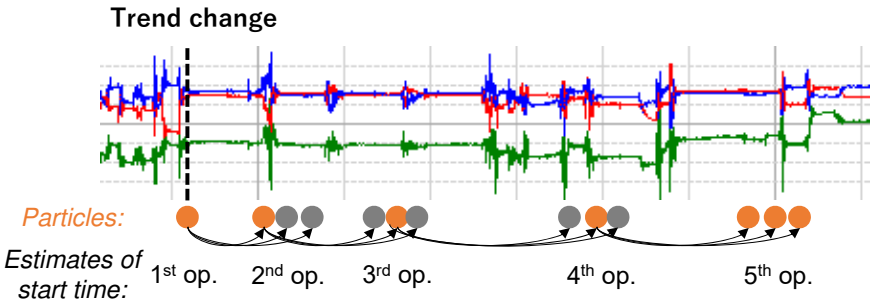


Figure 2.3: An example of how paths of estimated start times for each operation are generated using a particle filter. We start with an initial particle at some trend change between the first and second motifs, which is a candidate start time for the first operation. We then sample several particles from this initial particle (three in this example), which are the initial estimates for the start time of the second operation. These sampled particles are then weighted and resampled, leaving us with our most likely estimates for the start time (one in this example). The most likely estimates are then sampled to generate the estimates for the start time of the third operation. The particle filter continues in this way, iteratively generating the start times for each operation based on its estimates for the start time of the previous operation.

of these remaining particles then serve as initial particles when starting the sampling phase of the next iteration. At the end of each work period, we choose our estimated sequence of operation start times as being the path generated by our particle filter with the highest weight, and the duration of each operation can be roughly estimated by the temporal distance between its corresponding particle and the following particle (e.g., the estimated duration of the first operation in Figure 2.3 can be calculated as the temporal distance between the particles of 1st and 2nd ops, which correspond to estimated start times for the first and second operations, respectively.).

2.3.4 Work Model

The first step in our process is to construct a work model that encapsulates the information obtained from the process instructions. It consists of two main parts: a tree-structured representation of the flow of operations and a matrix-structured representa-

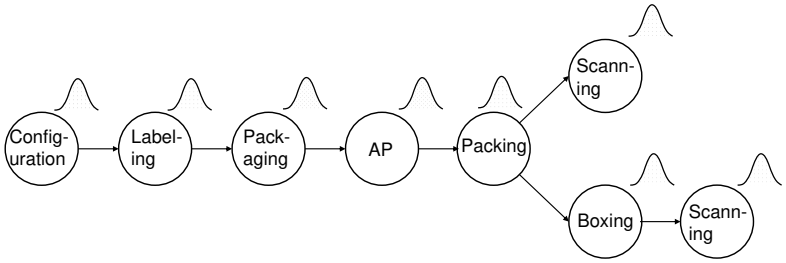
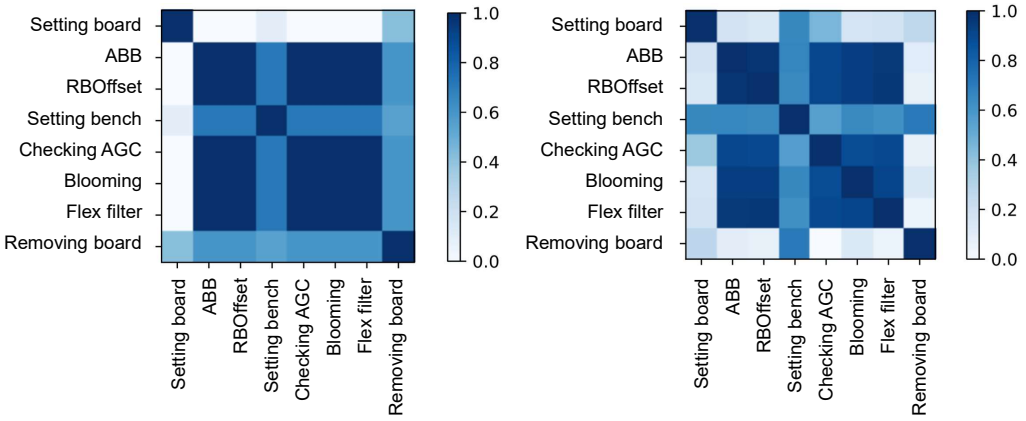


Figure 2.4: Example flow of operations in the tree structure. (Due to the proprietary nature of these operations, only abstracted names are shown.)

tion of the semantic similarity between the operations' description texts. The flow of operations is represented as a tree structure (Since there is no child node connecting to multiple parent nodes, the flow of operations is represented as a tree structure. In general, the flow of operations can be represented as a Directed Acyclic Graph structure.), whose root node corresponds to the first operation with at least one additional node in the tree structure for every possible operation in the work period, as shown in Figure 2.4. Edges connect each operation with the operations immediately preceding and following it, with branches possible in the case of optional operations, e.g., the branch before the optional "Boxing" operation in Figure 2.4. Each tree node contains information about the expected duration of the operation of interest, modeling this duration as a lognormal distribution. This operation flow model is then used to sample the durations of operations when generating particles to use as candidate start times for operations. Specifically, because our work model extracts only simple information, i.e., the flow of operations, from the process instruction document, it takes only a few minutes to construct the work model by factory managers.

The semantic similarity matrix allows us to estimate the sensor data similarity between operations based on the semantic similarity of their descriptions in the process instructions. Because our process instructions are written in Japanese, we first perform a morphological analysis of the descriptions based on [31], during which we tokenize the Japanese descriptions (i.e., perform text segmentation) and extract the verbs and nouns from the descriptions to use as keywords for the description. We capture the semantics of these keywords by generating a word vector for each keyword using a word2vec



(a) Semantic similarities between operations (b) Sensor data similarities between operations

Figure 2.5: Example semantic similarity matrix and sensor data similarity matrix of a work period. (Due to the proprietary nature of these operations, only abstracted names are shown.)

model [47] that is trained externally using the Japanese edition of Wikipedia[1] to obtain a word vector that represents the semantics of that keyword. The keywords for each description are combined by computing the mean vector for all word vectors generated from that description, which is then regarded as the semantic vector for that description. Finally, we compute the cosine similarity between each pair of mean vectors to construct our semantic similarity matrix for the operations.

Figure 2.5 (a) shows an example of a semantic similarity matrix for a work process. As is shown in the matrix, the semantic similarities between the "ABB," "RBOffset," "Checking AGC," "Blooming," and "Flex filter" operations are all high. All of these operations relate to manipulating dials when testing modules on a board. Figure 2.5 (b) shows a similarity matrix for the same work, with this matrix calculated using dynamic time warping [52] to compare ground truth sensor data segments of the operations. Here we can see the same overall trend in operation similarities as is seen in the semantic similarity matrix, with the dialing operations all showing relatively high similarity to each other. Therefore, the semantic similarity matrix can be used to evaluate each potential segmentation of a work period based on how well the sensor data similarities between

[1] http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/

segments match our expectations based on their semantic similarities.

2.3.5 Motif Detection and Tracking

The first step in identifying the start time for an overall work period is to identify and track a motif that occurs only once in each work period based on the method proposed in [41]. This method takes a time series of acceleration data and the standard duration of a work period from our work model as input and outputs timestamps for the locations of the most likely motif found in the sensor data (see Figure 2.1 for an example of this output). This motif corresponds to some characteristic action that is completed once per work period, e.g., a drilling action that is performed only once in each work period. Because the starting time of the n-th work period exists between the $n-1$-th and n-th occurrences of the motif, as shown in Figure 2.1, we can then narrow our search for this starting time to the area between these two occurrences of the motif. Algorithm 1 shows the algorithm used during motif detection.

Distribution of Durations for Work Periods

When searching for our motif, we need to model the duration $t \in \mathbb{R}$ (lead time) of the work period based on the work model we generated from the process instructions. We can model this lead time using a Gaussian mixture model (GMM) prior distribution $p(t)$ on t given by:

$$p(t) = \sum_{i=1}^{N} \pi_i \mathcal{N}(t, \mu_i, \sigma_i), \tag{2.1}$$

where N is the number of mixtures, π_i is the mixture weight of the i-th multivariate Gaussian distribution of the GMM (with $\pi_i = 1/N$ in our implementation), and μ_i and σ_i are the mean and variance of the i-th Gaussian distribution which is referred to as $\mathcal{N}(t, \mu_i, \sigma_i)$, respectively. The number of mixtures N represents the total number of possible patterns for the lead time (duration) in the work period. In the example shown in Figure 2.4, there are two possible patterns for the lead time because of the optional operation "Boxing." Therefore, N corresponds to the number of leaf nodes in the process flow tree, with each of the N distributions representing a complete path from the root node to one of the leaf nodes in the tree. The mean of the i-th distribution

Algorithm 1: `Motif detection and tracking`

Input: $\mathbf{X}$: time-series of acceleration, $\mathcal{G}$: Gaussian mixture model of work process

1 `/* Randomly generating motif candidates */`
2 $\mathbf{X}_{init} \leftarrow \mathbf{X}[0 : t_{init}]$
3 $\boldsymbol{M} \leftarrow$ EXTRACT-RANDOM-SEGMENTS$(\mathbf{X}_{init})$
4 $\boldsymbol{S} \leftarrow \emptyset$
5 **for** $\forall m_i \in \boldsymbol{M}$ **do**
6 `/* Tracking motif candidate using particle filter */`
7 $\boldsymbol{P} \leftarrow \{m_i\}$
8 **repeat**
9 `/* Sampling */`
10 $\boldsymbol{P}^- \leftarrow \emptyset$
11 **for** $\forall p_i \in \boldsymbol{P}$ **do**
12 $\boldsymbol{P}^- \leftarrow \boldsymbol{P}^- \cup$ SAMPLING$(p_i, \mathcal{G})$
13 `/* Weighting */`
14 **for** $\forall p_j \in \boldsymbol{P}^-$ **do**
15 $p_j.score \leftarrow$ WEIGHTING$(p_j, \mathbf{X})$
16 `/* Resampling */`
17 $\boldsymbol{P} \leftarrow$ ROULETTE-WHEEL-SELECTION$(\boldsymbol{P}^-)$
18 **until** *repeat the process until the end of* $\mathbf{X}$
19 $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \boldsymbol{P}$
20 `/* Scoring particles */`
21 **for** $\forall p_i \in \boldsymbol{S}$ **do**
22 $p_i.score \leftarrow$ SCORING$(p_i, \mathbf{X})$

Output: A particle in $\boldsymbol{S}$ with the maximum score

is then computed by summing the standard durations of all nodes from the root node to the the i-th leaf node in the tree.

Random Generation of Motif Candidates

We begin our motif search by randomly extracting motif candidates, i.e., sensor data segments with length w, from the sensor data between time 0 and time t_{init} (line 1 in Algorithm 1). Time t_{init} is a hyperparameter for the algorithm, set as half of the standard

lead time in our work. We then use a particle filter to generate the potential locations of subsequent occurrences for each candidate based on the lead-time distribution model described in Equation 2.1.

Tracking Motif Candidates Using a Particle Filter

Using the locations of the randomly generated motif candidates as our starting locations, we then search for potential locations where each of these motif candidates will reoccur in the data using a particle filter (line 6 in Algorithm 1). This particle filter initializes a particle at the exact location (timestamp) of each one of our randomly generated motif candidates, with each particle storing information about the motif candidate it represents, i.e., the sensor data segment for the motif. The particle filter then generates each motif's potential subsequent locations by iterating through phases of sampling, weighting, and resampling as follows.

Sampling: We randomly sample the subsequent occurrences of each particle based on the lead-time GMM $\mathcal{G}$ (line 9 in Algorithm 1). Assuming that $t(p_i, n)$ is the timestamp for a particle p_i generated during the n-th iteration of sampling, then the timestamp for some new particle p_j from among the m particles generated from p_i during the $n+1$-th iteration of sampling is determined using the following equation:

$$t(p_j, n+1) = t(p_i, n) + \Delta t,$$

where Δt is the estimated interval between occurrences of the motif candidate that is randomly sampled from the GMM $\mathcal{G}$. Each of these m particles generated from p_i are then our estimates for the possible $n+1$-th occurrences of the motif, with each new particle p_j storing time $t(p_i, n)$ along with the times for each of the previous particles in its chain (one from each of the n sampling iterations) as a history of the n occurrences of the motif candidate (in addition to storing the sensor data segment from the initial motif candidate p_a).

Weighting: After having sampled new particles based on the GMM, we assign each new particle a score as follows (line 13 in Algorithm 1). In brief, each particle p_j is assigned a score that is highest when the motif candidate p_a occurs at $t(p_j, n+$

1) and does not occur at any other time between $t(p_i, n) + w$ and $t(p_j, n + 1)$, exclusively. By doing so, we can identify motifs that appear only once in a work period. More specifically, when assigning a score to particle p_j, we compare the sensor data segment for p_a (with length w) to all windows with length w in the interval $[t(p_i, n) + w, t(p_j, n + 1) + w]$. This score is increased with increasing similarity of $[t(p_j, n+1), t(p_j, n+1)+w]$ to $[t(p_a, 1), t(p_a, 1)+w]$ and is decreased with increasing similarity of any segment of length w in $[t(p_i, n)+w, t(p_j, n+1)]$ to $[t(p_a, 1), t(p_a, 1) + w]$. For more details on calculating the scores used during weighting, refer to [41].

Resampling: The sampled particles are probabilistically resampled using roulette wheel selection (line 16 in Algorithm 1). In roulette wheel selection, items are selected one at a time with each item randomly selected according to a probability that is associated with that item. In the case of our method, the probability of selecting a particle is set as proportional to the score assigned to the particle during the weighting phase. Using this method, we can probabilistically select high-score particles and discard the remaining particles.

These three procedures are iterated to produce new generations of particles until reaching the end of the input sensor data.

Computing Final Scores and Selecting the Best Motif

The final step in motif detection is to select the best candidate motif (line 20 in Algorithm 1). Using the steps outlined above, we now have several candidate motifs for each of which we have several sequences of particles that represent possible occurrences of that motif across our sensor data. We first compute a single score for each of these candidate motifs, which represents the sequence of particles for that motif that had the highest weights. We then compare the candidate motifs based on these scores to select the best motif. For more details on motif scoring and selection, refer to [41].

2.3.6 Segmentation

In order to identify likely starting locations for work periods in the sensor data as well as likely starting locations for operations in the work periods, we segment the sensor

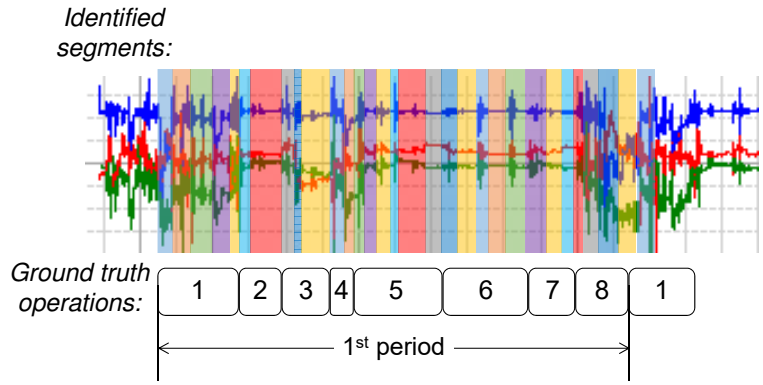


Figure 2.6: Example segmentation result for acceleration data

data based on trend changes in the data. We do this in an unsupervised manner by employing a Bayesian nonparametric version of HMM called the hierarchical Dirichlet process HMM (HDP-HMM) [23]. HDP-HMM is a nonparametric model that allows the number of states to be learned from the data. The HDP is used to infer a posterior distribution of the number of states of an HMM, permitting us to decode (segment) a time-series using an HMM without manually specifying the number of states of that HMM. For more details about HDP-HMM, refer to [23].

Figure 2.6 shows an example result from this segmentation method when applied to acceleration data. The overlaid color rectangles represent the segments generated by HDP-HMM, while the numbered rectangles below the data show the ground truth operations for the data. As can be seen in this figure, while HDP-HMM segmentation resulted in many false positives, many of the ground truth segmentation boundaries were included in its results. Therefore, we can use HDP-HMM to generate a list of candidate start times for the individual operations, which in turn can also be treated as candidate start times for the work period.

2.3.7 Tracking with Particle Filtering

Overview

Using the methods described in Section 2.3.5 and Section 2.3.6, we are able to obtain timestamps for the occurrences of a motif M and timestamps for trend changes (candi-

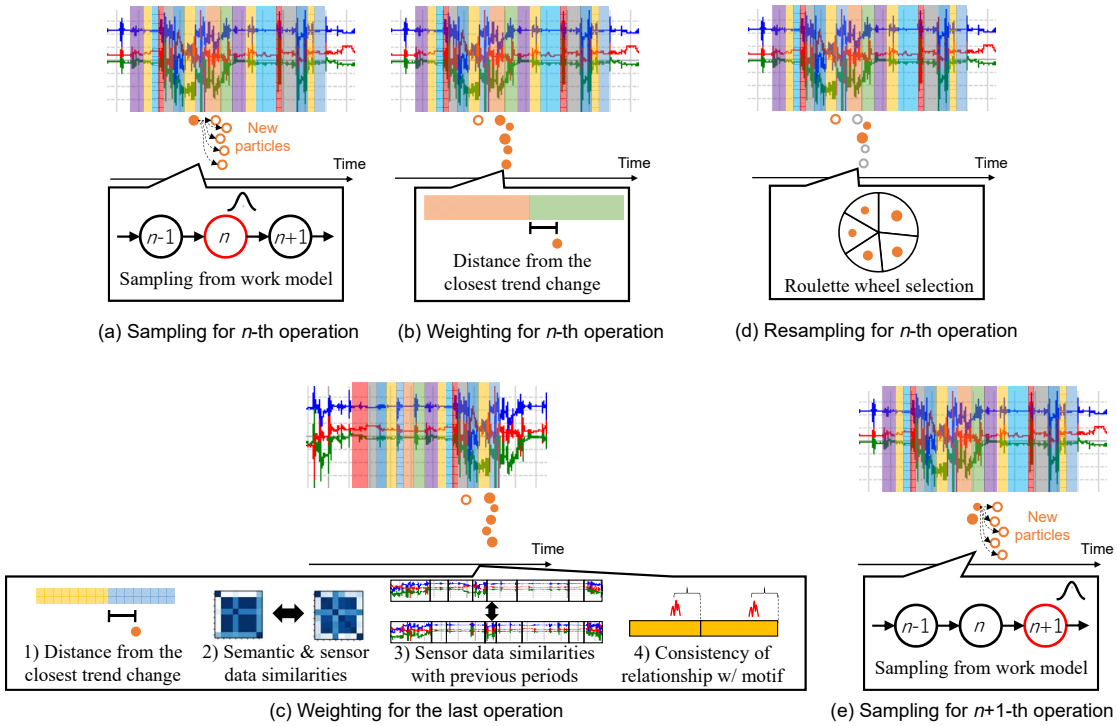


Figure 2.7: Sampling, weighting, and resampling procedures in the proposed method

date start times) C. We now want to use a particle filter to produce our estimate for the most likely starting time for our work period along with the starting times for all operations in that work period. We start by initializing this particle filter based on C and M to generate particles that act as candidates for the starting time of the work period. We then track each of these particles to identify the starting times of the other operations in the work period by iterating through the particle filter's three phases of sampling, weighting, and resampling as shown in Figure 2.3. Figure 2.7 illustrates the sampling, weighting, and resampling procedures, while Algorithm 2 shows the algorithm of this procedure.

Algorithm 2: Particle filtering for operation recognition

Input: $\mathbf{X}$: time-series of acceleration, $\mathcal{W}$: work model of work process, $\mathbf{M}$: timestamps of occurrence of motif, $\mathbf{C}$: timestamps of trend changes

```
/* Initialize particles */
P ← ∅ /* For initial particles */
P^F ← ∅ /* For final results */
for ∀c_i ∈ C do
    if M[1] < c_i < M[2] then
        P ← P ∪ {c_i}
/* Particle filtering */
for ∀p_j ∈ P do
    p_j.score ← 0
    p_j.period ← 1
    P^- ← {p_j}
    repeat
        for ∀p_i ∈ P^- do
            p_i.state ← W.root /* Set to initial state */
            p_i.period ← p_i.period + 1
        repeat
            /* Sampling */
            P^+ ← P^-
            P^- ← ∅
            for ∀p_i ∈ P^+ do
                P^- ← P^- ∪ SAMPLING(p_i)
            /* Weighting */
            for ∀p_j ∈ P^- do
                if IS-LEAF-NODE(p_j.state) then
                    if p_j.period > 2 then
                        p_j.score ← TREND-BASED-SCORE(p_j, C) +
                          SEMANTICS-BASED-SCORE(p_j, W, X) + SENSOR-BASED-SCORE(p_j, X) +
                          MOTIF-BASED-SCORE(p_j, M) /* Normalize before sum */
                    else
                        p_j.score ← TREND-BASED-SCORE(p_j, C) +
                          SEMANTICS-BASED-SCORE(p_j, W, X)
                else
                    p_j.score ← TREND-BASED-SCORE(p_j, C)
            /* Resampling */
            P^- ← ROULETTE-WHEEL-SELECTION(P^-)
            for ∀p_j ∈ P^- do
                p_j.state ← GET-NEXT-STATE(W, p_j.state)
        until repeat the process until the leaf nodes of W
    until repeat the process until the end of X
    P^F ← P^F ∪ P^-
```

Output: A particle in $\boldsymbol{P^F}$ with the maximum score

Initialization

As can be seen in Figure 2.1, the starting time of the second work period exists somewhere between the first and second occurrences of our motif. Therefore, we can generate initial particles at each timestamp from $\mathbf{C}$ that fall between the first and second occurrences of the motif (found in $\mathbf{M}$), which become our candidates for the starting time of the first operation in the second work period (line 1 in Algorithm 2). We also set the current state of each initial particle as the root node (first operation) of the operation flow from our work model.

Sampling

As shown in Figure 2.7 (a), with each iteration we sample new particles from each existing particle that represent estimates of the starting time for the following operation (line 17 in Algorithm 2). Assuming that $t(p_i, n, k)$ is the timestamp for an existing particle p_i that represents the starting time of the n-th operation in the k-th work period, then the timestamp $t(p_j, n+1, k)$ for a new particle p_j for the $n+1$-th operation is sampled from p_i as follows:

$$t(p_j, n+1, k) = t(p_i, n, k) + \Delta t,$$

where Δt is randomly sampled based on the duration-time distribution found in the currently selected node in the work model's operation flow (i.e., the n-th operation's node). Therefore, $t(p_j, n+1, k)$ is an estimate of the starting time of the $n+1$-th operation. Note that each new particle p_j stores time $t(p_i, n, k)$ along with the times of all previous particles in its chain as a history of the starting times for previous operations.

Weighting

Each particle generated in the sampling phase is then assigned a weight using a combination of four scores: a trend-based score, a semantics-based score, a sensor-based score, and a motif-based score (line 22 in Algorithm 2). We determine which of these scores to use depending on the current state of the particle, with the trend-based score always used, as shown in Figure 2.7 (b), and the semantics-based, sensor-based, and motif-based scores used when the state corresponds to the final operation of a work period, as shown in Figure 2.7 (c). These scores are calculated as follows:

Trend-based score: This score enables us to detect operation starting times based on the trend changes discovered by HDP-HMM. As shown in Figure 2.7 (b), this score is computed based on the temporal distance between the timestamp of a particle and the timestamp of the closest trend change in $\mathbf{C}$ as follows:

$$\textsc{CumulativeDist}(p_j, n, k, \mathbf{C}) = \sum_{m=2}^{n+1} \textsc{Dist}(t(p_j, m, k), \mathbf{C})),$$

where $t(p_j, m, k)$ is the timestamp for the ancestor particle of p_j corresponding to the m-th operation in the k-th work period (when m is equal to $n + 1$, it is the timestamp for p_j itself) and

$$\textsc{Dist}(t, \mathbf{C}) = \min_{c \in \mathbf{C}} |t - c|.$$

The inverse of this distance is then used as the score, with this score used each time we compute weights for particles.

Semantics-based score: Unlike the trend-based score that can be computed for each new particle, the semantics-based score can only be computed for particles representing the end of a work period, i.e., the end time of the final operation. This score enables us to compare two operations in a work period to see how well the similarities in sensor data for the two operations match with the semantic similarities in the descriptions for those operations. As mentioned in Section 2.3.4, we calculate the semantic similarity between each pair of descriptions for operations to form the semantic similarity matrix. We then concatenate all entries below the main diagonal in the matrix to form a semantic-similarity vector. As mentioned in Section 2.3.4, we also calculate a sensor-data similarity matrix for all pairs of operations, with the sensor-data for each operation extracted based on the timestamps of p_j and all of its ancestor particles leading back to the first operation ($n = 1$) in the same period. Again we concatenate all entries below the main diagonal to form a sensor-data similarity vector. Finally, we compute the cosine similarity between the semantic-similarity vector and the sensor-data similarity vector as the semantics-based score. Note that, when optional operations are not observed in the work period, we ignore them during the similarity calculation.

Sensor-based score: Like the semantics-based score, the sensor-based score is only computed for particles representing the end of a work period. It enables us to

compare two work periods by leveraging the expected similarity of sensor data for the same operation even when conducted in different work periods. We calculate the sensor data similarity for each operation using dynamic time warping (DTW) as follows:

$$DTW(\mathbf{X}(p_j, n, k), \mathbf{X}(p_j, n, k-1)),$$

where $DTW(\cdot, \cdot)$ calculates the DTW distance between two sensor data segments. Here, $\mathbf{X}(p_j, n, k)$ gives the sensor data segment of the n-th operation in the k-th work period identified by particle p_j. $\mathbf{X}(p_j, n, k-1)$ gives the sensor data segment of the n-th operation in the $k-1$-th work period identified by an ancestor particle of p_j corresponding to the n-th operation in the $k-1$-th work period. We compute the sum of the dynamic time warping distances and the regard the inverted value as the sensor-based score. Note that, to reduce computation costs, we standardize and then discretize the two sensor data segments, and then represent the segments as piece-wise approximations where the time-series is divided into equal-sized frames and the mean value of data within each frame becomes a representative value of the frame [33].

Motif-based score: Like the semantics- and sensor-based scores, the motif-based score is only computed for particles representing the end of a work period. Because the motif tracking method [41] finds motifs that occur only once per work period, motifs will occur with an interval corresponding to the lead time for the work period, with the location of a motif within any work period consistent across adjacent work periods (see Figure 2.1), even when some work periods contain optional operations (which is proven in [41]). Therefore, we can evaluate the quality of an estimated series of operation start times based on the consistency of motif locations in the estimated work periods. We start by computing two values for each occurrence of a motif that measure the temporal distances from the motif to the start time (d_s) and end time (d_e) of the work period. We then use these temporal distances to compare the k-th and $k-1$-th occurrences of a motif by computing the absolute difference between each value for k-th and $k-1$-th occurrences (e.g., $|d_s^k - d_s^{k-1}|$), using the smaller of these two absolute differences in the motif-based score.

Additionally, since the motif should be observed in sensor data segments corre-

sponding to the same operation in each work period (as shown in Figure 2.1), we also compute the temporal distance between its location and the starting/ending time of the operation in which it occurs. We then incorporate the absolute difference between these distances for the current work period and the previous work period as part of the motif-based score. Furthermore, when the motif does not appear in the same operation for the current and previous work periods, we apply a penalty to the score. This score enables us to reduce the overall score for estimated paths that do not match the standard lead time of the work period.

The four scores described above are computed at the end of each work period. The final score used for any given particle is then calculated as the cumulative sum of these four scores for the current particle together with these scores from all of its ancestor particles (i.e., the particles in its path from each previous work period).

Resampling

The sampled particles are probabilistically resampled using roulette wheel selection as shown in Figure 2.7 (d) (line 31 in Algorithm 2). We then set the current state for each remaining particle as the following operation in the operation flow (line 34 in Algorithm 2). In the case of a branch in the operation flow, we randomly select the branch to follow. Each of these remaining particles is then used to generate new particles during the sampling phase of the next iteration, as shown in Figure 2.7 (e). Note that, because we can only calculate the trend-based and semantics-based scores at the end of the second work period, the weighting used at this point yields less precise estimates of operation start times than are possible with later work periods. Therefore, at the end of the second work period, we resample two times as many particles as are sampled for later periods. In addition, because the durations of optional operations such as "boxing accumulated items" and "waiting on an item from the previous worker" often have high variance, we also resample two times as many particles for these operations as the other operations. Further, because an estimate of the ending time of the last operation of the k-th work period should exist between the k-th and $k + 1$-th occurrences of the motif as shown in Figure 2.1, we discard estimates (particles) for the ending time of the last operation that do not satisfy this condition.

We iterate through sampling, weighting, and resampling until reaching the end of the

Table 2.1: Overview of our dataset

	work A	B	C	D	E	F
# operations	11 (1)	8	6 (2)	11	7	7 (1)
standard duration [s]	50 (59)	124	55 (115)	52	55	46 (60)
periods	11	12	10	12	6	7
data duration [s]	655	1440	614	639	360	420
work overview	install screws on circuit board	test and record circuit board information	bag and box circuit boards	install screws on circuit board	check final product and record results	check final product and record results

sensor data, at which point we select the particle with the highest weight as representing the most likely sequence of start times for operations in the sensor data.

2.4 Evaluation

2.4.1 Dataset

We evaluated the proposed method using data collected from factory workers in actual factories. Table 2.1 gives an overview of the work processes observed for this study. The "# operations" row in the table shows the number of standard operations included in each work process, with the number of optional operations shown in parentheses. For example, work process A consists of 11 standard operations along with one optional operation. The "standard duration" row shows the standard duration of one work period for each work process, with the standard duration when optional operations are performed shown in parentheses. For example, the standard duration of work process A is 50 s without the optional operation and 59 s with the optional operation. The "periods" row shows the number of work periods included in the observed sensor data for each work process. The "data duration" row shows the total duration of collected sensor data for each work process. The standard duration of each operation was roughly determined by the actual duration of the operation; compute the average duration without outliers and keep the first decimal place.

The sensor data was collected using a Sony SmartWatch3 SWR50 attached to each worker's right wrist, with the acceleration data sampled at approximately 60 Hz. Video

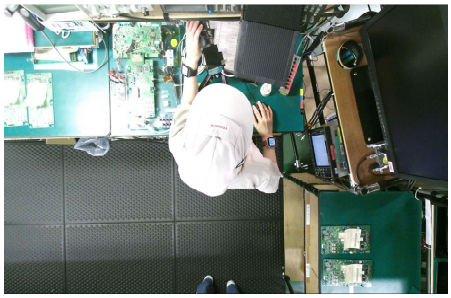


Figure 2.8: Image from video captured in the factory

was also recorded in order to obtain the ground truth information needed for evaluation, with Figure 2.8 showing an example of the images captured. All data processing was conducted offline in our laboratory after collection, which includes 6 datasets performed by different workers as shown in Table 2.1.

2.4.2 Evaluation Methodology

Our method provides the starting (and ending) times for all operations in each work period, allowing us to classify each sensor data point as belonging to an operation class (excluding data points in the first work period). During evaluation, we can then treat our results as classification results estimated per data point (acceleration sample), allowing us to calculate the macro-averaged F-measure of our results.

The goal of our experimental evaluation is to test the effectiveness of the proposed method. We did not discuss the performance of our method comparing to other supervised learning approaches in this scenario because preparing enough labeled training data is time-consuming in real factories, which is the main reason for not using the supervised learning approaches.

To evaluate the effectiveness of the scores in the proposed method, we prepared the following methods.

- **Proposed**: This is the proposed method.
- **w/o Sem**: The proposed method without the use of the semantics-based score during the weighting phase of the particle filter. Therefore, we assume a condition where description texts about operations are unavailable.

- **w/o Sens**: The proposed method without the use of the sensor-based score during the weighting phase of the particle filter.

- **w/o Motif**: The proposed method without the use of the motif-based score during the weighting phase of the particle filter.

- **only Trend**: The proposed method using only the trend-based score during the weighting phase of the particle filter. That is, this method uses neither the semantics-based score, the sensor-based score, nor the motif-based score.

Here we explain the way of selecting hyper-parameters of HDP-HMM. The maximum number of HMM states is set as the number of operations. The hyper-parameter of the prior distribution on state-transition parameters, which relates to the duration of an operation, is set as the data length divided by ten plus a bias.

2.4.3 Results

Recognition Accuracy

Figure 2.9 shows the classification accuracy for each method. Note that because the methods used in this study are based on particle filters, each method can produce several estimated segmentation paths per work period. The results shown in Figure 2.9 select a single path when calculating the F-measure by selecting the path corresponding to the particle with the highest score at the end of the *final* work period. Additionally, the F-measures for the "w/o Motif" and "only Trend" methods are zero for some work processes because all particles were discarded in the resampling phase at the end of a work period, i.e., no particles satisfied the condition that estimates for the ending time of the last operation of the k-th work period should be between the k-th and $k + 1$-th occurrences of the motif.

Overall, the Proposed method achieved the highest average F-measure across all six work processes (83.2%), even though it employs an unsupervised approach. However, Proposed did not achieve the highest F-measure for all six work processes, with the F-measure of "w/o Sem" higher than that of Proposed for several work processes, despite performing slightly worse than Proposed overall (81.5%). Also of note are the poor

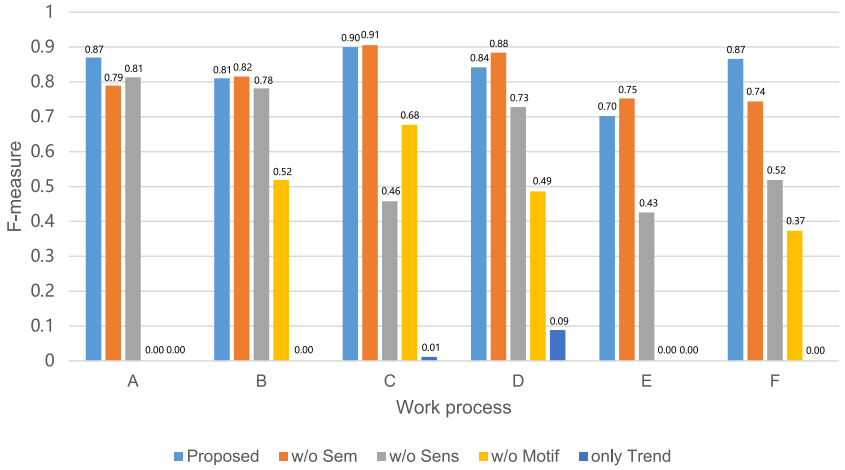


Figure 2.9: Accuracies (F-measures) of the five methods calculated using the path of the particle with the highest score at the end of the final work period

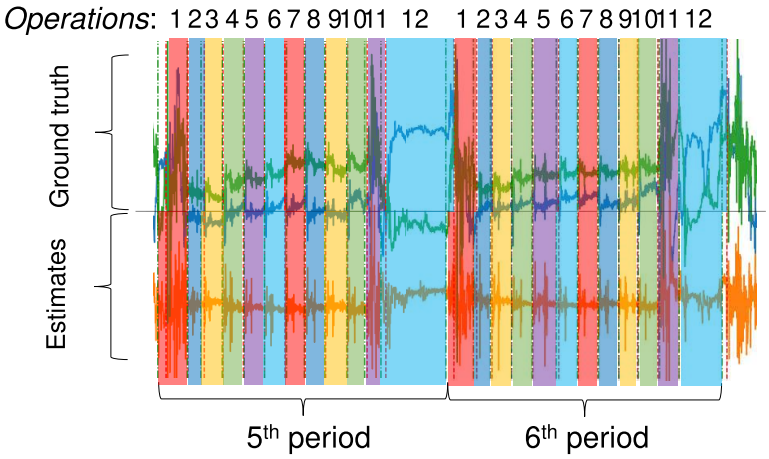


Figure 2.10: Example output from Proposed for the 5th and 6th periods of work process A

results from the only Trend method, which indicate that it is not possible to recognize the operations using only trend changes in the sensor data.

Figure 2.10 shows an example output from the Proposed method for the fifth and sixth periods of work process A. The individual operations are indicated by the colored bars, with the upper half of the bars indicating the ground truth segmentation and the

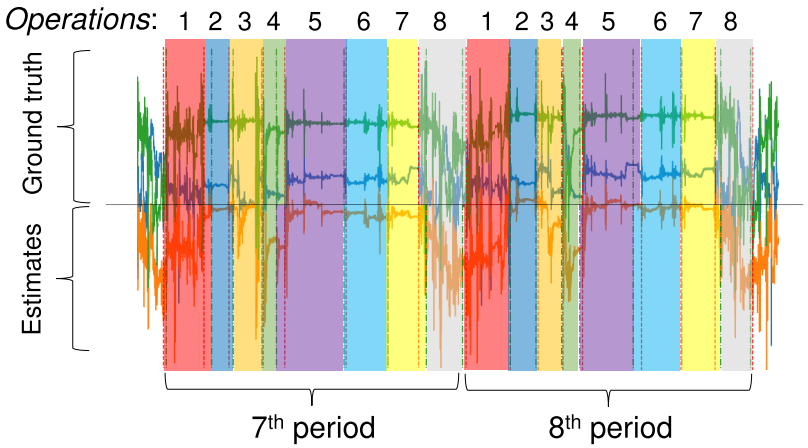


Figure 2.11: Example output from Proposed for the 7th and 8th periods of work process B

lower half of the bars indicating the estimated segmentation output by Proposed. The numbers in the upper margin correspond to the operation number for each ground-truth segment. From this figure, we can see that the estimates from Proposed match closely with the ground truths and that Proposed was even able to identify the optional operation (operation 12) found at the end of each period with only small errors in the ending and starting times of the neighboring operations (operations 11 and 1). Additionally, since the Proposed method leverages the correlation between semantic and sensor-data similarities, it was able to do well at identifying operations 2 to 10, which all correspond to screwing operations with high semantic and sensor-data similarities. Moreover, Figure 2.10 suggests that our method using trend changes could precisely predict start time of each operation. As shown in Figure 2.9, the accuracy of our method was high in all the data, indicating the effectiveness of using trend changes.

Figure 2.11 shows another example result from the Proposed method. In this case, the operations in the work process do not have similar sensor data like in work process A, but Proposed is still able to accurately identify the operations with only small errors by leveraging the trend-based, sensor-based, and motif-based scores.

Figure 2.12 shows an example output of Proposed for work process E, for which the F-measure of Proposed is somewhat poorer than for the other work processes. This may be due to the large variance in the durations of the operations in work process E, as can

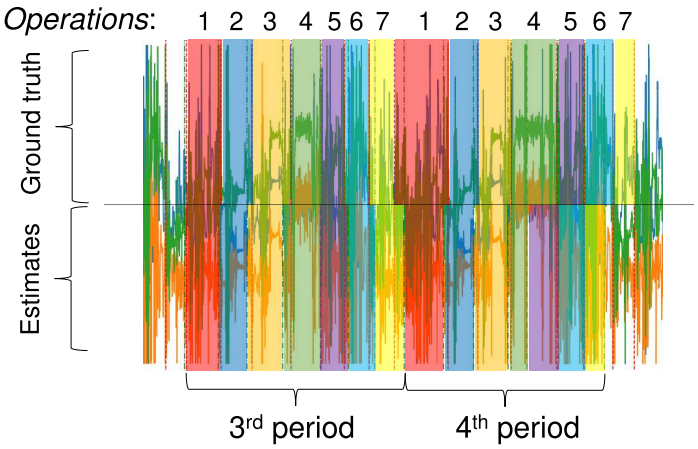


Figure 2.12: Example output from Proposed for the 3th and 4th periods of work process E

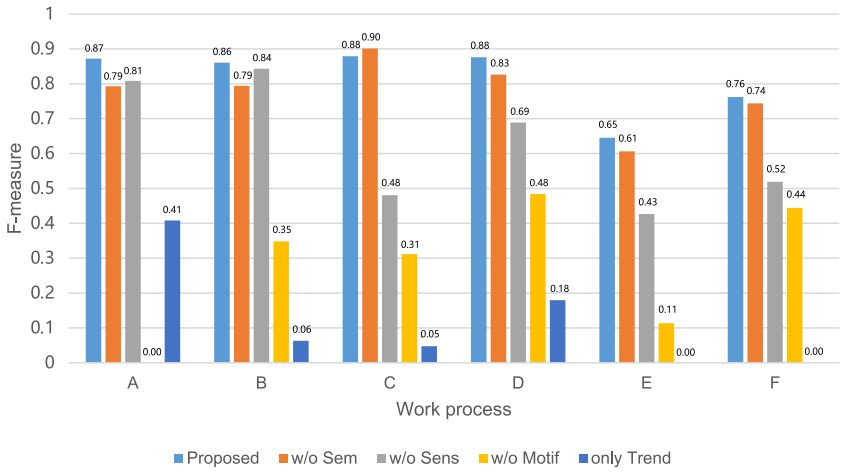
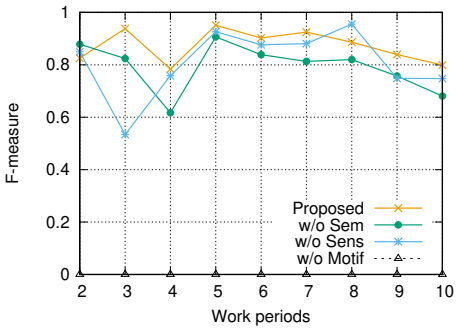


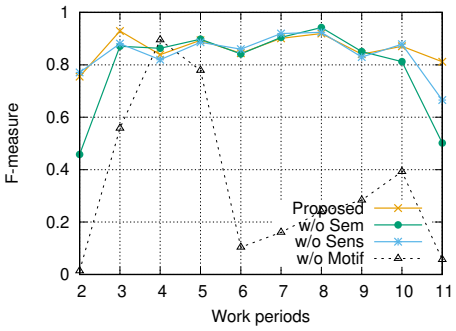
Figure 2.13: Accuracies (F-measure) of the five methods calculated from the path of the particle with the highest score at the end of each work period

be seen when comparing the consecutive segments for operations 1 or 4. Because this work process involves inspecting items, the duration of some operations will depend on condition of the items.

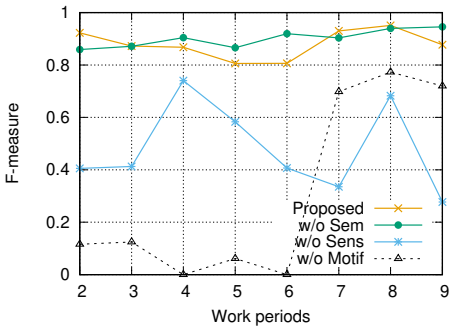
Figure 2.13 shows the average F-measure per work period for each method, with the path used for each work period selected as the path with the highest score at the



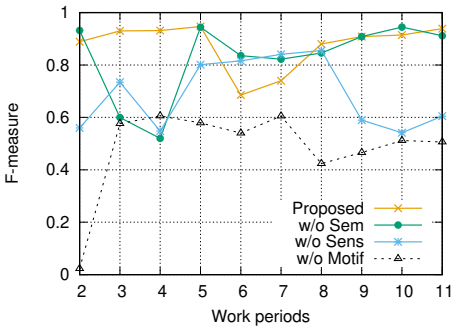
Results from work process A



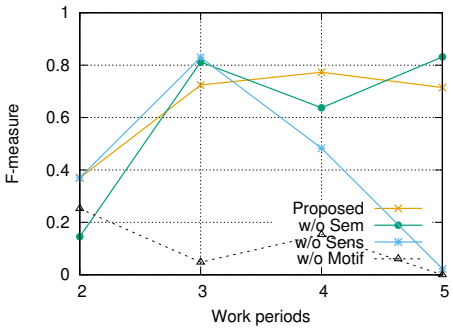
Results from work process B



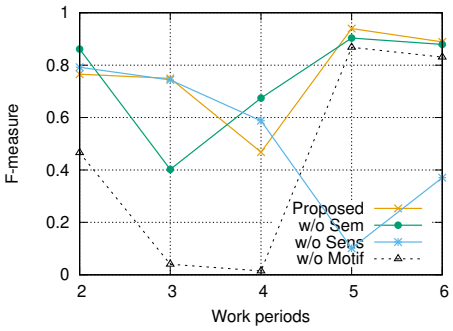
Results from work process C



Results from work process D



Results from work process E



Results from work process F

Figure 2.14: Transitions in the accuracies (F-measure) of the methods calculated from the path of the particle with the highest score at the end of each work period

end of that work period. This differs from Figure 2.9 in which the results are based on a single contiguous path across all work periods. Interestingly, the F-measure of Proposed is higher than that of "w/o Sem" and "w/o Sens" for many work processes in this figure. Comparing Figure 2.13's per-work-period averages to Figure 2.9's per-

work-process averages, we see that when we use fewer scores (i.e., "w/o Sem" and "w/o Sens"), we can still achieve high accuracy when we can consider the entire work process's sensor data when generating the estimates. In contrast, when we can use only the sensor data up to and including the work period of interest, using all the scores (i.e., Proposed) provides the most accurate estimates. Additionally, the higher F-measures in Figure 2.9 when compared to Figure 2.13 indicate that these methods could improve on their estimates from past work periods when processing an entire work process. This ability to improve on earlier estimates comes from how the timing of each work period is dependent on the previous work period, meaning that a contiguous path that does well at predicting the timing of given work period is likely to have done well at predicting at the very least the ending of the previous work period.

Transitions in Recognition Accuracy

Figure 2.14 breaks down the average F-measures from Figure 2.13 into the individual F-measures for each work period. Here we can see that the F-measure for the second work periods seems to be poor for many work processes. This is because we calculate the particle scores using only the trend- and semantics-based scores at the end of the second work period, while later work periods benefit from including the sensor- and motif-based scores.

In addition, we can find sudden decreases in the F-measure. This is mainly caused by the first occurrence of an optional operation. Because we cannot employ the sensor-based score for such operations, the F-measure significantly decreases in such cases. Further, as can be seen in the results, the transitions of Proposed seem to be more stable than those of the other methods. This robust estimation is achieved by increasing the diversity of input by incorporating all four score functions.

Effect of Semantic Similarity

As can be seen in the "w/o Sem" results in Figures 2.9 and 2.13, the F-measure of "w/o Sem" is higher for many work processes than those of "w/o Sens" and "w/o Motif." This indicates that the contribution of the semantics-based score is lower than the contributions of the sensor-based and motif-based scores. Although a semantic similarity matrix can often capture the overall trends in a work process, as shown in Figure 2.5, there is no

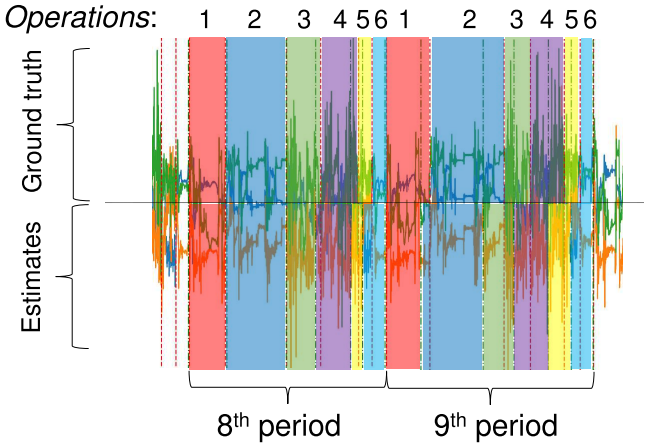


Figure 2.15: Example output from "w/o Sens" for the 8th and 9th periods of work process C

guarantee that the semantic similarities are equivalent to the sensor data similarities. In particular, the F-measure of Proposed for process C is poorer than that of "w/o Sem" in Figure 2.13. In this case, the nouns used in the process instructions of many operations in work process C were similar with each other despite the actions performed during the operations not being similar, resulting in a semantic similarity matrix that did not match well with the sensor data similarity matrix.

Effect of Sensor Data Similarity between Different Periods

Figures 2.9 and 2.13 also show the results for "w/o Sens," which does not employ the sensor-based score. As can be seen in these results, the F-measures of "w/o Sens" are poor for many work processes, indicating that the sensor-based score was an important aspect of the Proposed method. This is because the sensor-based score increases the consistency of estimated start times across work periods by comparing the sensor data for each operation from the current and previous periods. For example, Figure 2.15 shows an example result for "w/o Sens" for work process C, which was generated from the path of the particle with the highest score at the end of the final work period. As shown in this result, the operations identified by "w/o Sens" in consecutive periods are inconsistent. For example, segments of the second operation in the eighth and ninth periods identified by "w/o Sens" are not consistent, reducing the classification accuracy

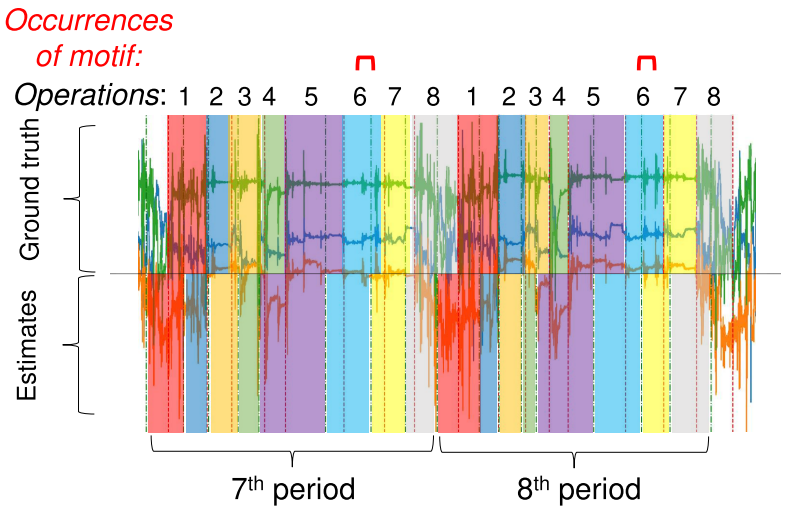


Figure 2.16: Example output from "w/o Motif" for the 7th and 8th periods of work process B

for later operations.

Effect of Relative Location of Motif

Looking at the "w/o Motif" results in Figures 2.9 and 2.13, we can also confirm the effectiveness of the motif-based score, since the F-measure of "w/o Motif" is poor for many work processes. The motif-based score ensures a consistent identification of the start times for the first operation and the operation containing the motif. Figure 2.16 shows an example result for" w/o Motif" for work process B, in which the motif used by our method was located in the sixth operation. Because "w/o Motif" does not consider the location of the motif, the segments identified as the sixth operation in the two periods are inconsistent, with the motif included in the sixth operation of the seventh period and in the seventh operation of the eighth period.

As above, we could confirm the effectiveness of our particle filter-based architecture and our proposed scoring methods using sensor data collected in actual factories. We also confirmed that the motif-based score and sensor-based score are necessary to achieve high recognition accuracy. In addition, we confirmed that even when text descriptions of the operations are unavailable, the proposed method (w/o Sem) can still

achieve high recognition accuracy.

2.5 Conclusion

In this chapter, we presented an unsupervised method for recognizing assembly work done by factory workers using wearable sensor data. The proposed method detects the starting times and duration of each operation included in a work process by leveraging information about (i) trend changes in the sensor data detected by HMM, (ii) semantic similarities between operations extracted from process instructions, (iii) data similarities between operations and corresponding operations in preceding work periods, and (iv) the consistency of a motif's location in the current work period with its location in the previous work period. We confirmed the effectiveness of the proposed method using sensor data collected from workers in actual factories. The Proposed method achieved a high average accuracy (F-measure) of about 80%.

The proposed method relies on the similarity features of the sensor data, which is challenging to identify the occurrences of unexpected events (i.e., outlying operations). In the next chapter, we plan to supplement our method by considering the consistency of sensor data structure among the work periods to cope with outlying operations that may arise in a factory.

Chapter 3

Structure-based Unsupervised Factory Activity Recognition

In this chapter, we explore another approach for identifying the starting times and duration for individual operations through the analysis of the consistent data structure of the work periods.

3.1 Introduction

Because activity recognition using inertial sensors can be applied to various applications in both home settings, e.g., analyzing the daily lives of people [45], and industrial settings, e.g., quality assurance for factory workers. In the last and this chapter, we focus on recognizing the starting times and duration of operations that performed by factory workers in the industrial settings, in which each product passes through the same sequence of work processes in series with workers positioned at various stages along the production line repetitively performing predefined sequences of operations. Three-axis acceleration data are collected from a commercial smartwatch worn on the workers' right wrists. In this chapter, we focus on using the consistent data structure of work periods to robustly recognize the starting times of operations for individual workers in an unsupervised manner.

Previous unsupervised factory activity recognition approaches are not robust to the outlier activities that usually occur in real factory scenario. These outlier activities occur

when factory workers stop their work for reasons such as: (i) walking to another position to pick something up (e.g., tools or materials), to socialize with another worker, or to use washing facilities, (ii) waiting for a product to arrive at their station, possibly due to a delay caused by a previous worker in the line, and (iii) responding to inquiries from either another worker or a manager that visits their station. For example, the similarity-based approach mentioned in the last chapter mainly considers the sensor data similarities of expected operations, which is challenging to recognize outliers because of no prior knowledge. The similarity-based method relies on a single sensor data motif that appeared only once per work period and tracked the occurrences of the motif in the following periods based on the predefined standard duration of period (standard lead time) in instructions. When outliers occur, the estimated occurrence of the next motif is still tracked according to the standard lead time, indicating that the estimated occurrence of the motif will be ahead of the expected occurrence of the motif.

Figure 3.1 shows an example of acceleration data collected from a factory worker's smartwatch. In this example, a typical series of operations are iterated three times, with each iteration (i.e., work period) comprising a sequence of six operations. This figure also shows the occurrences of an example sensor data motif (indicated by the red brackets). In this example, the motif corresponds to an action characteristic of the second operation, which is performed only once per work period. Here we can see that because outlier activities occur after the sixth operation in the second work period, the intervals between occurrences of the motif become irregular, making it difficult to estimate the third occurrence of the motif using only our prior information (i.e., standard lead time). In addition, it is impossible to know where in the sequence of operations that the outlier activities occurred, making it difficult to estimate the start times for the individual operations. Therefore, the similarity-based method can break down when encountering such outlier activities.

In this chapter, we develop a structure-based approach to recognize factory activities by leveraging the consistent structure of work periods to recognize abnormal periods with outliers were included. The temporal structure contains multiple motifs along with their temporal distances, with each motif only occurs once in each period. Specifically, we differentiate between two kinds of motifs in the proposed method: **period motifs** and **action motifs**, and leverage multiple period motifs and action motifs to achieve robust recognition. In this chapter, the term period motif refers to a unique data segment that

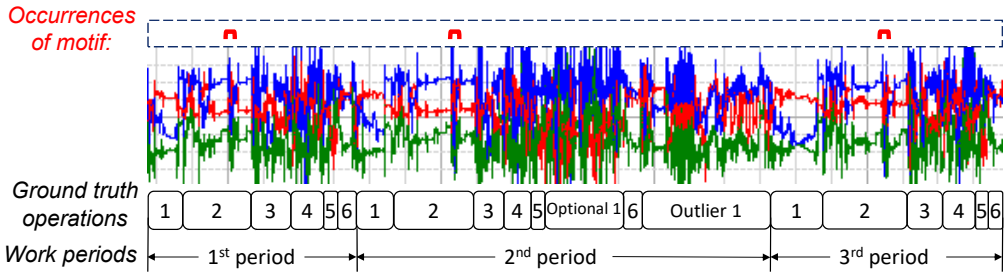


Figure 3.1: Example acceleration data collected from a worker. The red, green, and blue lines show the x-, y-, and z-axis data, respectively. The rectangles below show the ground truth labels of the operations. The duration of each operation period is about 59 s. Each work period consists of six operations, which are defined in the process instructions. The red brackets show the occurrences of an example motif. "Optional 1" shows an optional operation.

occurs only once in each work period, e.g., data corresponding to a hammering action that is performed only once in each work period. The term action motif refers to a data segment that occurs several times in each work period, e.g., data corresponding to a common repetitive action such as screwing. The proposed method concurrently tracks multiple period motifs, detecting the candidate locations for each motif by monitoring the sensor data using a computationally efficient method, and then verifies each possible occurrence of a motif by using not only the standard lead time of a work period, but also the temporal relationship between that occurrence of the current period motif and the neighboring occurrences of other period motifs. Finally, we incorporate the temporal structure of multiple period motifs into a particle filter in order to achieve a robust motif tracking algorithm.

By leveraging multiple period motifs per work period, our method is able to capture the rough structure of each work period in a robust way. For example, assume that we have a work sequence that consists of eight operations, where we have extracted period motifs from the first, second and sixth operations as shown in Figure 3.2. If an outlier activity were to occur between the first and second operations, we would roughly know that the outlier segment exists between the first and second operations based on the temporal relationship (temporal distance) between the period motifs of the first and second operations, enabling us to roughly estimate the ending time of the first operation

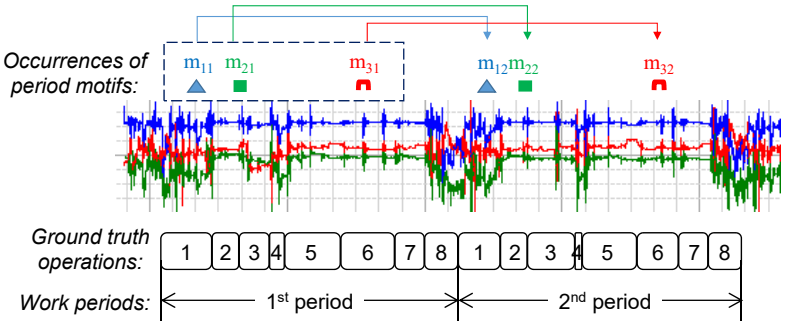


Figure 3.2: Example acceleration data collected from a worker. The duration of each operation period is about 130 s. Each work period consists of eight operations, which are defined in the process instructions. Three period motifs are extracted in this example. m_{xy} shows the y-th occurrence of the x-th motif.

and starting time of the second operation. In this way, leveraging the temporal structure of multiple motifs enables us to improve the robustness of the recognition process. The temporal structure of the period motifs can also be used to robustly track multiple period motifs. In the example shown in Figure 3.2, assume that it is difficult to detect the occurrence of motif m_{22} based on the sensor data due to noise in the sensor data. In this case, we can roughly locate motif m_{22} by using the temporal relationships that were observed in previous work periods between m_2 and m_1 and between m_2 and m_3.

Along with the period motifs described above, in this chapter, we also incorporate information derived from action motifs. Action motifs occur much more frequently within any given work period than period motifs, because action motifs correspond to repetitive actions such as screwing. Therefore, good action motifs can be used to understand the fine-scale structure of a work period (i.e., the start time of each operation). When estimating the start time for each operation within a work period, we can then combine information about the locations of occurrences of action motifs with information about the locations of period motifs, information about trend changes in the sensor data, and information from the process instructions. Specifically, we prepare a secondary particle filter that is responsible for tracking the start time of each operation by leveraging the above information as follows.

- The transition points between operations in a work period may contain noticeable

trend changes in the sensor data. We leverage a nonparametric Bayesian hidden Markov model (HMM), which was mentioned in Chapter 2, to detect the trend changes as the candidates of operation start times.

- Because several period and action motifs are observed in each work period, the temporal relationships between the start time of an operation and the occurrences of period and action motifs can be used to estimate the start time of the operation. For example, in Figure 3.2, the start times of the third operation can be estimated as being just after occurrences of the second period motif (i.e., m_{21} and m_{22}). However, in the case of action motifs, because they can occur many times within each work period, it is difficult to precisely track each of its occurrences. Therefore, we instead feed simple statistical features related to occurrences of each action motif (e.g., the number of estimated occurrences between the estimated start and end times of an operation) into the secondary particle filter.

The research contributions in this chapter are summarized as follows.

- We propose a robust unsupervised factory activity recognition method that makes use of the temporal structure of two types of motifs: period motifs and action motifs.

- Our method showed significant improvement in activity recognition accuracy over previous unsupervised methods when using real-world sensor data collected from factory workers, even when outliers were included in the sensor data.

- As a side effect of our method for tracking multiple motifs, we can estimate the start times of operations with lower computation costs than the previous method that relied on the dynamic time warping (DTW) [64], with our method showing a significant improvement in computation speed when using real-life sensor data collected in a factory.

The proposed method outperformed the similarity-based method when outliers were included in the sensor data. Because the proposed method constructed a motif temporal structure for every period, the consistent temporal distances between the multiple period motifs can be used to detect the occurrences of outlying activities. In addition, we can

estimate the start times of operations more precisely using the temporal relationship between the occurrences of period motifs, which are densely observed, and the operation start times. Besides, statistical features extracted based on the occurrences of action motifs (e.g., information corresponding to the number of occurrences of an action motif) permit us to calculate the similarity between sensor data segments, which is used to compare data segments with those from previous periods, with smaller computation costs than the previous method, which relied on DTW.

By using our novel idea of leveraging two types of motifs during unsupervised factory activity recognition as listed above, we were able to improve recognition accuracy by 36% for work processes containing outliers and reducing computing costs by 86%.

In the rest of this section, we first introduce work related to activity recognition including factory activity recognition. Then, we describe the design of our method for factory work recognition. Finally, in the evaluation section, we test our method using real-world sensor data collected in factories.

3.2 Related Work

The concept of motif was proposed based on the idea that approximately repeated patterns can be informative, which are now widely used in the field of bioinformatics [10, 56, 17]. Similarly, for time-series processing in data mining, it is also possible to detect the repeated patterns to achieve a better segmentation and classification. The first formal definition of motif in time series appeared in 2003 by Chiu et al. [16], who proposed an algorithm to find approximately repeated subsequences (i.e., motifs) with high accuracy and low computation costs. Since then, there have been dozens of papers that use motifs in a variety of applications.

Several activity and gesture recognition studies have extracted motifs in an unsupervised manner. Minnen et al. [48] first found motif seeds using a minimum description length (MDL) criterion and then refined the motif seeds by splitting, merging, and extending the motifs. The MDL criterion permitted them to find motifs that were as long as possible and had high intra-motif similarity. They employed left-to-right HMMs to detect the occurrences of the motifs. Tanaka et al. [70] discovered motifs from multidimensional data based on MDL principle to detect some characteristic actions from

a series of motion data. Arita et al. [5] used motifs in a "real-time human proxy" (i.e telepresence) application. The idea is that motif discovery is used offline to capture typical human motions, for example "pointing with right hand". These complex actions can then be represented and efficiently transmitted as a single symbol, rather than a complex set of real valued motion trajectories. Dragomir et al. [79] proposed an algorithm that allows discovery of motifs whose lengths are different from each other. Berlin et al. [11] recognized leisure activities by employing useful motifs in acceleration signals. They selected motifs that occurred at least two times and had a minimum length. In contrast to these studies, our study finds motifs useful for tracking repetitive works.

3.3 Methodology

3.3.1 Problem Set-Up

In this chapter, outlying activities randomly occur in work periods. The goal of this study is to robustly find a set of timings $D_{T,estimation}$ on a time-series that has a minimal overall error to their corresponding ground truth operation start times $D_{T,groundtruth}$ even outlying activities exist.

Since the occurrences and duration of outlying activities are unknown, it is difficult to track the starting times of operations only using the sensor data similarity and semantic similarity in word descriptions described in Chapter 2. We assume that outlying activities can exist in every operation, and the duration of an outlying activity can be labeled as t_{ol}. When tracking starting times of operations using particle filter, particles corresponding to each operation are generated according to not only the standard duration of operations but also the random occurrences of t_{ol}. After that, many sets of estimated operation start times were generated, with each set being evaluated by several criteria proposed in this chapter. Note that the estimated length of t_{ol} is randomly generated, which is various in different operations. Therefore, this chapter focuses on criteria designed based on the consistent data structure of work periods to detect the outlying activities efficiently. While these criteria do not guarantee that the obtained $D_{T,estimation}$ yields the smallest error to $D_{T,groundtruth}$, we evaluated the criteria by investigating if the proposed criteria achieve a small error using actual data.

In the following sections, we will introduce the proposed method of finding the best

$D_{T,estimation}$.

3.3.2 Preliminaries

Sensor Data Collection

This study assumes that workers are wearing a body-worn accelerometer. In our experiment, each worker wore a smartwatch on his/her right wrist, with the smartwatch's accelerometers collecting three-axis acceleration data at a sampling rate of 60 Hz.

Process Instructions and Work Models

Similar to the last chapter in Section 2.3.2, the process instructions are documents prepared in advance by a line manager for each work process to be performed by a worker. The process instructions specify the flow of operations performed during a work period, including the standard duration and description of every individual operation. While our goal in this chapter is to identify the starting times and duration of operations in an unsupervised manner, by only using the information from the process instructions, it is impossible to locate the actual starting timing and duration of each operation.

Similar to the last chapter in Section 2.3.4, we construct a work model to encapsulate key information from the instructions. The work model is represented by a tree-structure that describes the flow of operations, where each node in the model contains the name and expected duration of its corresponding operation, which is modeled as a log-normal distribution. The edges that directed from one node to another indicate the order of operations. Optional operations are represented in the tree by assigning multiple child nodes to the node immediately preceding the optional operation, with one branch of the split leading to the operation that normally follows and the other branch leading to the optional operation.

Because this work model describes the order of operations and the duration of each operation, it is used to estimate the start time of the next operation in our method. While the work model applied in the last chapter used the semantics of the operations acquired from the descriptions of the operations, whose evaluation result revealed that this information did not contribute much to the method's accuracy and incorporating this information required high computation costs when processing it in conjunction with the

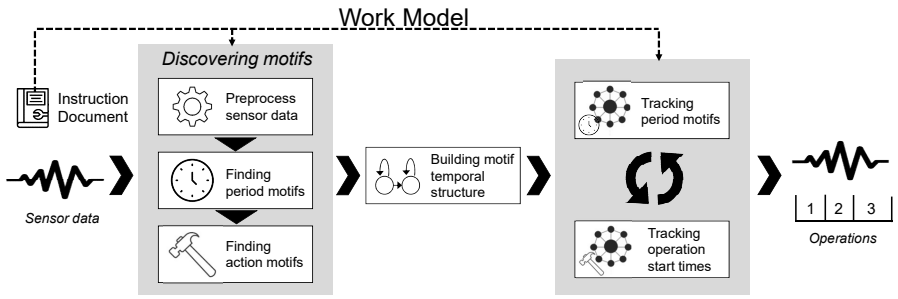


Figure 3.3: Overview of the proposed method

sensor data. Therefore, this chapter does not use that semantic information.

3.3.3 Overview

Figure 3.3 shows an overview of the proposed method. It is mainly composed of four procedures: (i) Discovering motifs, (ii) Building motif temporal structure, (iii) Tracking period motifs, (iv) Tracking operation start times. When discovering motifs, we first preprocess the data and then find period and action motifs within an initial segment of the sensor data (i.e., the first t_{dm} minutes). We then build a model that describes the temporal structure of the occurrences of the period motifs within that initial data segment. Finally, we run particle filters to track the period motifs through all work periods and combine the results from those particle filters with information about the action motifs in order to run a secondary particle filter that tracks the operation start times. Note that these particle filters are based on the work model and that our method does not require any labeled training data. Table 3.1 lists the symbols used in the following subsections.

3.3.4 Discovering Motifs

Preprocessing Sensor Data

In order to reduce computation costs, we first symbolize the input acceleration data based on previous factory activity recognition methods [41, 76]. In brief, we convert each of downsampled numerical acceleration values into a symbol based on the value range associated with each symbol. We then segment the sensor data to identify the

Table 3.1: List of the symbols used in the following descriptions

Parameter	Description
t_{dm}	the duration of initial sensor data used for finding motif candidates
t_{init}	the duration of initial sensor data used for calculating scores in motif selection
l_m	window length of motif, which is the shortest standard duration of operations in the instructions
$\mathbf{S}$	time-series of similarity values for a motif
k_p	the number of period motifs
k_a	the number of action motifs
τ_{pm}	similarity threshold used when detecting a period motif
d_f	a hyper-parameter defining the possible minimum lead time ($0 < d_f < 1$)
t_{ol}	an estimate of the duration of an outlier existing between two period motifs
$\mathbf{C}$	a set of trend changes

likely starting locations for operations using HMMs (HDP-HMM). The segmentation boundaries in the results output by HDP-HMM give us the locations of trend changes in the sensor data. These detected trend changes are then used to estimate the start times of operations in the tracking phase. Although this segmentation method resulted in many false positives for operation start times, the number of false negatives is generally low. Therefore, we can use this segmentation method to generate candidates for operation start times, but they will require further refinement by later processes.

Finding Period Motifs

We start by finding period (and action) motifs within the first t_{dm} minutes of sensor data starting from the beginning of collection. Having already preprocessed the data as described above, we randomly extract candidate period motifs with a length of l_m from the first t_{init} minutes of the sensor data, with $t_{init} < t_{dm}$. For each candidate period motif, we then compute a time series of similarity values. As shown in Figure 3.4 (b, c), we compute the similarity between each candidate period motif and consecutive data segments in the first t_{dm} minutes of sensor data using a sliding window, which results in a new time series $\mathbf{S}$ of similarity values for the first t_{dm} minutes of data. Note that

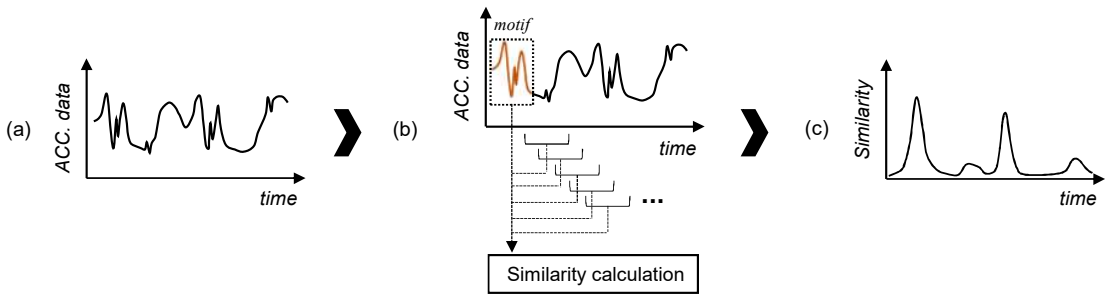
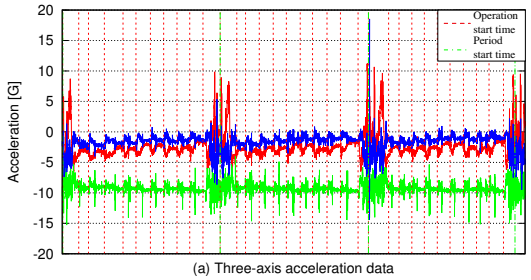


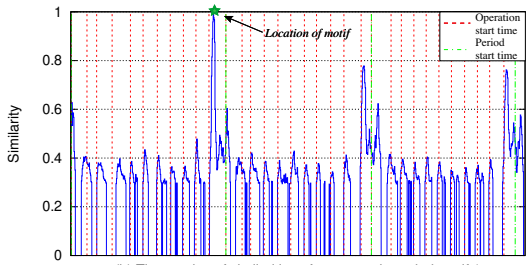
Figure 3.4: Calculating the similarity between segments of time series data. (a) Example original sensor data, (b) calculating the similarity between a motif and consecutive data segments using a sliding window, and (c) resulting output time series of similarity values.

the similarity metric is computed based on the Levenshtein distances between segments of symbolized data. Using the similarity time series $\mathbf{S}$, we are now able to choose good period motifs from amongst the candidates. Figure 3.5 shows two examples of such similarity time series, with (a) showing the raw acceleration data and (b) and (c) each showing the similarity times series for a different candidate motif. Figure 3.5 (b) shows an example of a good candidate for a period motif, with the location of the motif's template indicated by a star. Meanwhile, Figure 3.5 (c) shows an example of a bad candidate for a period motif, with the template again indicated by a star. In the case of Figure 3.5 (b), we can find large peaks in the similarity values at the end of each work period, with a clear differentiation between the values in those peaks and the similarity values in rest of the work periods. This means that the waveform of the candidate motif is dissimilar to unrelated segments and appears consistently throughout the periods. In contrast, in Figure 3.5 (c), the differences between the similarity values at peaks and at other times are relatively small, making it difficult to detect occurrences of the motif. As this example shows, candidates that yield clear peaks should be selected.

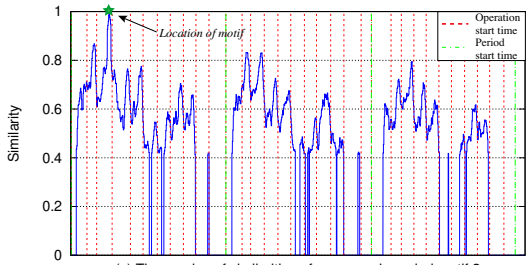
We select the top-k_p best period motifs from amongst the candidates using a combination of the following scores: (1) As shown in Figure 3.5 (b, c), the similarity values at peaks should be much larger than those at other times in the periods. Therefore, we compute the first score based on the difference between the peak similarity value and similarity values at the other time slices. (2) In addition to the above score, we simply



(a) Three-axis acceleration data

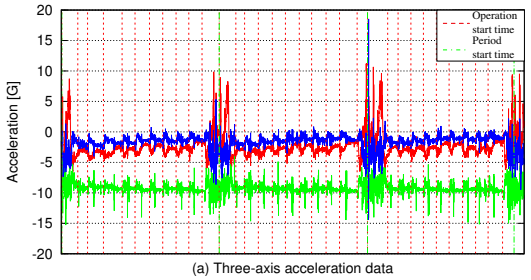


(b) Time-series of similarities of an example period motif 1

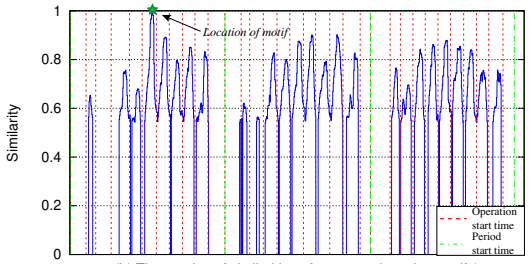


(c) Time-series of similarities of an example period motif 2

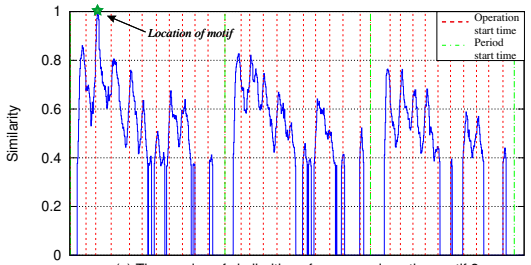
Figure 3.5: Comparing the similarity time series for candidate period motifs. (a) The raw three-axis acceleration data. (b) The similarity time series for a good candidate for a period motif. (c) The similarity time series for a bad candidate for a period motif. The vertical dotted green lines show the start times for work periods, i.e., start time of the first operation. The vertical dotted red lines show the start time of each operation (except the first operation). The locations of the motif templates are indicated with stars.



(a) Three-axis acceleration data



(b) Time-series of similarities of an example action motif 1



(c) Time-series of similarities of an example action motif 2

Figure 3.6: Comparing the similarity time series for candidate action motifs. (a) The raw three-axis acceleration data. (b) The similarity time series for a good candidate for an action motif. (c) The similarity time series for a bad candidate for an action motif. The vertical dotted green lines show the start times for work periods, i.e., start time of the first operation. The vertical dotted red lines show the start time of each operation (except the first operation). The locations of the motif templates are indicated with stars.

use the similarity value of the candidate motif as the second score.

Here we explain the first score used for selecting a candidate period motif at time t.

$$\max(\text{peak}(seg_1)) - \text{mean}(\text{peak}(seg_2)),$$

where seg_2 is a data segment over $(t, t + 0.8\mathcal{M}.min_lead_time]$ within $\mathbf{S}$, seg_1 is a segment over $[t + 0.8\mathcal{M}.min_lead_time, t_{init}]$ within $\mathbf{S}$, $\mathcal{M}.min_lead_time$ is the shortest standard lead time described in the work model $\mathcal{M}$, and peak(seg) detects peaks (local maxima) within seg by simply computing the derivative. Note that we slightly shorten the threshold of $\mathcal{M}.min_lead_time$ by multiplying it with "0.8" in order to ensure that the motif in the next period will not be included in seg_2. We roughly select the value "0.8" based on the average duration of operations in our datasets, which performs well in our data sets in our preliminary investigation. However, the optimal threshold for all data sets is unknown. Because we calculate the average peak values in seg_2, aiming to reduce the effect of the hyper-parameter setting, we are confident that the hyper-parameter will not have a significant influence on finding period motifs.

Here, seg_1 shows a segment that can contain occurrences of the motif and is used to obtain similarity values at occurrences of the motif. In addition, seg_2 shows a segment that does not contain occurrences of the motif and is used to obtain similarity values other than the occurrences of the motif. With this score, we can capture the difference between the similarity value at the highest peak and those at the other time slices.
We also store a parameter $\tau_{pm} = 0.8 \max(\text{peak}(seg_1))$ for each candidate motif, which gives us a threshold to use when determining if similarity values for sensor data should be considered to be an occurrence of the motif during the procedure used when tracking period motifs.

We compute the similarity value of the candidate motif as the second score.

$$\max(\text{peak}(seg_1)).$$

Finding Action Motifs

Just as with period motifs, we start our process of selecting good action motifs by first extracting candidate motifs and then computing a similarity time series $\mathbf{S}$ for each candidate using the first t_{dm} minutes of sensor data. Figure 3.6 shows two examples of such similarity time series, with (a) showing the raw acceleration data and (b) and (c) each

showing the similarity times series for a different candidate motif, with (b) showing an example of a good candidate and (c) showing an example of a bad candidate. In Figure 3.6 (b), the motif corresponds to a screwing action and the presence of clear peaks that repeat within each work period make it a good candidate. The thin peaks shown in this example are indicative of a good candidate for an action motif, as they make it easier to differentiate the peaks from background noise. In contrast, in Figure 3.6 (c), there is not a clear differentiation between similarity values at the peaks versus at other times in the work period, indicating that the candidate motif is not unique and is therefore hard to detect. Furthermore, the time series in Figure 3.6 (c) varies greatly from one work period to the next, indicating that the motif candidate yields an unstable similarity time series. In contrast, the time series in Figure 3.6 (b) is much more stable across work periods.

In order to find good action motifs such as the example shown in Figure 3.6 (b), we select the top-k_a best action motifs from the candidates as follows. As shown in Figure 3.6 (b), the similarity values at occurrences of an action motif should be high and easily differentiated from the similarity values from other unrelated data segments. Therefore, we calculate the score as the difference between the mean similarity values of peaks and that of the other time slices. We calculate the score based on the average prominence of peaks in a segment of similarity time series that is expected to contain the initial work period:

$$\text{mean}(\text{prominence}(\text{peak}(seg_3))),$$

where seg_3 shows a segment that contains the initial work period, the "prominence(peaks)" measures how much a peak stands out due to its intrinsic height and its location relative to other peaks, which is calculated by [20].

3.3.5 Building Motif Temporal Structure

Using the processes described in the previous subsection, we are able to select good candidates for period motifs (and action motifs). Here we build a model of the motifs' temporal structure ($\mathcal{T}$) that captures the temporal distances between pairs of period motifs. This model represents the temporal structure using lists of observed temporal distances between the nearest occurrences of each of the motifs. Each list is initialized based on the relationship between the original data segments for the corresponding mo-

tifs in the first t_{init} minutes of sensor data. Although each list contains only a single value upon initialization, the lists will be updated during the following procedures, enabling us to accurately model the temporal distances between period motifs. Given a temporal structure model $\mathcal{T}$, $\mathcal{T}(m_i, m_j)$ outputs a single value calculated as the average value for the list of temporal distances between the pair of motifs m_i and m_j, which is used in the tracking procedure.

3.3.6 Tracking Period Motifs

We track period motifs using particle filters, with a separate particle filter prepared for each motif. Note that of these particle filters, the one with the highest score is hereinafter referred to as the *base particle filter*, with the period motif tracked by this particle filter referred to as the *base period motif*. Because the base particle filter's results are more reliable than those from the other lower-scoring particle filters, its results can be used to correct the results for those other particle filters. We employ particle filters to track period motifs (and operation start times) because the robustness and effectiveness of particle filters in unsupervised factory activity recognition were confirmed in previous studies [41, 76]. In this chapter, we introduce multiple period motifs into the particle-filter-based activity recognition in order to achieve more robust tracking.

Particle filtering consists of three procedures: sampling, weighting, and resampling. During the sampling procedure, we generate new particles that represent predicted locations for the occurrence of a period motif in the $(i+1)$-th period based on its estimated occurrence in the i-th period, as shown in Figure 3.7 (a). In the weighting procedure, we calculate a score for each of those new particles based on the motif's similarity time series and the results from the other motifs' particle filters. In the resampling procedure, we probabilistically discard particles with low scores. We explain these procedures in detail below.

Initialization

When tracking a period motif, we start by creating a tree structure, which we will refer to as a history tree, that is used to track all the particles generated for that motif. We generate the first particle for that motif at the location of its first occurrence in the sensor data and place that particle in the tree as the tree's root node.

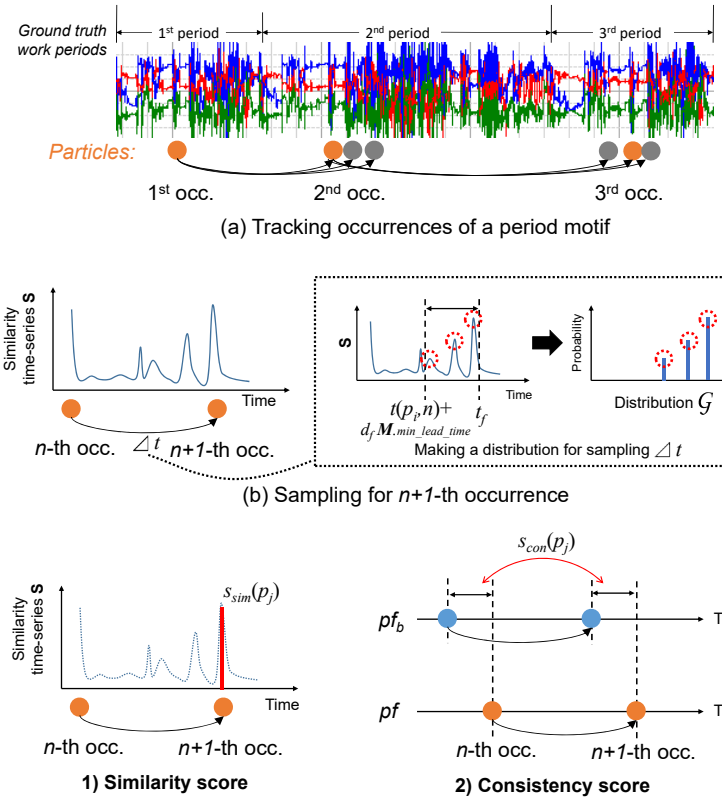


Figure 3.7: Overview of the procedures used when tracking period motifs based on particle filters. (a) We start with an initial particle at the first occurrence of the period motif. We then sample several particles from this initial particle (three in this example), which are the initial estimates for the second occurrence of the motif. These sampled particles are then weighted and resampled (one in this example), leaving us with our most likely estimate(s) for this occurrence. The most likely estimates are then sampled to generate the estimates for the third occurrence of the motif. (b) During the sampling phase, new particles are generated at a distance of Δt from the previous particle, with Δt sampled from a distribution $\mathcal{G}$ that is based on the similarity time series $\mathbf{S}$ with the likelihood of sampling a particle at t proportional to the similarity value at t. (c) The $n+1$-th occurrence of a particle is weighted using a combination of a similarity score and a consistency score. The similarity score is the similarity value at the $n+1$-th particle. The consistency score measures the consistency of the temporal distance between the $n+1$-th occurrence of the current motif pf, which corresponds to a raw sensor data segment, and the $n+1$-th occurrence of the base motif pf_b when compared to their temporal distances in past work periods.

Sampling

We randomly sample subsequent occurrences for each particle based on $\mathbf{S}$ and the standard lead times from the work model $\mathcal{M}$, which is illustrated in Figure 3.7 (b). Assuming that $t(p_i, n)$ is the timestamp for a particle p_i generated during the n-th iteration of sampling, then the timestamp for some new particle p_j from among the m particles generated from p_i during the $n+1$-th iteration of sampling is determined using the following equation.

$$t(p_j, n+1) = t(p_i, n) + \Delta t,$$

where Δt is the estimated interval between p_i and p_j that is randomly sampled from a distribution $\mathcal{G}$ generated from $\mathbf{S}$ and standard lead times. $\mathcal{G}$ is a discrete distribution that describes possible locations of the $n+1$-th occurrence of the motif, which is illustrated in the right portion of Figure 3.7 (b) and formulated as follows.

$$\mathcal{G} = \textsc{Disc}(\text{peak}(\mathbf{S}_{(t_{f0}, t_f)})),$$

$$t_{f0} = t(p_i, n) + d_f \cdot \mathcal{M}.min_lead_time,$$

where t_{f0} is a time between the occurrence of particle p_i and p_j, t_f is a time when $\mathbf{S}$ first exceeds τ_{pm} after t_{f0} plus a margin, d_f is a hyperparameter that defines the possible minimum lead time ($0 < d_f < 1$), $\mathbf{S}_{(t_{f0}, t_f)}$ is a segment between t_{f0} and t_f within $\mathbf{S}$, and $\textsc{Disc}(peaks)$ calculates a discrete distribution where the probability of sampling at a peak is proportional to the value at that peak and the probabilities at time slices other than the peaks are zero. Note that, when an outlier segment occurs after $t(p_i, n)$ and a sub-segment within the outlier segment exhibits a high similarity value with the motif of interest, the value of t_f may be erroneously set as a time before the next occurrence of the motif. To help cope with this, the particle filters conditionally apply a correction to t_f based on the results for t_f from the other particle filters. Briefly speaking, when $t_f - t(p_i, n)$ (i.e., temporal distance between t_f and the timestamp of p_i) for some particle filter is much smaller than $t_f - t(p_i, n)$ for other particle filters, we increase t_f for this particle filter based on $t_f - t(p_i, n)$ from the base particle filter.

Each of the m particles generated from p_i are then our estimates for the possible $n+1$-th occurrences of the motif, with each new particle p_j being stored as a child node of p_i in the history tree.

Weighting

After having sampled new particles based on the discrete distribution, we then calculate a score for each new particle p_j using a combination of the following scores, which are illustrated in Figure 3.7 (c):

- **Similarity score:** The similarity value at $t(p_j, n+1)$ should be high if the particle is a good candidate. Therefore, we simply calculate the similarity value of the particle at $t(p_j, n+1)$ as follows.

$$s_{sim}(p_j) = \mathbf{S}_{t(p_j,n+1)}$$

- **Consistency score:** We calculate this score by referring to the base particle filter and the temporal structure model $\mathcal{T}$. Because the temporal relationship between the occurrences of period motifs should be consistent throughout work periods, we calculate this score using the following equation:

$$s_{con}(p_j) = \frac{1}{|\mathcal{T}(pf_b, pf) - (o(pf_b, n+1) - t(p_j, n+1))|},$$

  where pf is a motif that tracks from a particle filter, the pf_b is a base motif that tracks from a base particle filter which is more reliable, and $o(pf_b, n+1)$ is the posterior estimate of the $n+1$-th occurrence of the base motif (refer to Section 3.3.6). Therefore, the above function calculates the consistency in the temporal distance between the occurrences of the motifs $(o(pf_b, n+1) - t(p_j, n+1))$ by referring to the history of the time difference in the temporal structure model $(\mathcal{T}(pf_b, pf))$[1].

Resampling

Here we employ roulette wheel selection to probabilistically resample the generated particles. In roulette wheel selection, each particle is randomly selected with a probability proportional to the score assigned to that particle during the weighting phase. This method enables us to select the high-scoring particles with higher probability and discard the remaining particles. The discarded particles are removed from the history tree.

[1]To deal with an optional operation, we have $\mathcal{T}(m_i, m_j)$ for each possible pattern of the process (i.e., the number of patterns is identical to the number of leaf nodes in the operation flow model).

The posterior estimate of the occurrence time for the motif of interest $o(pf.motif, n+1)$ is the weighted average of the resampled particles, with their scores used as the weights.

Finally, we update the temporal structure model for each pair of motifs by appending the temporal distance between their most recent occurrences (posterior estimates) to the list.

3.3.7 Tracking Operation Start Times

The particle filters for period motifs output timestamps for the occurrences of the motifs. In addition, we have timestamps for trend changes $\mathbf{C}$ (candidates of operation start times) that are output by HDP-HMM. Furthermore, we have information about the standard duration for each operation which is described in the work model $\mathcal{M}$. We now use this information as input to a particle filter that produces our estimates for the most likely start times for all operations in a work period. We start by initializing this particle filter by generating initial particles as candidates for the start time of the work period (i.e., the start time of the first operation). We then track each of these particles to identify the start times of the following operations in the work period by iterating through the particle filter's three phases of sampling, weighting, and resampling as shown in Figure 3.8 (a). Furthermore, because our method tracks the start times of the operations in the order that they are described in the work model, our method can determine the class label of a segment between two adjacent start times in the unsupervised manner. We explain each of these procedures in detail below.

Initialization

As shown in Figure 3.2, the start time of the first work period exists somewhere before the first occurrence of the first motif (m_{11}). Therefore, we can generate initial particles at each timestamp from $\mathbf{C}$ that fall between the starting time of the sensor data and the first occurrence of the first motif, which become our candidates for the start time of the first operation in the first work period. We track the estimated operation start times using a tree structure, with a tree created for each initial particle with that initial particle set as the tree's root node.

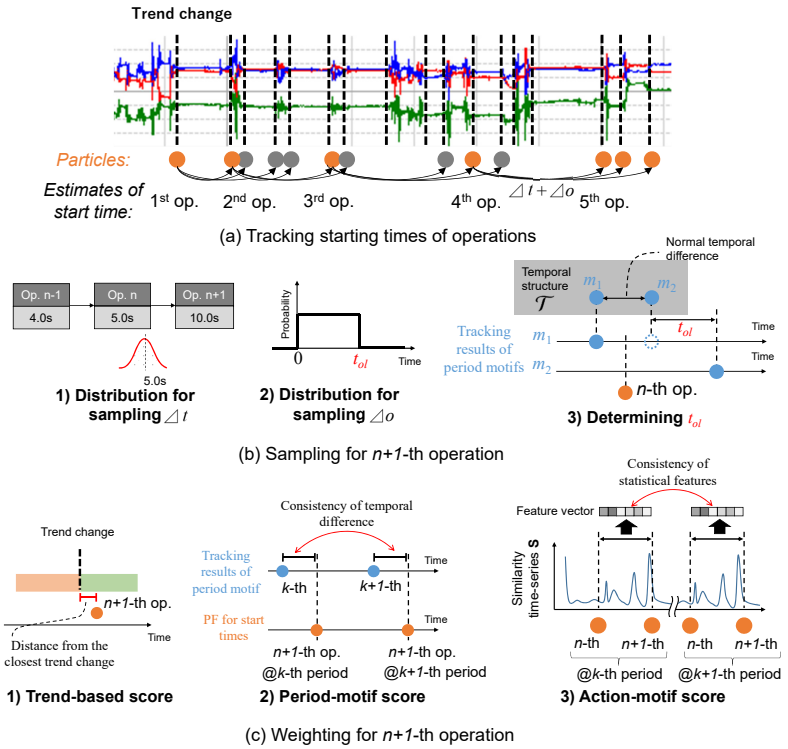


Figure 3.8: Overview of the procedures used when tracking operation start times based on particle filters. (a) Starting from an initial particle representing our estimated start time for the first operation, we sample several particles (three in this example), which are the initial estimates for the start time of the second operation. These sampled particles are then weighted and resampled, leaving us with our most likely estimate(s) for the start time of the second operation (one in this example). (b) During the sampling phase, the temporal distance between the particles for the n-th and $n+1$-th operations is $\Delta t + \Delta o$. Δt is sampled from a distribution of the duration of the n-th operation described in $\mathcal{M}$. Δo is sampled from a uniform distribution over $[0, t_{ol}]$. t_{ol} is an estimate of the duration of outliers, which is calculated as the extended time of the temporal distance between period motifs m_1 and m_2 compared to that in the temporal structure model $\mathcal{T}$. m_1 and m_2 are motifs just before/after the start time of the n-th operation. (c) Weighting phase of the particle filter for a particle corresponding to the start time of the $n+1$-th operation. The trend-based score for the particle is calculated based on the temporal distance from the closet trend change. The period-motif score for the particle is calculated based on the consistency of the temporal distances from adjacent period motifs. The action-motif score for the particle is calculated based on the consistency of statistical features extracted from the similarity time series corresponding to the estimated n-th operation.

Sampling

As shown in Figure 3.8 (a,b), with each iteration we sample new particles from each existing particle that represent estimates for the start time for the following operation. Assuming that $t(p_i, n, k)$ is the timestamp for an existing particle p_i that represents an estimate of the start time of the n-th operation in the k-th work period, then the timestamp $t(p_j, n+1, k)$ for a new particle p_j for the $n+1$-th operation is sampled from p_i as follows:

$$t(p_j, n+1, k) = t(p_i, n, k) + \Delta t + \Delta o,$$

where Δt is randomly sampled based on the duration-time distribution in the work model tree that corresponds to the current node (i.e., the n-th operation's node). Δo represents estimates for the duration of an outlier that possibly exists between the n-th and $n+1$-th operations and is randomly sampled from a uniform distribution over $[0, t_{ol}]$, where t_{ol} is an estimate of the duration of any outlier segments existing between the occurrence of period motif m_1 just before $t(p_i, n, k)$ and the occurrence of another period motif m_2 just after $t(p_i, n, k) + \Delta t$. When the temporal distance between the two occurrences, i.e., $o(m_2, k) - o(m_1, k)$, is much larger than $\mathcal{T}(m_2, m_1)$ as shown in the right part of Figure 3.8 (b), we can assume that there are some outlier segments between the two occurrences. Therefore, t_{ol} is calculated as follows.

$$t_{ol} = \begin{cases} (o(m_2, k) - o(m_1, k)) - \mathcal{T}(m_2, m_1) & ((o(m_2, k) - o(m_1, k)) - \mathcal{T}(m_2, m_1) > th_{ol}) \\ 0 & (\text{otherwise}) \end{cases}$$

Each new particle p_j is then stored as a child node of p_i in the history tree for operation start times.

Weighting

As shown in Figure 3.8 (c), we compute a weight for each particle generated in the sampling phase using a combination of three scores: a trend-based score, a period-motif score, and an action-motif score. These scores are calculated as follows:

- **Trend-based score:** This score enables us to detect operation start times based on the trend changes discovered by HDP-HMM. This score is computed based on

the temporal distance between the timestamp of a particle and the timestamp of the closest trend change in $\mathbf{C}$ as follows:

$$\textsc{CumulativeDist}(p_j, n, k, \mathbf{C}) = \sum_{m=2}^{n+1} \textsc{Dist}(t(p_j, m, k), \mathbf{C})),$$

where $t(p_j, m, k)$ is the timestamp for the ancestor particle of p_j corresponding to the m-th operation in the k-th work period (when m is equal to $n+1$, it is the timestamp for p_j itself) and

$$\textsc{Dist}(t, \mathbf{C}) = \min_{c \in \mathbf{C}} |t - c|.$$

The inverse of this distance is then used as the score.

- **Period-motif score:** The particle filters for period motifs output posterior estimates for the occurrences of the period motifs. This score evaluates the quality of a series of estimated operation start times based on the consistency of period-motif locations in the estimated work periods, based on our expectation that the temporal distances between operation start times and occurrences of period motifs should be consistent. We start by computing two values for the start time of the particle of interest (i.e., $t(p_j, n, k)$) that measure its distances from the occurrence of the motif just before $t(p_j, n, k)$ and the occurrence of the motif just after $t(p_j, n, k)$. We then use these distances to compare the k-th and $k-1$-th occurrences of a motif by computing the squared difference between each value for k-th and $k-1$-th occurrences. The inverse of the sum of the squared differences is then used as the score.

- **Action-motif score:** A similarity time series is given for each action motif, which represents the occurrences of that action motif. This score then evaluates the quality of an estimated series of operation start times based on the consistency of the waveforms for the action motifs' similarity time series between $t(p_i, n, k)$ and $t(p_j, n+1, k)$, i.e., a segment corresponding to the n-th operation, because waveforms for corresponding segments in different work periods should be similar to each other when the start and end times of the operations are properly tracked. We extract statistical features from the segments for k and $k-1$ to form two feature vectors and then compute the Euclidean distance between the two vectors.

The following statistical features are extracted from each segment: (i) standard deviation, (ii) skewness, (iii) kurtosis, (iv) #peaks divided by the duration of the segment, and (v) mean. The inverse of the distance is then used as the score.

Resampling

First, we probabilistically resample the particles using roulette wheel selection with each particle resampled with a probability proportional to its weight that was assigned above, with any discarded particles removed from the history trees. We next need to determine which operation to use when estimating the following operation start time, since the value for Δt will be set based on that operation's duration-time distribution in the work model. In general, we can simply use the following operation in the work model, unless there is a branch in the model, in which case we randomly select the branch to follow. Each of the remaining particles is then used to generate new particles during the sampling phase of the next iteration.

Note that, as shown in Figure 3.2, the end time of the first work period exists somewhere between the first occurrence of the last motif (m_{31}) and the second occurrence of the first motif (m_{12}). Therefore, we delete particles that do not fall between the n-th occurrence of the last period motif and the $n+1$-th occurrence of the first period motif at the end of each period.

We iterate through these processes of sampling, weighting, and resampling until reaching the end of each period, with the most likely overall sequence of start times finally selected based on the particle that has the highest weight at the end of the period.

Post-processing

During post-processing, we examine each operation that was determined to contain outliers in order to identify which segments in that operation correspond to outlier activity and which correspond to non-outlier activity, i.e., the actual operation. We do this by analyzing the operation using a sliding window with length set equal to the standard duration of the operation, extracting features from the sliding windows using the features described in the section on action motifs. For each window, we then compute the average Euclidean distance between that window's feature vector and feature vectors extracted from the current operation when it occurred without outliers in prior work

periods. The time window with the smallest averaged distance then corresponds to the non-outlier segment, and all remaining segments are regarded as being outlier segments.

3.4 Evaluation

3.4.1 Dataset

We evaluated the proposed method using six datasets collected from workers in real factories. Table 3.2 shows an overview of all the datasets. Note that since each work process is performed by a different worker, the total number of workers is six. The number of standard operations included in a work process is shown in the "# operations" row, with the number of optional operations shown in parentheses, e.g., work process B consists of 6 operations plus 1 optional operation. The standard duration for one work period is shown in the "standard duration" row, with the standard duration when optional operations are present shown in parentheses, e.g., the standard duration of work process B is normally 59 seconds and is 87 seconds when the optional operation is performed. The total number of work periods included in the observed sensor data for each work process is shown in the "periods" row. The total duration of the observed sensor data for each work process is shown in the "data duration" row. Note that the datasets used in this evaluation were collected in the same factories as datasets used in [76]. However, [76] did not use datasets containing outliers because that did not assume the presence of outlier data segments. In contrast, this study uses these datasets containing outliers, which are usually observed in actual factories. The total duration of the outlier activities in the observed sensor data for each work process is shown in the "outlier duration" row. The datasets were collected in two different factories (factories A and B), with the "factory" row identifying the factory in which each work process was conducted. An overview of each work process is shown in the "work overview" row. Work process A involves testing the functions of a circuit board. To test each board, the worker places the board on a dedicated table, operates several dials, and pushes buttons. Work process B involves checking and packing items. The worker checks the appearance of an item, attaches a label to the item, scans a bar-code on the label, bags the item, and packs the bag in a box. When the box is full, the worker arranges a new box. Work processes C and D involve checking products. The worker places a product on a dedicated table,

Table 3.2: Overview of our dataset

	work A	B	C	D	E	F
# operations	8	6 (1)	7	7	11	11
standard duration [s]	130	59 (87)	53	50	55	50
periods	10	9	14	13	12	11
data duration [s]	1440	614	778	632	676	674
outlier duration [s]	0	76	51	14	0	79
factory	A	A	A	B	B	B
work overview	test and record circuit board information	bag and box circuit boards	check final product and record results	check final product and record results	install screws on circuit board	install screws on circuit board



Figure 3.9: Image from video captured in the factory

scans a label on the product, measures the size of the product, checks the appearance, and boxes the product. Work processes E and F involve installing screws on a circuit board. The worker places parts on a board, installs eight screws to fasten the parts, and pushes several buttons to finalize the process.

The sensor data used in this study was captured at approximately 60 Hz using a Sony SmartWatch3 SWR50 attached to each worker's right wrist. The ground truth data needed for evaluation was recorded using video cameras, with Figure 3.9 showing an example of the images captured. The data was processed offline in our laboratory after data collection.

3.4.2 Evaluation Methodology

Our method outputs the start times for all operations in each work period, allowing us to classify each acceleration data point as belonging to an operation class. Therefore, we treat our results as classification results estimated per acceleration data point, allowing us to calculate the macro-averaged F-measure for our results. Because our method can also label outlier segments, an "outlier" class is included when calculating the F-measure.

To evaluate the effectiveness of the proposed method, we prepared the following methods.

- **Proposed**: This is the proposed method.
- **Only-Base**: The proposed method only using a single period motif (base period motif).
- **W/o-Trend**: The proposed method without the use of the trend-based scores in the particle filters used for tracking start times.
- **W/o-Period**: The proposed method without the use of the period-motif scores in the particle filters used for tracking start times.
- **W/o-Action**: The proposed method without the use of the action-motif scores in the particle filters used for tracking start times.
- **Xia2019**: This is the proposed method from the last chapter. This method relies on a single motif and similarity calculation based on DTW. For more details, see the previous chapter and [76].

3.4.3 Results

Recognition Accuracy

Figure 3.10 shows the classification accuracy for each method. Overall, the "Proposed" method achieved the highest average F-measure across all six work processes (0.81).

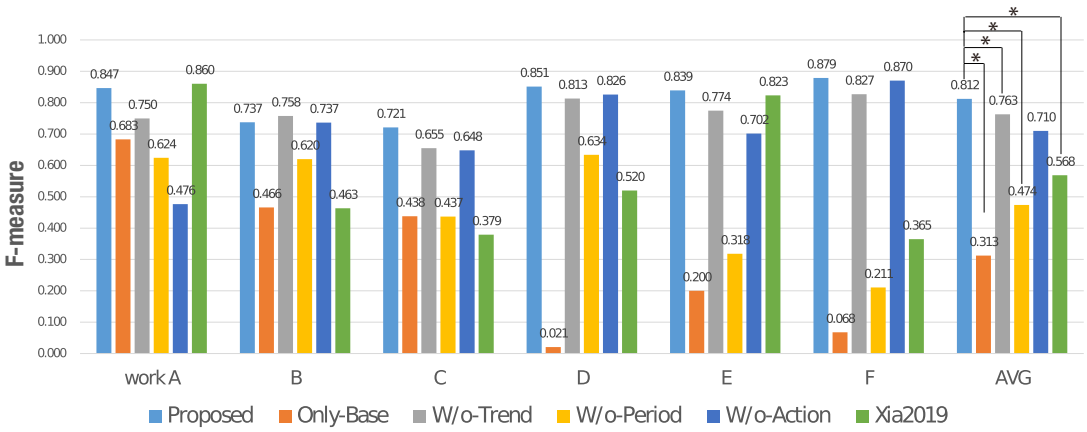


Figure 3.10: F-measures of the methods. Paired t-tests were used to confirm any significant differences between the results for each method. $*p < 0.05$.

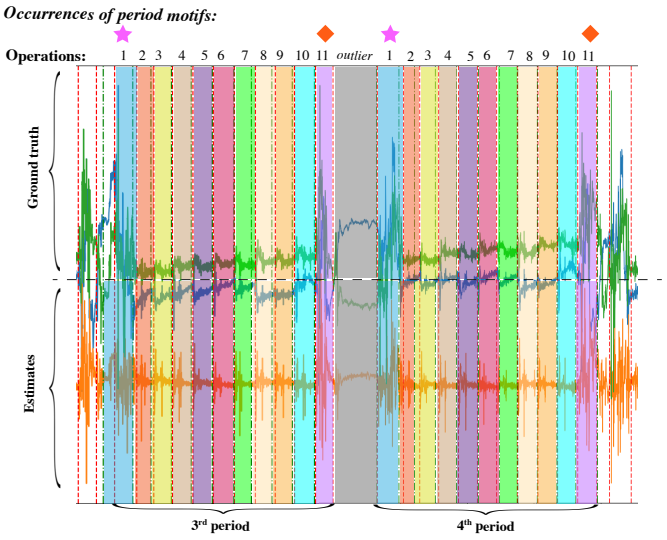


Figure 3.11: Example output from "Proposed" for the third and fourth periods of work process F

Even when only considering the work processes that contain outliers, the average F-measure for "Proposed" was still 0.80. These results confirm the robustness of the proposed method when used on real-world factory work sensor data that contains outliers. In addition, the overall trends of the six methods for the precision and recall are simi-

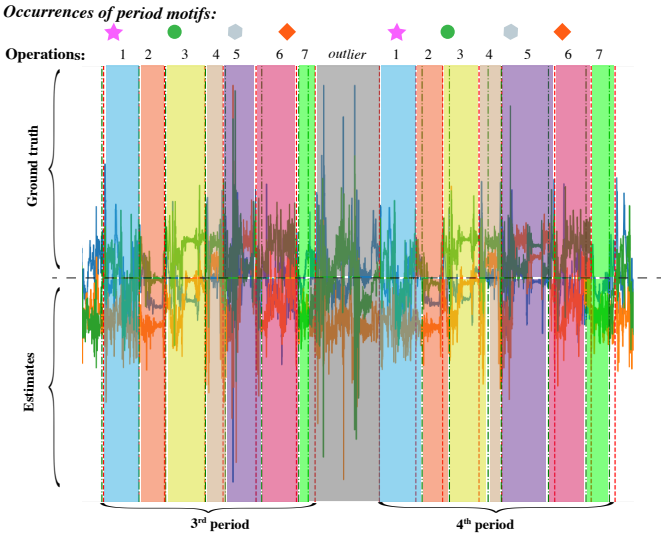


Figure 3.12: Example output from "Proposed" for the third and fourth periods of work process D

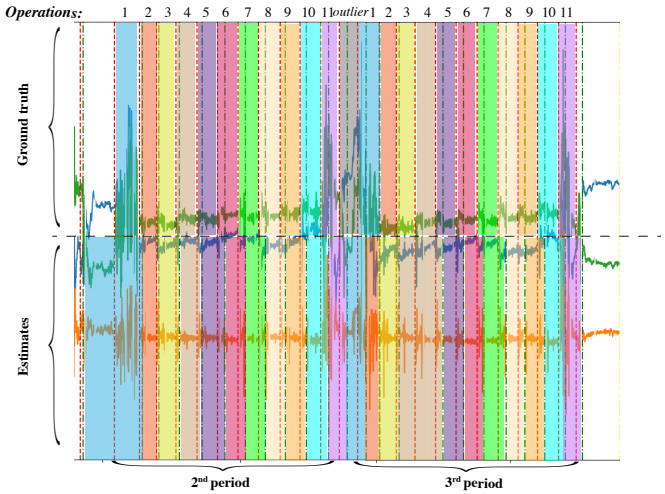


Figure 3.13: Example output from "Xia2019" for the third and fourth periods of work process F

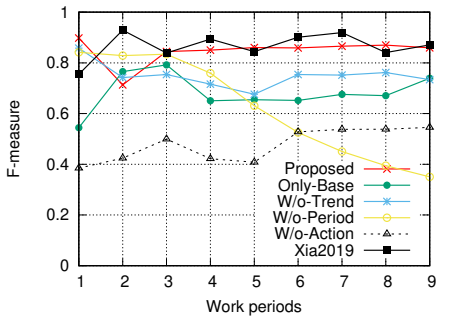
lar to the trend of F-measure. Because the recognition errors are mainly caused by the temporal shifts of estimates from the ground truth, the precision, recall, and F-measure of a method have similar values.

Figures 3.11 and 3.12 show examples of output from the "Proposed" method. The individual operations are indicated by the colored bars, with the upper half of the bars indicating the ground truth segmentation and the lower half of the bars indicating the estimated segmentation output by "Proposed". The numbers in the upper margin correspond to the operation number for each ground-truth segment. The tracking results for selected period motifs are also shown in the upper part of the figure. As shown in these figures, "Proposed" was able to successfully identify the occurrences of the outlier activities even though information about outliers was not given in advance. Figure 3.11 also provides a good example of the importance of using multiple period motifs. In this case, we can see how that the period motifs located in the 1st and 11th operations effectively surrounded the outlier activity allowing "Proposed" to precisely locate its start and end times.

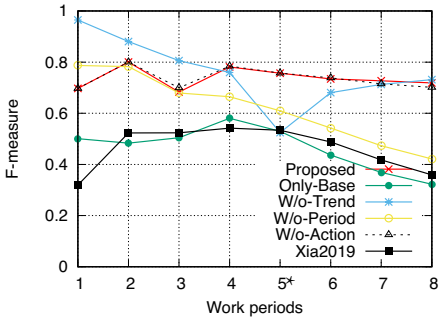
"Proposed" did not achieve the highest F-measure for every work process, with the F-measure of "Xia2019" higher than that of "Proposed" for a work process without outliers (work process A), despite performing significantly worse than "Proposed" overall. Because "Xia2019" employs DTW to exactly match waveforms of an operation in different periods, "Xia2019" achieved the high F-measures when outlier activities did not occur (work processes A and E). However, the use of DTW in "Xia2019" significantly increases its computation costs when compared to "Proposed," which is discussed in further detail below. Also of note are the poor results from the "Xia2019" method in the work processes that contained outliers, again indicating the need for factory activity recognition methods to include some mechanism for handling outliers. For example, Figure 3.13 shows the output from "Xia2019" when an outlier segment was present. Because "Xia2019" could not detect the outlier segment, the estimated start times for the operations that occurred after the outlier segment were shifted forward.

Transitions in Recognition Accuracy

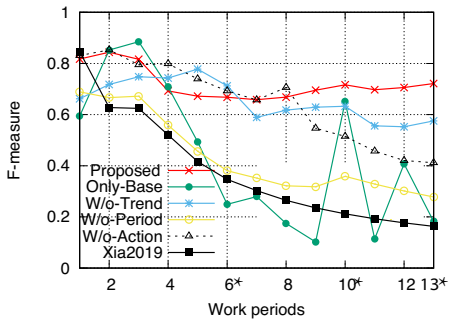
Figure 3.14 breaks down the average F-measures from Figure 3.10 into the individual F-measures for each work period. As shown in these results, the F-measures dropped during or after outlier activities in many cases, reducing the overall F-measures. However, in some cases "Proposed" could maintain a high F-measure even when outlier activities occurred (e.g., the 4th work period in work process D). As shown in Figure



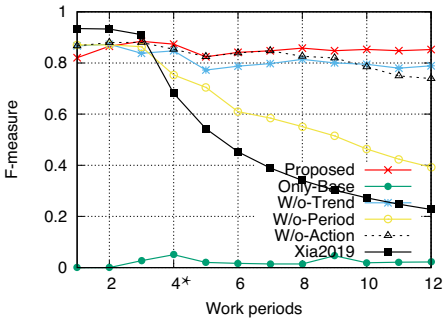
Results from work process A



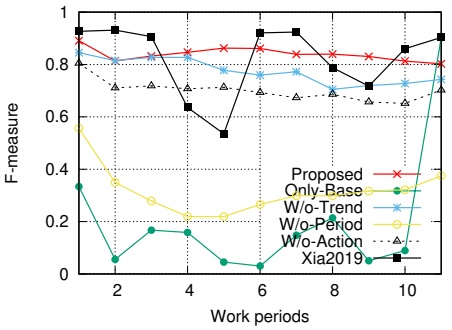
Results from work process B



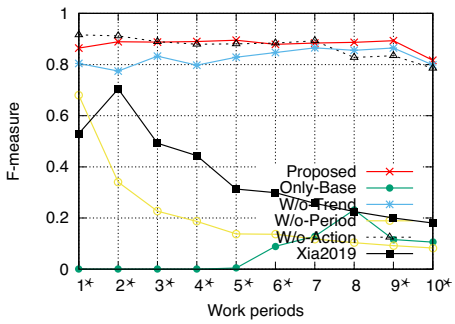
Results from work process C



Results from work process D



Results from work process E



Results from work process F

Figure 3.14: Transitions in F-measures of each method. The (*) in the horizontal axis indicates periods with outliers.

3.14, the F-measures of "Xia2019" dropped in the 4th period of work process C, which does not contain outliers. This is because the duration of one particular operation in that period was much longer than usual. While the durations of many operations in work process C were unstable, deteriorating the F-measures for all the methods, "Proposed"

was able to maintain reasonable F-measures. Overall, "Proposed" maintained high F-measures even in the later work periods, as shown in Figure 3.14. This can be because "Proposed" robustly tracks the start times of operations by leveraging multiple period motifs. In contrast, the F-measures of "Xia2019" gradually decreased in many cases.

Effect of Multiple Period Motifs

Figure 3.10 includes the results for the Only-Base method, which employs only a base period motif. As can be seen in these results, the F-measures of Only-Base are poor for many work processes, indicating that tracking multiple period motifs was an important aspect of "Proposed". While the base particle filter is reliable, it is difficult to recognize the fine-grained structure of the work (i.e., start times of operations) using only a single period motif.

Effect of Trend-based Score

As can be seen in the "W/o-Trend" results in Figure 3.10, the F-measure of "W/o-Trend" is higher for many work processes than those of "W/o-Period" and "W/o-Action". This indicates that the contribution of the trend-based score is lower than the contributions of the period-motif and action-motif scores. Note that the process of detecting trend changes is the most time-consuming part of the "Proposed" method, described in further detail below, indicating that eliminating this score is one possible way to further improve computation costs.

Effect of Period-motif Score

Figure 3.10 also shows the results for "W/o-Period," which does not employ the period-motif score. The F-measures of "W/o-Period" are poor for many work processes, indicating that the temporal relationship between the occurrences of period motifs and operation start times was an important aspect of "Proposed". Because "Proposed" tracked about 5 to 10 period motifs in comparison to the 6 to 11 operations per process, the density of these motifs allowed it to precisely detect the start times of operations. Figure 3.15 shows an example result for "W/o-Period" for work process F. Here, we can see that although "W/o-Period" employs the action-motif and trend-based scores, it still

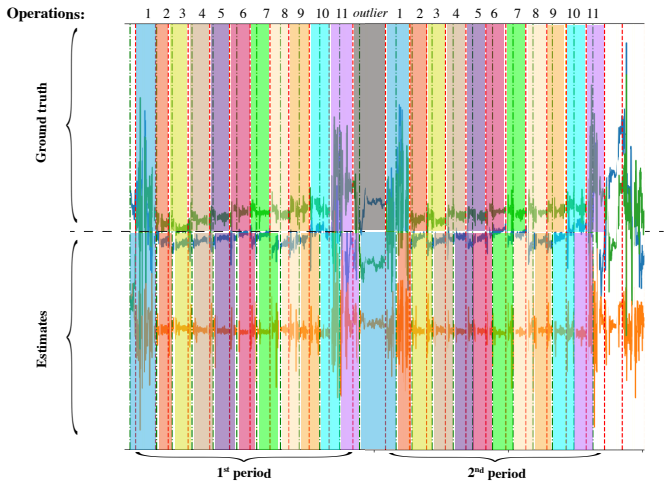


Figure 3.15: Example output from "W/o-Period" for the first and second periods of work process F

could not precisely identify the start times of the operations. A major factor in "W/o-Period"'s poor performance was its inability to detect outlier segments, which can also be seen in Figure 3.15. These results indicate that the use of dense period motifs is integral to the proposed method's ability to detect outliers.

Effect of Action-motif Score

Looking at the "W/o-Action" results in Figure 3.10, we can confirm that the action-motif score was the second most important contributor among the three scores. Note that there was not a significant difference between the overall results for "Proposed" and "W/o-Action" because "Proposed" greatly outperformed "W/o-Action" only in work processes A and E. The F-measure for work process A for "W/o-Action" was poor because we had few period motifs in this work, indicating that action motifs played an important role in this work. Figure 3.16 shows an example result for "W/o-Action" for work process E. As shown in the figure, the estimated start times of operations in the second period are slightly shifted. In this case, our method only found period motifs at the beginning and ending parts of this work process since the sensor data from operations 3 to 10 (all screwing actions) were similar to each other. Recall that

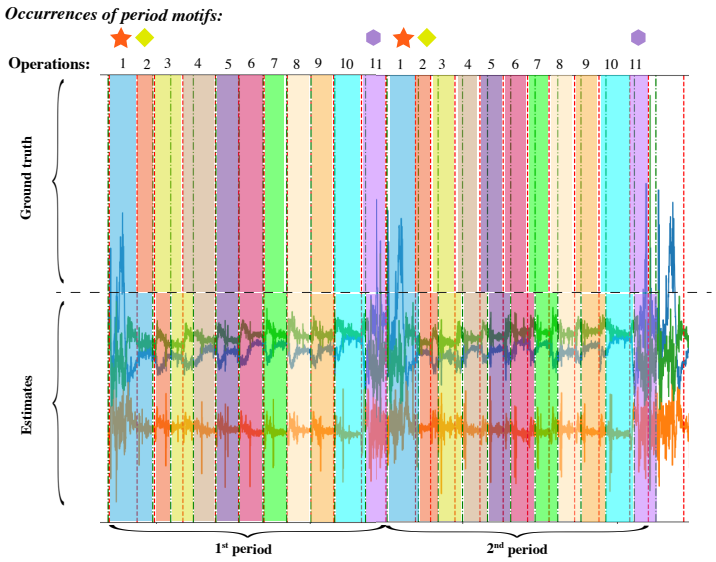


Figure 3.16: Example output from "W/o-Action" for the first and second periods of work process E

period motifs are defined as occurring only once per work period, making it unlikely to discover good candidates for them amongst so many operations that consist largely of similar actions. Therefore, it was difficult to precisely detect these operations without using the action-motif score.

Verifying of Action Motif Selection

To verify the performance of action motif selection, we compared an action motif selected by our method to several action motifs randomly selected in the dataset of work process A. In practice, we calculate the F-measures of the proposed method using different action motifs, verifying if the proposed action motif selected by our method (i.e., the "Proposed" in Figure 3.17) is better than other motifs that were chosen randomly (i.e., the "Motif_1," "Motif_2" and "Motif_3" in Figure 3.17). Figure 3.17 shows the F-measures of work process A when using different action motifs. Figure 3.18 breaks down the average F-measures from Figure 3.17 into the individual F-measures for each work period. As shown in the results, "Proposed," in which the proposed action motif was applied, reaches the highest overall accuracy. The results of "Motif_1" and "Mo-

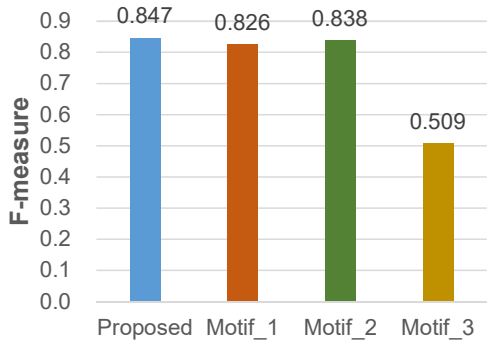


Figure 3.17: F-measures of the "Proposed" method for work process A when using best action motif (Proposed) and other action motifs (Motif_1, Motif_2 and Motif_3) that randomly selected.

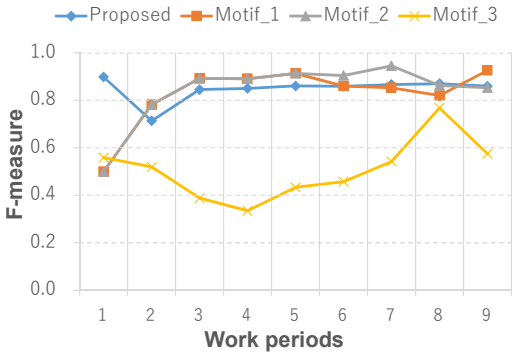


Figure 3.18: Transitions in F-measures for each method for work process A when using best action motif (Proposed) and randomly selected action motifs (Motif_1, Motif_2 and Motif_3).

tif_2" are similar to that of "Proposed," while the result of "Motif_3" is much poorer than that of "Proposed," indicating that it is necessary to select a good action motif to maintain high accuracy. In particular, because the corresponding similarity time series S (S represents the similarity values of a motif to the whole series data, refer to Subsection 3.3.4) of "Motif_1" and "Motif_2" are similar in every period, the "action-motif score," which is calculated from these two motifs, can help detect consistent operation segments within periods of interests, and therefore, contributes to the recognition accuracy. In contrast, the similarity time series corresponding to the "Motif_3" are different in different periods, making it difficult to find same operations in different periods. This experiment suggests that a good action motif should have the corresponding time series S similar in every period, which can be identified by our proposed method. While it is not guaranteed that our method can always select the best action motif since we did not verify the performance of every action motif for each data set.

Limitation of Proposed Method

From Figure 3.14 we can see that the transition accuracy of work process A shows a higher F-measure on "Xia2019" compared to the "Proposed" method in several periods. This is because the "Proposed" method, which employs "action-motif score,"

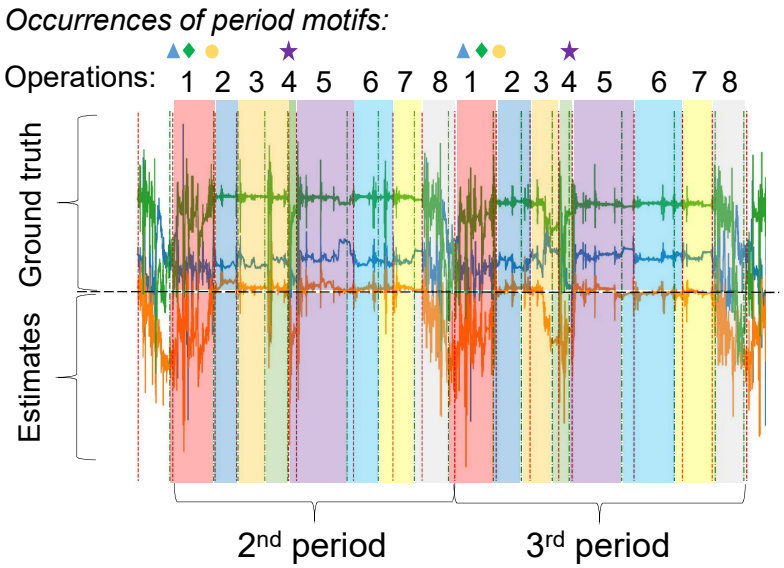


Figure 3.19: Example output from "Proposed" for the second and third periods of work process A.

cannot precisely compare data similarity across different periods. Specifically, when the variance of the operation durations in the dataset is large, the accuracy of the "Proposed" method is poorer than that of "Xia2019". Figure 3.19 presents the estimation results of operation start times of periods 2 and 3 by using the "Proposed" method. As shown in this result, the ground truth duration of the third operation in the 2nd period is about two times longer than that of the 3rd period, resulting in the wrong estimated operation start times of the forth operation in the 2nd period, which means that the "Proposed" method fails to detect consistent operation segments on these periods if the variance of operations among different periods is large.

Computation Time

Tables 3.3 and 3.4 show computation times for each process for "Proposed" and "Xia2019" along with their total computation times. As shown in these results, the total computation time for "Proposed" is much shorter than that of "Xia2019," which relies on a

computationally expensive technique, i.e., DTW[2]. In contrast, "Proposed" leverages period and action motifs to quickly detect the start times for the operations. In fact, the total computation time for "Proposed" is shorter than the duration of the input sensor data, indicating the possibility for its use in real-time unsupervised activity recognition. Since the computation time for HDP-HMM currently dominates the overall computation time for "Proposed," we plan to replace HDP-HMM with a lightweight technique as a part of our future work.

3.4.4 Discussion

In this section, we discuss the performance of our proposed method compared with supervised learning methods. Based on our results ("W/o-Sens" in Chapter 2 and "W/o-Action" in Chapter 3), we noticed that the sensor data similarity of same operations in different periods is high, indicating that it is easy to reach a high recognition accuracy by using supervised learning methods if we have enough data for training models. However, labeling every operation for different workers is time-consuming, which is impractical in real factory settings.

Besides, in the paper of [41], a supervised method has also been evaluated, where a sensor data segment of a starting time of the first operation is given. By tracking the segment using the particle filter tracker based on DTW, this method finds the start time of each period. In comparison, the result of the supervised method performs slightly poorer than the proposed unsupervised method over the same datasets. Our study also uses similar datasets that collected from the same factory and a similar particle filter to track motifs in every period, indicating that our proposed method that uses an unsupervised learning approach has sufficient accuracy for identifying operations and occurrence of outlying activities, which can help the manager estimate the work performance of factory workers efficiently.

[2]This method employs DTW to compute the consistency of sensor data for an estimated operation between different work periods.

3.5 Conclusion

In this chapter, we proposed a structure-based unsupervised factory activity recognition method that is robust to outlier sensor data segments. Our method robustly estimates the start time for each operation in a work period by leveraging two types of motifs: period motifs and action motifs. We evaluated the proposed method using sensor data collected from workers in actual factories and achieved state-of-the-art performance. In addition, we confirmed that the computation time of the proposed method is significantly shorter than the similarity-based method.

As a part of our future work, we plan to implement a lightweight HMM algorithm to reduce the time consumption on detecting the trend changes, and design a real-time factory activity recognition system based on our method in an actual factory. In addition, we plan to take the advantages of both the similarity-based and structure-based approaches, enabling the fusion of these two approaches to be robust and efficient in factory work activity recognition in real working scenarios.

Table 3.3: Computation time [s] of "Proposed"

				Tracking Operation Start Times					
	Preprocess (HDP-HMM)	**Motif Discovery**	**Tracking Period Motifs**	**Trend-based Score**	**Period-motif Score**	**Action-motif Score**	**Manipulate Tree+Others**	**Others**	**Total**
work A	353.5	3.9	6.2	0.007	1.6	3.5	39.3	4.1	**412.1**
B	187.8	0.5	5.3	0.005	0.7	2.1	74.1	3.1	**273.6**
C	230.2	0.5	6.9	0.009	1.8	3.5	181.5	3.0	**427.3**
D	189.8	0.5	3.0	0.002	1.3	3.2	69.7	3.0	**270.5**
E	224.5	0.5	4.0	0.014	2.2	4.3	46.8	3.5	**285.7**
F	210.8	0.5	0.9	0.013	1.7	3.8	56.5	4.3	**278.5**

Table 3.4: Computation time [s] of Xia2019

	Preprocess (HDP-HMM)	**Motif Discovery**	**Tracking Operation Start Times**	**Others**	**Total**
work A	362.8	0.7	1914.4	9.4	**2287.3**
B	223.0	0.1	3038.0	1.0	**3262.1**
C	221.3	0.1	1450.0	4.0	**1675.4**
D	194.7	0.1	4558.8	2.7	**4756.3**
E	188.2	0.1	753.2	3.1	**944.8**
F	194.6	0.1	892.7	27.0	**1114.4**

Chapter 4

A Supporting Technique for Identifying Skill Knowledge

In this chapter, we describe a supporting method for identifying significant activity differences between skilled and unskilled factory workers by a neural network with an attention mechanism using wrist-worn accelerometer sensor data collected in real manufacturing, enabling factory managers to obtain skill knowledge easier.

4.1 Introduction

In a factory, the skill level of workers influences the quality of products and deteriorate work productivity [69]. Therefore, it is an urgent need for industrial engineers to help unskilled workers to improve their work skills. In the current situation, engineers manually compare the unskilled worker with the skilled worker by watching videos or sensor data collected from every worker to find activity differences between them. By learning from the activity differences, the unskilled workers can modify their movements based on the skilled workers to get to the higher skill level fast [24]. Unfortunately, there are many workers in a factory, which is time-consuming for industrial engineers to manually locate the differences in minutes between skilled and unskilled groups.

Figure 4.1 shows two segments of raw sensor data collected from the right wrists of skilled and unskilled workers. As shown in the figure, a complete work process (i.e., work period) is comprised of a sequence of operations, such as "$Scan_1$", "Set",

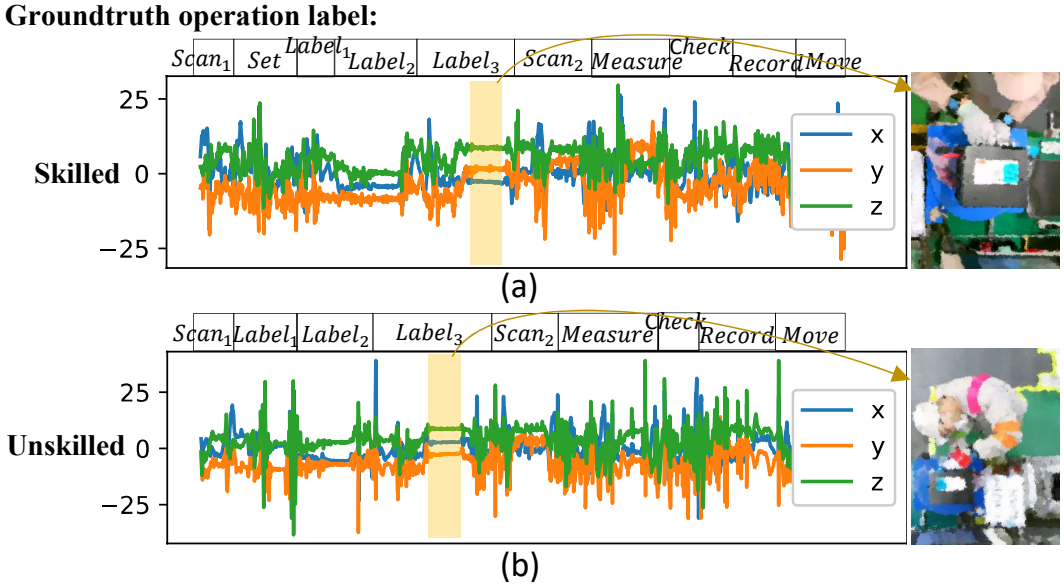


Figure 4.1: An example of a series of operations collected from two workers' right wrists by using a 3-axes accelerometer. The rectangles indicate an activity difference between workers, which implies a skill knowledge about working posture to reduce body burden

"$Label_1$", etc. However, some activities performed in an operation for the skilled and unskilled workers are different. For example, the pictures in Figure 4.1 presents different postures for the skilled and unskilled workers of attaching a label in the "$Label_3$" operation, where the skilled worker bends his spine for a smaller degree to perform the work satisfactorily. In contrast, the unskilled worker is more likely to suffer back pain caused by the large curvature of the spine. After discovering the differences between the skilled and unskilled workers, the industrial engineer can guide the unskilled workers for better performance. This chapter aims to identify activity differences between the skilled and unskilled workers by using acceleration data collected from the workers' wrists. However, it is challenging to compare activity differences by simply calculating sensor data differences. Since even the same activities in different periods performed by the same worker can cause sensor data differences, we focus on identifying candidates of segments with significant activity differences between the skilled and unskilled workers.

In this chapter, we propose a neural network model based on the attention mechanism to help the engineer quickly find candidates of activity differences between the skilled and unskilled workers by using acceleration data collected from the workers' wrists. A neural network that is trained to classify a time-series for a period data into skilled and unskilled workers captures differences between the two classes, and the attention mechanism weights the input segments based on the data differences between the two classes. Therefore, we can extract segments of the input series with high attention values from attention layers as the candidate segments, which are likely to contain activity differences. However, attention only represents the weight of its corresponding input instance, after getting a candidate segment from an input in a worker class, it is necessary to identify the corresponding segment on the other worker class (e.g., as shown in Figure 4.1, if a candidate is an attaching label action in the "$Label_3$" for the skilled worker, the engineer should locate that action of the unskilled worker to compare the differences). However, it is difficult to directly find corresponding segments using raw sensor data, as the activities done by different workers are different. In this chapter, we leverage a clustering algorithm for the latent representations in the neural network to roughly eliminate individual differences. So that latent representations with similar activities can be clustered into the same cluster. Finally, the system will output every candidate of activity differences and their corresponding segments at the other worker class to the industrial engineer for skill assessment. We assume that significant differences between skilled and unskilled workers are more likely to contain skill knowledge that can help the unskilled workers promote their skills. So, we ask an industrial engineer to assess the skill knowledge inside the candidate segments detected by our proposed method to analyze our method qualitatively.

The contributions in this chapter are summarized as follows.

- Our method provides an attention-based neural network to automatically detect candidates of activity differences between skilled and unskilled workers from their acceleration data, which reduces the time costs for engineers to spend on navigating activity differences.
- We propose a novel way of detecting corresponding activities between skilled and unskilled workers by using a clustering approach for the latent representations of the neural network by eliminating individual differences.

In the rest of this section, we first review studies for evaluating work performance. We then present the design of the proposed method and evaluate the method using sensor data collected in actual factories.

4.2 Related Work

There is a growing interest in studying individual's work performance to help amateurs improve their skill level fast. Many of the earliest work uses archival records, rating scales, and job knowledge tests [14, 7] for work performance assessment. However these metrics have drawbacks, the results reported by workers are not accurate, and some unconscious skills of workers can not be detected [14]. Therefore, recent studies try to introduce activity data collected by electronic devices to complement traditional assessments. For example, Mirjafarl et al. [49] use mobile phones, wearables, and beacons to study behavior differences between higher and lower work performers in companies. Das Swain et al. [18] leverage sensors in commodity devices to quantify the daily activities of workers.

The classifiers in the above studies [49, 18] simply classify activities into specific classes, and these studies did not pay attention to the key activities (i.e., sensor data segments) in the whole data which contribute to the classification results. In contrast, many recent studies implement an attention component in neural networks, trying to discover more detailed information. For instance, Zeng et al. [82] develop two attention models: temporal attention and sensor attention for detecting important signals and sensor modalities, respectively, which can be applied to identify the most important activities and sensor modalities for detecting Parkinson's disease. Murahari et al. [53] also propose an attention model as a data-driven approach for exploring the relevant temporal context in time-series data. Maekawa et al. [42] also present an attention-based neural network over animals' trajectories to detect segments in trajectories that are characteristic of one group, enabling biologists to focus on these specific segments and formulating new hypotheses.

Besides, there are many studies that use autoencoder to extract features for an individual dataset automatically. According to the special structure, the autoencoder transforms input data to a lower space and then retrieves it, which is very useful to compress

data and extract features [55]. For guiding the autoencoder to generate better features, Xie et al. [78] and Guo et al. [19] focus on learning feature representations and a clustering assignment simultaneously for helping the network extract better features for what they expected.

In this chapter, we arrange to combine the advantages of the attention component and autoencoder. Our neural network tries to identify activity differences between skilled and unskilled workers with high attention values, while the corresponding data segment on the other worker by clustering the latent representations of the autoencoder.

4.3 Methodology

In this chapter, at least two workers are performing one type of work: one skilled worker and one unskilled worker. The data collected from each type of work are triaxial accelerometer data from the workers' both wrists. Each worker has multiple time-series data, with each time-series corresponding to a period.

4.3.1 Preliminaries

This chapter aims to identify candidates of segments with significant activity differences between the skilled and unskilled workers (e.g., as shown in Figure 4.1, the attaching label activity performed by the skilled worker) and detect the corresponding segments of each candidate on the rest of the worker (e.g., the corresponding attaching label activity found in the unskilled worker).

There are two assumptions in this chapter:

- Skill knowledge is more likely to exist in significant activity differences between skilled and unskilled workers.
- Data segments with significant activity differences contribute more to skilled and unskilled worker classification.

According to the first assumption, we can evaluate the effectiveness of our method in identifying activity differences by validating if skill knowledge exists in the candidates of segments or not. Based on the second assumption, we design our method that

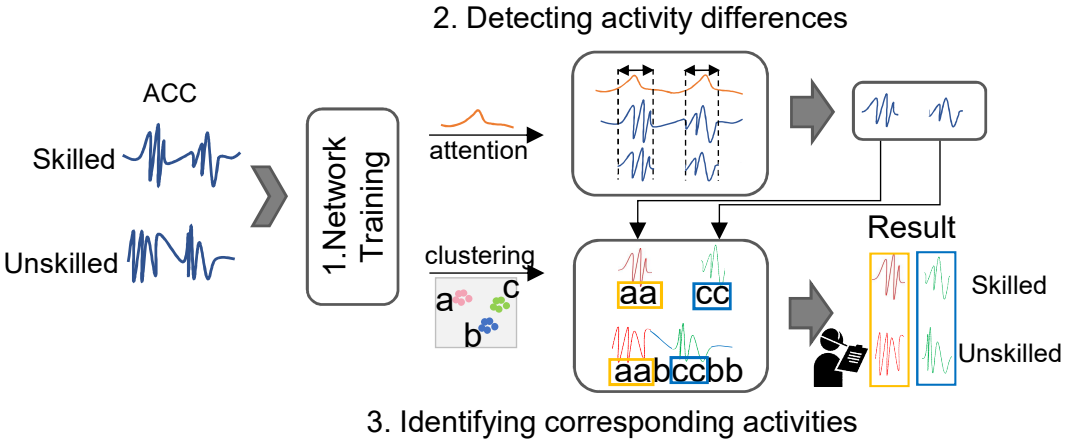


Figure 4.2: Overview of the proposed method

classifies a time series into a skill or unskilled class while focusing on important data segments.

4.3.2 Method Overview

Figure 4.2 shows an overview of the proposed method, which mainly consists of three phases: (1) network training, (2) detecting activity differences, and (3) identifying corresponding activities. Initially, we collect acceleration data from both skilled and unskilled workers' wrists for training the network. After training, we calculate and keep attention values of each hidden layer as well as the latent representations of the network for the next step. In the *detecting activity differences* phase, we select a representative input (period) for each class. For each representative, we extract segments with high attention values as candidate segments of activity differences. The *identifying corresponding activities* phase first symbolizes input series according to the clustering result, then, for each candidate of segment in the previous phase, finds the corresponding segment on the other worker class's representative input. Finally, the system will output a pair of segments for each candidate of segment. The pair of segments is composed of a candidate of segment and the corresponding segment on the other worker class's representative input. With the help of the paired segments, the industrial engineer can easily compare the activity differences between skilled and unskilled workers and iden-

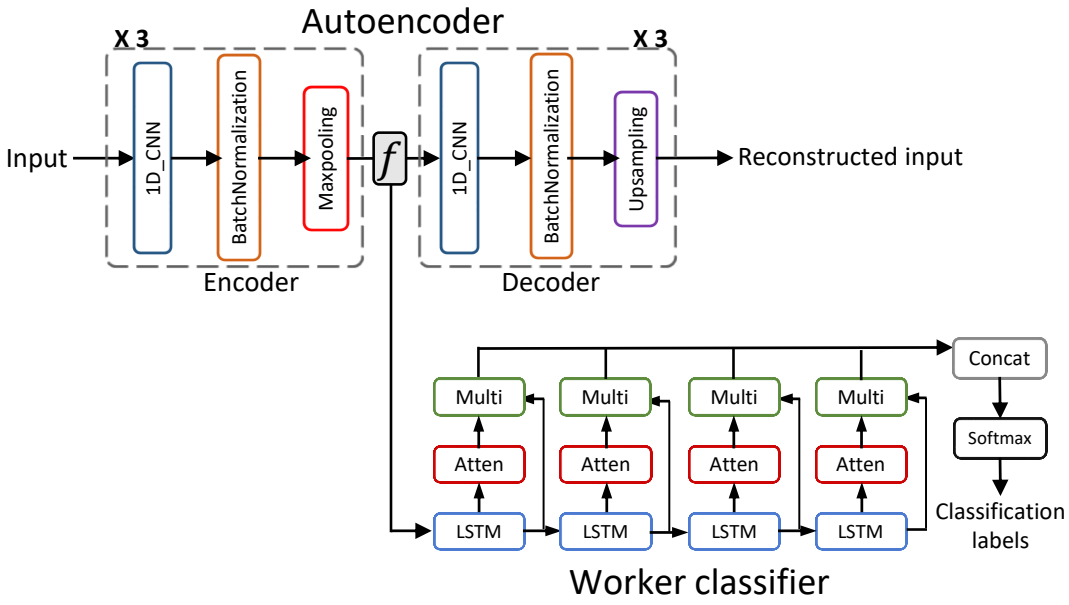


Figure 4.3: Overview of the neural network

tify useful skill knowledge from them.

4.3.3 Network Training

The neural network shown in Figure 4.3 is composed of two parts: (1) an autoencoder that extracts latent representations of input series and (2) a skilled and unskilled worker classifier to fine-tune the latent representations to detect activity differences between the two classes.

We assume that X denotes an input sequence of the neural network, which corresponds to a period of data for a worker. The class label $y \in [0, 1]$ is associated with every X, where 0 and 1 correspond to the skilled and unskilled workers, respectively.

Network Architecture

In order to detect candidate segments of activity differences, we employ an attention-based neural network to classify time-series from the skilled and unskilled workers, as shown in Figure 4.3. The autoencoder architecture consists of three encoding blocks as well as three decoding blocks. A single encoder block consists of a 1D-CNN layer

Table 4.1: Summary of hyper-parameter values

Name	Parameter
1D_CNN	4 kernels, kernel size is 16
Maxpooling	Window size is 2
UpSampling1D	Window size is 2
LSTM	8 hidden units
Batch size	32
Input dimension	Varies for each dataset

plus a BatchNormalization layer and a MaxPooling layer. A decoder block consists of a 1D-CNN layer plus a BatchNormalization layer and an UpSampling1D layer. The 1D-CNN layer captures a salient waveform from the input, and the MaxPooling layer compresses the size of the 1D-CNN outputs by choosing the maximum value within a scale. By using the two layers, we can get the latent representation f, where the feature vector at each timestep corresponds to a short-term data within input series. Besides, in the worker classifier architecture, four stacks of LSTM layers are connected to the encoder's output for extracting long-term dependencies in the data used for classifying skilled and unskilled workers. Blocks labeled "LSTM" include LSTM and BatchNormalization layers. Blocks named "Atten" processes the output of every "LSTM" using Eq. 4.1, which calculates the attention weight of the "LSTM" layer output. For every time-series inputs, a corresponding attention series is computed in each "Atten" layer, with the input series with a higher attention weight being more important over the whole input series for the skilled and unskilled worker classification. Blocks labeled "Multi" multiply the attention and the outputs of the "LSTM" to emphasize important timings for classification. Blocks "Concatenate" and "Softmax" refer to the concatenate and softmax layers, respectively. Table 4.1 illustrates hyper-parameters of the neural network structure.

We assume that T denotes the length of the latent representations f, z_t is a D dimensional vector calculated by "Atten" at time t ($t \in \{1, ..., T\}$), h_t is a real-valued hidden-state vector at time t output by an "LSTM" layer. The equation of calculating attention at time t is denoted as follows:

$$\alpha_t = \exp(z_t) / \sum_{s=1}^{T} \exp(z_s), \tag{4.1}$$

$$z_t = \tanh(W h_t + b), \tag{4.2}$$

where W and b are the weight matrix and bias, respectively.

Then, we multiply attention and the output of "LSTM" in "Multi" as follows:

$$H = \sum_{t=1}^{T} \alpha_t h_t. \tag{4.3}$$

The output of the classifier is calculated by the "Softmax" block, whose target is the class label y for classifying the skilled and unskilled workers.

Network Training

The loss of the network is composed of two components: reconstruction loss L_a and binary cross-entropy loss L_c. The reconstruction loss aims to learn latent representations using the autoencoder in an unsupervised manner while preserving intrinsic local structure in data. The binary cross-entropy loss is responsible for learning features that differentiate skilled and unskilled workers. The overall loss function of the network is defined as follows:

$$L = L_a + \lambda L_c, \tag{4.4}$$

where the parameter λ controls the trade-off between L_a and L_c. In this study, λ equals to 1.

4.3.4 Detecting Activity Differences

In this phase, we select a representative input for each worker and then extract candidate segments with significant activity differences according to the attention values for each

representative. We argue that the skill of workers can be discovered from their repeated work periods, and their performance of the same task will be consistently similar [25].

To detect the consistent skill of each worker, we first look for a representative input (period) for each worker that is most similar to all the remaining input instances of the worker, in other words, the centroid of all instances. We use DTW algorithm to compare the similarity between two time-series data. For an input series X_i in the skilled or unskilled worker class, we calculate the DTW distance between X_i and each of the remaining inputs within the same class. The best X_i is supposed to have the minimal overall DTW distances in the class, selected as the centroid instance of the class.

After getting the centroid instances for both skilled and unskilled classes, the corresponding attention values of the centroid instances can be applied to extract candidate segments of activity differences. According to the structure of the worker classifier showing in Figure 4.3, there are four layer-wise attention time-series generated from each LSTM layer of the classifier, which focus on different time scales. Since a value of attention reveals the weight of the corresponding temporal position of input series in terms of the contributions to the final classification result, segments with high attention values will be more dissimilar in different worker classes. Therefore, for each attention layer, we extract candidates of input segments with the highest attention values in the top-k%.

In the next section, we will introduce how to detect the corresponding segment of each candidate on the other centroid instance.

4.3.5 Identifying Corresponding Activities

To extract skill knowledge from candidates of segments, it is important to find the corresponding segment on the other worker class to help investigate the actual activity differences between the workers. In this section, we propose a clustering approach to find the corresponding segment for each candidate with latent representations of the neural network. Because activities performed by different workers result in sensor data differences, it is impractical to find the corresponding segments by simply comparing sensor data similarity. Instead, we look for similar latent representations of the neural network.

Figure 4.4 introduces the main idea of detecting the corresponding segment for a

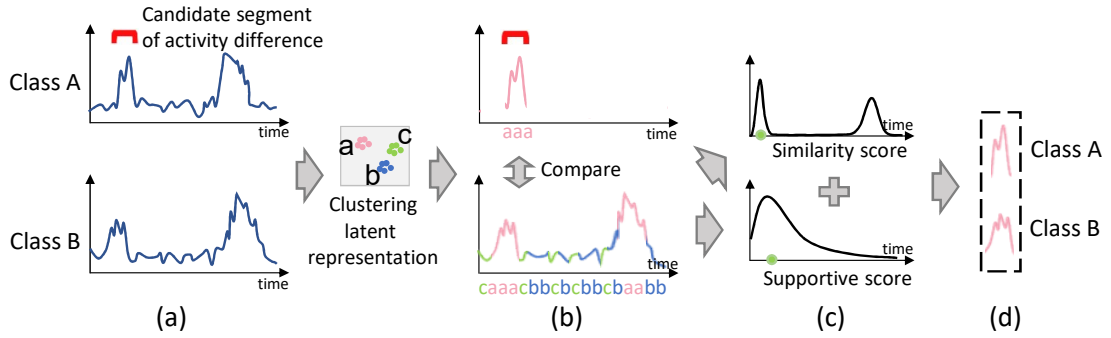


Figure 4.4: Procedure of identifying corresponding segments

candidate segment in one worker class. Since the latent representations of similar activities are supposed to be similar, we leverage the clustering method to eliminate the differences between workers so that similar activities in different workers can be clustered into the same cluster. Four steps are implemented to identify the corresponding activities. Firstly, we employ a clustering approach to cluster all data points in every timestep of the latent representations f. Then, for every centroid instance, we symbolize the raw sensor data according to the clustering result. Next, for each candidate, which is detected in Section 4.3.3, we identify the corresponding segment on the other centroid instance by using a sliding window across the whole data. Finally, each candidate and its corresponding segment will be offered to the industrial engineer for skill assessment.

Clustering Latent Representations

As Figure 4.1 shows, the raw sensor data between workers are different due to the different movements in workers. However, compared with different activities (e.g., the scan activity in the skilled worker comparing with the attaching label activity in the unskilled worker), data belonging to similar activities (e.g., the attaching label activity for both skilled and unskilled worker) has more similar latent representations. Therefore, we leverage a clustering algorithm to roughly cluster the latent representations to eliminate the difference between individuals, with similar activities being clustered to the same clusters. We assume that $f^D_{X_i,t}$ indicates the feature representation of input series X_i at time t with D channels. We leverage a k-Means algorithm to cluster every data points in

$k(\leq X)$ clusters $S = \{S_1, S_2, ..., S_k\}$. This process is formulated to find S that yields the minimum overall intra-cluster distance as follows:

$$\underset{S}{\operatorname{argmin}} \sum_{m=1}^{k} \sum_{f^D_{X_i,t} \in S_m} ||f^D_{X_i,t} - \mu_m||^2, \tag{4.5}$$

where μ_m is the center of S_m. Since the CNN layers of the autoencoder will not change the temporal relationship of the data points in the input series, the temporal relationship of data points in the latent space will still correspond to the points in the input series. Therefore, we can roughly label data points of every input series based on the clustering result of the latent representations at the corresponding timesteps.

Symbolizing Series Data

As shown in Figure 4.4, we symbolize the centroid instances of the workers based on the clustering results, where different characters represent different clusters (e.g., in the figure, as the latent representations are clustered to three clusters, which are labeled as a, b and c, the corresponding data points of the centroid instances are therefore colored to "red", "blue" and "green", respectively). We symbolize the centroid instances according to the clustering results since the clustering algorithm eliminates the small differences between workers, indicating that similar activities have the same characters.

Next, we will introduce the idea of finding the corresponding segment by using the symbols.

Finding Corresponding Segment

Now that we use the symbolized centroid instances to detect their corresponding segments. In Figure 4.4, we have a candidate segment of activity difference in class A. Our aim is to find the corresponding segment for class B. After symbolizing the centroid instances of classes A and B, we compare the symbolized candidate segment for class A with a segment within a time window from the other worker class B by sliding the time window along the complete sequence for class B. We calculate two scores: (1) similarity score and (2) supportive score, to find the corresponding segment for class B.

The similarity score represents the cluster similarity between two segments. Since latent representations of similar activities will be clustered into the same clusters, two segments with similar symbols are supposed to show similar activities. For instance, when the symbolized candidate segment for class A is "aaa", the most similar segment in "caaacbbcbcbbcbaabb" from class B will be "aaa" with the similarity score of 3. We calculate similarity score S_{simi} by counting the number of paired points with the same clusters based on the following formula:

$$C_t = \begin{cases} 1, & \text{if } S(i_t) = S(j_t), \\ 0, & \text{otherwise}, \end{cases} \tag{4.6}$$

where $S(i_t)$ and $S(j_t)$ represent the tth character of symbolized segment i and j for classes A and B, respectively.

$$S_{simi} = \sum_{t=j_1}^{j_{|j|}} C_t, \tag{4.7}$$

where $j_{|j|}$ represents the last data point in segment j.

In addition, we calculate the supportive score. Since the order of the operations done by workers is predefined by industrial engineers, the occurrence timing of the corresponding activities (in class B) should be similar to that of the candidate segment (in class A). Therefore, we employ the supportive score $Score_{supp}$ to find segments in similar timings, which are calculated as follows:

$$Score_{supp} = \frac{1}{||Seg_i - Seg_j||^2}, \tag{4.8}$$

where Seg_i and Seg_j are the elapsed times of the starting points of the segments from the beginning in series i and j, respectively.

We sum up the above two scores as the overall score. The segment with the highest overall score on the other series (i.e., class B) is supposed to be the most similar segment corresponding to the candidate segment (i.e., a candidate in class A).

Finally, the pair of a segment for each candidate and its corresponding segment will be provided to the industrial engineer for skill assessment.

Table 4.2: Overview of recorded datasets

Worker	Number of instances	Skill level	Dataset
1	38	Skilled	Screwing
2	41	Unskilled	
3	42	Skilled	Final check
4	44	Unskilled	

4.4 Evaluation

4.4.1 Dataset

We evaluated the proposed method using two datasets from four individuals working in a real factory. Every worker wears two smartwatches (Sony SmartWatch3 SWR50) on each wrist, collecting acceleration data with an approximate sampling rate of 60Hz. Table 4.2 shows an overview of the recorded datasets. In the dataset of "Screwing", workers were employed to install screws on circuit boards, consisting of many predefined operations, such as setting, screwing, recording data, etc. The workers in the dataset of "Final check" are required to check final products and record checking results, consisting of attaching labels, scanning labels, measuring box, etc. The skill level of each worker is labeled by an industrial engineer based on the workers' job performances in real manufacturing.

4.4.2 Evaluation Methodology

In order to measure if the proposed method can find the corresponding data segment on the other worker class, we calculate Mean Squared Error (MSE) between the estimated starting time of the segment on the other worker and the groundtruth. We provide the results for the proposed method against other comparative methods, leaving one score out to evaluate the effectiveness of our scoring metrics. The methods to be tested are listed as follows:

- **Proposed**: This is the proposed method.

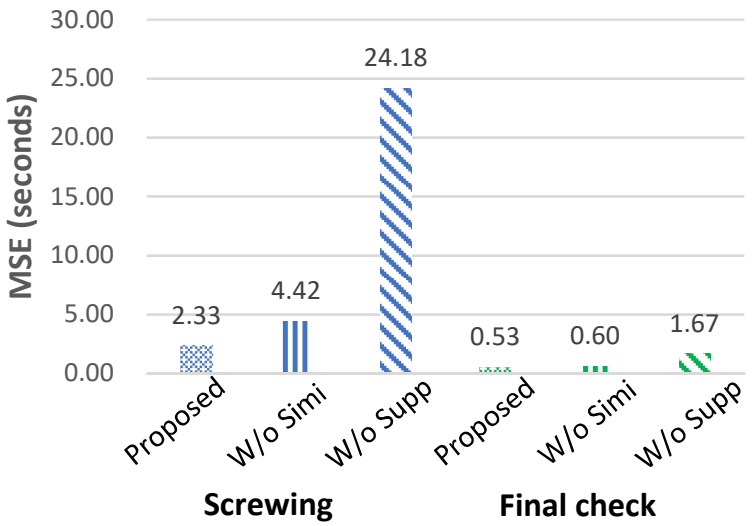


Figure 4.5: MSE of the three methods on two datasets

- **W/o Simi**: The proposed method without using the similarity score when comparing the similarity between a candidate segment and segments on the other worker class.
- **W/o Supp**: The proposed method without using the supportive score when comparing the similarity between a candidate segment and segments on the other worker class.

In addition, to qualitatively analyze the attention mechanism for detecting activity differences between skilled and unskilled workers, we discussed with industrial engineers whether the detected activity differences include skill knowledge or not. Detailed discussion is showing in Section 4.4.5.

4.4.3 Results of Identifying Corresponding Segments

As shown in Figure 4.5, we calculate the average of MSE for all candidates of segments in each dataset and compare the performance of three methods on both datasets. The proposed method achieved the lowest MSE in both datasets, which is the most robust method among the three methods. We observe that “W/o Supp” in the “Screwing”

dataset has the highest MSE, which is because that the candidate segment corresponding to a screwing action occurs several times in each instance, many segments corresponding to screwing actions on the other class will have a high similarity score to the candidate segment, but they do not belong to the same operation. "W/o Simi" also shows a high MSE on both datasets since the starting times of an activity performed in different periods are different, proving that the similarity score can fine-tune the estimated starting time of the activity on the other class by detecting the most similar data segment on the other class.

4.4.4 Performance of Network Structure

In this section, we discuss if the combination of the autoencoder and worker classifier can improve the performance of our method. As shown in Figure 4.6, we compare the Proposed network structure with the "W/o decoder" block, in order to evaluate if the autoencoder can protect the structure of latent representations similar to the input series. As can be seen, the Proposed network structure has similar distributions of latent representations between the skilled and unskilled workers, with similar activities clustered into the same cluster (e.g., the sensor data segments with blue color are corresponding to screwing activities). Whereas the "W/o decoder" block shows different distributions of latent representations between the skilled and unskilled workers, resulting in similar activities in different workers clustered into different clusters.

4.4.5 Discussion

4.4.6 Qualitatively Analyze Performance of Identifying Candidates

Since we do not have groundtruth information to evaluate the identified activity differences, we asked industrial engineers to assess if skill knowledge exists in the candidates of segments detected by the attention network, as our ultimate goal is to improve the efficiency to help the engineers to find skill knowledge. Tables 4.3 and 4.4 present the skill information of each activity candidate detected in the "Screwing" and "Final check" datasets, respectively. The "No." column represents the indices of candidate segments detected in each dataset. The "Activity description" column briefly describes the main activities performed in the corresponding candidates. The skill knowledge types

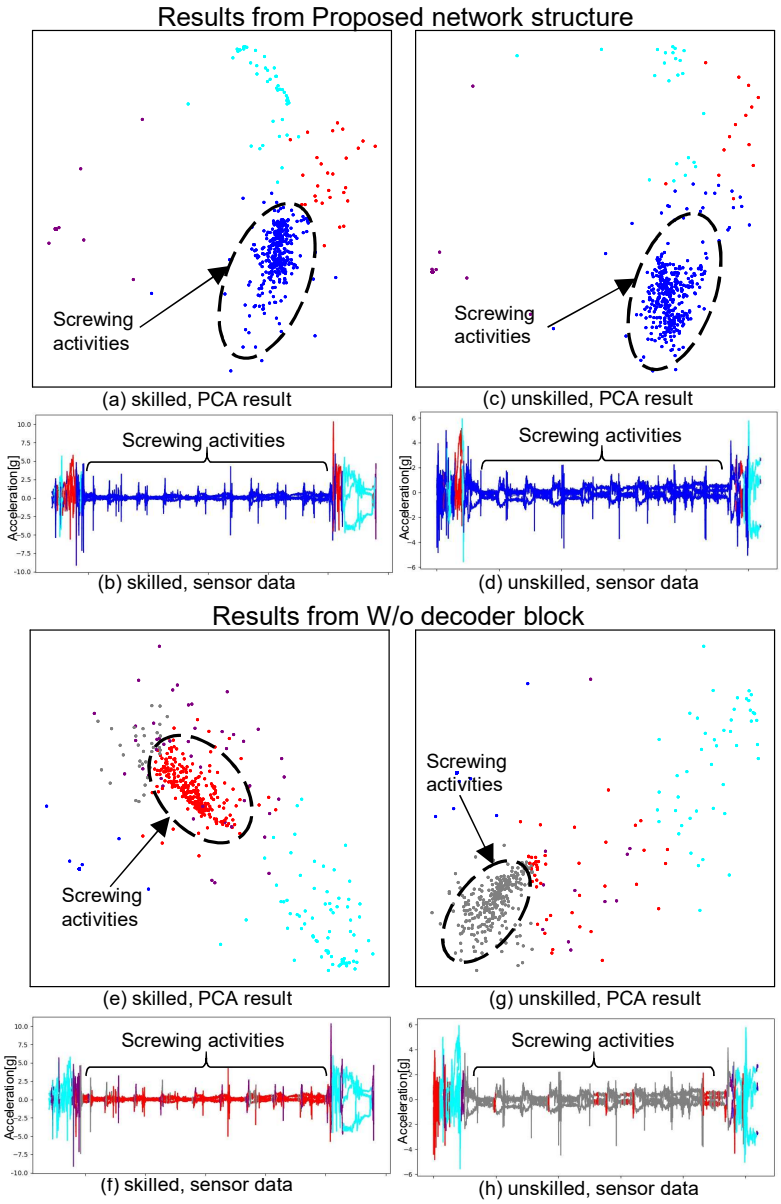


Figure 4.6: Clustering results of latent representations for “Screwing” dataset, different color shows different clusters. (a), (c), (e), and (g) show latent representations visualized by PCA. (b), (d), (f), and (h) show clustering results visualized on raw acceleration data of right wrists.

are shown in the "Skill knowledge" column, which was decided by the industrial engineers. In the "Other" column, we offer some extra information related to the activities for assessment, such as the average duration of the activities for the skilled and unskilled workers.

When determining if an activity candidate has skill knowledge, the engineer follows the "Principles of motion economy" strategy proposed in [46] to identify the skill type of the candidates. The types of skill knowledge in this chapter are listed as follows:

- Time conservation (1): Even temporary delay of work by a man or machine should not be encouraged.
- Time conservation (2): Two or more jobs should be performed upon at the same time, or two or more operations should be carried out on a job simultaneously if possible.
- Arrangement of the work place (1): Arrange the height of the workplace and chair for comfortable sitting and standing.
- Arrangement of the work place (2): Tools, materials, and controls should be located close to and in front of the operator.
- Arrangement of the work place (3): There should be a definite and fixed place for all tools and materials.

In the dataset of "Screwing", the No.2 candidate shows that both the skilled and unskilled workers have a high overall waiting time, in which the waiting time for the skilled worker is two times longer than the unskilled. The information indicates that workers upstream of the production line of the current workers have a lower work efficiency, which will deteriorate the productivity of the whole production line. According to the strategy of "Time conservation (1)", it is advised to shorten the delay by upstream workers by assigning more workers upstream the production line. No.3 and No.4 correspond to the *6-th* and the last screwing actions, respectively. The average duration of the screwing action for the skilled worker is slightly shorter than the unskilled worker since the skilled worker can use both hands for different activities simultaneously. For example, while controlling the screwing tool with the right hand, the skilled worker uses

his left hand to set a new screw into the next hole. However, the unskilled worker cannot perform these activities at the same time. Based on the idea of "Time conservation (2)", the unskilled worker should learn how to cooperate with both hands to promote work efficiency. As for the skill information in the No.1 candidate, the major difference between the workers is that they use a different hand to set circuit boards, which is not recognized as skill knowledge from the engineer's point of view.

In the dataset of "Final check", the skill knowledge of the No.1 candidate also belongs to "Time conservation (2)", where the skilled worker places his left hand on the box to control the height of the label and decides the timing to stick the label by the right hand. In contrast, the unskilled worker holds the sides of the label with both hands, spending a more extended time locating the appropriate place to stick the label. The No.2 candidate shows a different skill knowledge, in which the height of the workplace for the unskilled work is not suitable for him, so that he usually bends over to attach the label. As a result, the unskilled worker suffers a heavier body burden and feels tired soon. The "Arrangement of the work place (1)" strategy suggests that the engineer coordinates a suitable workplace for workers to reduce body burden. In the No.3 candidate, workers need to remove a label from a box, then attaching the label back to the box. After removing the label, the skilled worker places the label in front of the box, while the unskilled worker sticks the label far from the box, which takes more time to finish the activity. The corresponding strategy advises the label be placed close to the box. The main difference between workers in the No.6 candidate is that the skilled worker sometimes scans the box before he moves the box to the work table, while the industrial engineer requires workers to process the box just over the work table to ensure every operation can be performed appropriately. Therefore, the skill knowledge we observed from the No.6 candidate was not recommended by the engineer. Unfortunately, it is challenging to evaluate skill knowledge for the No.5 and No.7 candidates, as these activity candidates happen at the end of the work data without available video information.

However, there are some limitations to the attention-based network. Our method cannot detect small activity differences, which is also likely to contain useful skill information. For example, the way of holding the measuring tool is an important skill knowledge to get accurate measuring results, but the activity difference between the skilled and unskilled workers mainly different in the direction of the wrist, which does

not have high attention values.

As a result, we extracted 11.7% and 19.3% candidate segments over the whole centroid inputs in "Screwing" and "Final check" datasets, respectively, in which 7 out of 11 segments include skill information identified by the industrial engineer. The attention mechanism is a practical supporting technique to be adopted to detect skill knowledge within significant activity differences.

4.4.7 Selection of Data Sets

Here, we discuss the number of data sets applied for evaluation. This study tested two data sets in the assembly work domain: the screwing and final check tasks. These two tasks are common in the assembly work, and their activities are different, indicating that our method can efficiently find candidate data segments that potentially contain skills to some extent. The effectiveness of our method was also verified by the experiment with industrial engineers. However, the number of data sets is still limited. As part of our future work, we plan to evaluate our method on data sets of other domains such as sports and cooking.

As for the limited number of workers in each data set, the discovered findings can be confident since the skill knowledge identified from the paired workers is extracted from many periods.

4.5 Conclusion

In this chapter, we proposes an attention-based neural network to identify significant activity differences between workers, which is applied to support industrial engineers to find skill knowledge from workers. We employ the attention mechanism to emphasize and visualize important input segments and design a clustering method to eliminate individual differences. According to the candidates of activity differences detected by attentions, the industrial engineer can find useful skill knowledge from workers.

We identified that our method could not detect knowledge regarding a small range of grabbing a tool that the engineer detected. As a part of our future work, we plan to improve our method to detect such actions by combining it with vision sensors. For example, skeleton tracking is available on Kinect. It is possible to detect the range of

movement of the body part by using the Kinect, which can be applied to weigh the candidate activities identified by our method using sensor data. In addition, we plan to verify our method on public data sets in our future work.

Table 4.3: Skill information for the dataset of Screwing

No.	Activity description	Skill knowledge	Other
1	Set box and push button	-	Avg. duration of skilled and unskilled (6.7s/7.2s)
2	Wait for the next box	Time conservation (1)	Total duration of skilled and unskilled (296.6s/102.4s)
3	Screwing	Time conservation (2)	Avg. duration of skilled and unskilled (4.4s/4.6s)
4	Screwing	Time conservation (2)	Avg. duration of skilled and unskilled (4.6s/5.8s)

Table 4.4: Skill information for the dataset of Final check

No.	Activity description	Skill knowledge	Other
1	Attach a small label on the box	Time conservation (2)	Avg. duration of skilled and unskilled (6.6s/8.9s)
2	Attach a large label on the box	Arrangement of the work place (1)	Avg. duration of skilled and unskilled (8.9s/10.8s)
3	Stick the large label on table	Arrangement of the work place (2)	-
4	Rotate the box to check information	-	Avg. duration of skilled and unskilled (1.7s/2.3s)
5	Bring the box to other place	-	No camera information
6	Set the box on the table	Arrangement of the work place (3)	Change activity order, not recommended
7	Bring the box to other place	-	No camera information

Chapter 5

Conclusion and Future Work

5.1 Thesis Summary

In this thesis, we developed several approaches for the analysis of acceleration data collected from factory workers, aiming to improve manufacturing productivity from two aspects: streamlining the work process and promoting skill training for workers.

To summarize this thesis, we revisit the research goals mentioned in Chapter 1. These goals were accomplished as the following.

- **Recognizing the starting time and duration of every operations done by workers in an unsupervised manner**. To streamline the work processes, it is necessary to know the actual starting time and duration of each operation in every period done by individual workers. In Chapter 2, we proposed a similarity-based approach to identify the starting time of operations in an unsupervised manner. Instead of recognizing operations using supervised learning methods, which were time-consuming in data collection and labeling, this chapter proposed a particle filter to track the estimation of operation start times. The particle filter generated several paths, in which every particle represented an estimated starting time of operation. We leveraged four scores to weigh each path generated from the particle filter: (i) semantic-based score, (ii) sensor-based score, (iii) trend-based score, and (iv) motif-based score. The best path was supposed to have the highest overall scores, and we can estimate the starting time of operations based on the particles of the based path. The duration of each operation can be calculated as the

temporal distance between its corresponding particle and the following particle.

This chapter explored the use of sensor data similarity between periods and similarity between sensor data as well as word descriptions in instructions as scoring criteria to evaluate paths, enabling factory activity recognition performed in an unsupervised manner.

- **Robust unsupervised operation recognition by sensor data with the temporal structure of multiple motifs**. For the factory workers working in a real industrial setting, outlying activities usually occur. The performance of the similarity-based approach will deteriorate when encountering outlier data, as there is no prior knowledge about the sensor data and semantic descriptions of the outliers. In Chapter 3, we proposed a structure-based approach to deal with the outlying activities. In particular, this chapter proposed two types of motifs: period motif and action motif, to roughly identify the starting time of operations and the occurrences of outliers. The period motif, which only occurs once in a period, can roughly locate the starting time of the corresponding operations in a period, with the consistent temporal relationship between the occurrences of period motifs to detect the occurrences of outlying activities. Besides, the action motif, which exists multiple times in a period, is used to calculate statistical features of sensor data segments with a smaller computation cost than calculating the sensor-based score in the similarity-based method.

  This chapter furthermore explored an unsupervised learning approach to robustly identify the starting times of operations by using the temporal structure of multiple period motifs, in conjunction with action motifs to reduce computation costs.

- **A supporting method for identifying skill knowledge in the activity differences between skilled and unskilled workers in factories**. Manually identifying skill knowledge from activity differences between workers by industrial engineers is time-consuming. In Chapter 4, we designed an attention-based neural network to extract candidates of activity differences between the skilled and unskilled workers to help the industrial engineers improve their efficiency. We identified candidates of sensor data segments with significant activity differences based on attention values calculated from the network. Besides, a clustering al-

gorithm was applied to find the corresponding segment of the other worker for each candidate segment. Our method offered pairs of segments, with each pair consisting of a candidate segment and its corresponding segment to the industrial engineers to help find skill knowledge more efficiently.

This chapter proposed a supporting technique for industrial engineers' manual works in extracting skill knowledge from workers by identifying candidates of segments with activity differences that are likely to contain skill knowledge.

5.2 Future Work

In this thesis, we presented techniques for improving manufacturing in both work process and skill training optimizations. In the following, we will summarize some possible directions for further improvements.

A fusion of the similarity-based and structure-based approaches. In this thesis, we explored two approaches to recognize the assembly work done by factory workers using wearable sensor data. However, the similarity-based approach could not deal with outlying activities, and the structure-based approach has a lower accuracy when identifying the starting time of operations in those periods without outliers encountered. Therefore, we plan to take advantage of both approaches to achieve robust factory activity recognition by using acceleration data, which can accurately recognize the starting time of operations even when outliers occur.

A real-time recognition system. In this thesis, the similarity-based and structure-based approaches were designed to process data offline. However, factory managers need an approach to roughly catch and report risks in time. In Chapter 3, we confirmed that the proposed method was possible to work in real time since the total computation time for the proposed method was shorter than the duration of the input sensor data, while the above methods were not designed for real-time recognition. In the future, we plan to design an online approach to recognize operations and detect the occurrences of outliers in real-time. Besides, we plan to implement a lightweight HMM algorithm (for now, it is the most time-consuming part for the structure-based approach) to calculate the trend changes of data with lower computation costs.

Improving performance of the attention-based neural network. This thesis pro-

posed an approach to identify candidates of segments with significant activity differences between skilled and unskilled workers, which helped industrial engineers find skill knowledge more efficiently by focusing on these candidates rather than the whole sensor data. However, the candidates of segments identified from the attention values in our neural network could only detect the segments with significant activity differences, even though some important skill knowledge may exist in small activity differences. In the future, we plan to improve the attention-based neural network to be able to identify candidates of segments with small activity differences containing skill knowledge while reducing false-positive errors of the system.

Acknowledgment

I would like to thank the many people who have supported me over the past three years as a Ph.D. student and helped facilitate this work.

First and foremost, I would like to express my deep and sincere gratitude to my research supervisor, professor Takuya Maekawa, for giving me the opportunity to do research at Hara Lab, Osaka University, and providing invaluable guidance throughout this research. It was a great privilege and honor to work and study under his guidance here. At the beginning of my Ph.D., he taught me how to schedule my thesis work within small steps that need to be achieved every week. During the time we had numerous meetings, discussions, and arguments when suffering from challenges, we finally addressed all these problems and made the thesis possible. I would not have completed this thesis without his enormous patience and persistence. I am also extremely grateful for his guidance on proposing research topics, dealing with problems, and writing research papers. The advice is very helpful for me to avoid many problems and traps in my research, encouraging me to continue research throughout my life.

I am also thankful to professor Takahiro Hara for sharing his time and expertise and providing constructive advice on my work. His wisdom and generosity have helped me through the difficult times in my Ph.D.

I would like to express my sincere gratitude to the committee members for my thesis. Professors Toru Fujiwara, Yasuyuki Matsushita, Shinji Shimojo, and Makoto Onizuka gave me many insightful and constructive comments that considerably improved the quality of the thesis.

My sincere thanks also go to professors Daichi Amagata, Kenji Yasunaga, Naoto Yanai, Yuki Arase, Chuan Xiao, Yuya Sasaki, Fumio Okura, Hiroaki Santo, Susumu Date, Norihiro Hagita, Takahiro Miyashita, and Satoru Satake, for giving me sincere advice and support on my research during the entrance examination, mid-term defenses, and doctor pre-defense.

I would like to thank all of my collaborators, lab-mates, friends, especially Joseph Korpela, Teerawat Kumrai, Jaime Morales, Zhi Li, Zesheng Cai, Heng Kim and Yihong Zhang, for providing me with invaluable discussions and ideas, and Naoya Yoshimura, Thilina Dissanayake, Zhaoye Zhang, Yiming Tian, Heng Zhou, Yuchen Ji, Yue Zhao

and Renée Schulz for helping to program and sharing their research experience with me.

Beyong that, this thesis is supported by Toshiba Corporation, Corporate Manufacturing Engineering Center. I am thankful to all Toshiba collaborators, especially Dr. Yasuo Namioka, Atsushi Wada, and Takanori Yoshii gave me many technical supports to help me cope with the problems regarding real manufacturing. In addition, I am also thankful to Ms. Chikako Kawabe, Kana Yasuda, Makiko Higashinaka and other members for helping me label data and other issues, ensuring everything in my research goes well.

Last but not least, I want to express my deepest thanks to my family. Without their love and support, this thesis would not have happened. I want to thank my parents for always supporting my endeavors and adventures throughout my life.

REFERENCE

[1] M. Aehnelt and S. Bader. Information assistance for smart assembly stations. In *Proceedings of the International Conference on Agents and Artificial Intelligence*, volume 2 of *ICAART 2015*, page 143–150, 2015.

[2] M. Aehnelt, E. Gutzeit, and B. Urban. Using activity recognition for the tracking of assembly processes: Challenges and requirements. In *Proceedings of the 1st International Workshop on Sensor-Based Activity Recognition and Interaction*, iWOAR '14, pages 12–21, 2014.

[3] M. Aehnelt and B. Urban. The knowledge gap: Providing situation-aware information assistance on the shop floor. In *Proceedings of the 2015 International Conference on HCI in Business*, pages 232–243, 2015.

[4] M. Aehnelt and K. Wegner. Learn but work! towards self-directed learning at mobile assembly workplaces. In *Proceedings of the 15th International Conference on Knowledge Technologies and Data-Driven Business*, i-KNOW '15, 2015.

[5] D. Arita, H. Yoshimatsu, and R. ichiro Taniguchi. Frequent motion pattern extraction for motion recognition in real-time human proxy. In *Proceedings of JSAI Workshop on Conversational Informatics*, pages 25–30, 2005.

[6] S. Bader, F. Krüger, and T. Kirste. Computational causal behaviour models for assisted manufacturing. In *Proceedings of the 2nd International Workshop on Sensor-Based Activity Recognition and Interaction*, iWOAR '15, 2015.

[7] A. B. Bakker, M. Tims, and D. Derks. Proactive personality and job performance: The role of job crafting and work engagement. *Human Relations*, 65(10):1359–1378, 2012.

[8] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Proceedings of the 2nd International Conference on Pervasive Computing*, Pervasive '04, pages 1–17, 2004.

[9] M. Bauer, L. Jendoubi, and O. Siemoneit. Smart factory–mobile computing in production environments. In *Proceedings of the MobiSys 2004 Workshop on Applications of Mobile Embedded Systems*, WAMES '04, 2004.

[10] J. Berg and M. Lässig. Local graph alignment and motif search in biological networks. *Proceedings of the National Academy of Sciences*, 101(41):14689–14694, 2004.

[11] E. Berlin and K. Van Laerhoven. Detecting leisure activities with dense motif discovery. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, page 250–259, 2012.

[12] M. Blum, A. Pentland, and G. Troster. Insense: Interest-based life logging. *IEEE MultiMedia*, 13(4):40–48, Oct 2006.

[13] A. Brajdic and R. Harle. Walk detection and step counting on unconstrained smartphones. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, page 225–234, 2013.

[14] J. P. Campbell, J. J. Mchenry, and L. L. Wise. Modeling job performance in a population of jobs. *Personnel Psychology*, 43(2):313–575, 1990.

[15] J. Chen, A. H. Kam, J. Zhang, N. Liu, and L. Shue. Bathroom activity monitoring based on sound. In *Proceedings of the 3rd International Conference on Pervasive Computing*, Pervasive '05, pages 47–61, 2005.

[16] B. Chiu, E. Keogh, and S. Lonardi. Probabilistic discovery of time series motifs. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, page 493–498, 2003.

[17] M. F. Chou and D. Schwartz. Biological sequence motif discovery using motif-x. *Current Protocols in Bioinformatics*, 35(1):13.15.1–13.15.24, 2011.

[18] V. Das Swain, K. Saha, H. Rajvanshy, A. Sirigiri, J. M. Gregg, S. Lin, G. J. Martinez, S. M. Mattingly, S. Mirjafari, R. Mulukutla, S. Nepal, K. Nies, M. D. Reddy, P. Robles-Granda, A. T. Campbell, N. V. Chawla, S. D'Mello, A. K. Dey, K. Jiang, Q. Liu, G. Mark, E. Moskal, A. Striegel, L. Tay, G. D. Abowd, and M. De Choudhury. A multisensor person-centered approach to understand the role of daily activities in job performance with organizational personas. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(4), Dec. 2019.

[19] X. Guo, L. Gao, X. Liu, and J. Yin. Improved deep embedded clustering with local structure preservation. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI '17, pages 1753–1759, 2017.

[20] A. Helman. *The Finest Peaks-Prominence and Other Mountain Measures*. Trafford Publishing, 2005.

[21] L. M. Hough and F. L. Oswald. Personality testing and industrial–organizational psychology: Reflections, progress, and prospects. *Industrial and Organizational Psychology*, 1(3):272–290, 2008.

[22] T. Huynh, M. Fritz, and B. Schiele. Discovery of activity patterns using topic models. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, page 10–19, 2008.

[23] M. J. Johnson and A. S. Willsky. Bayesian nonparametric hidden semi-markov models. *Journal of Machine Learning Research*, 14(1):673–701, Feb. 2013.

[24] Y. Kawase and M. Hashimoto. Analysis of skill improvement process based on movement of gaze and hand in assembly task. In *Proceedings of 17th International Conference on Computer Analysis of Images and Patterns*, CAIP '19, pages 15–26, 2019.

[25] A. Khan, S. Mellor, R. King, B. Janko, W. Harwin, R. S. Sherratt, I. Craddock, and T. Plötz. Generalized and efficient skill assessment from IMU data with applications in gymnastics and medical training. *ACM Transactions on Computing for Healthcare*, 2(1), Dec. 2021.

[26] O. Korn, A. Schmidt, and T. Hörz. The potentials of in-situ-projection for augmented workplaces in production: A study with impaired persons. In *Proceedings of the CHI'13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, page 979–984. 2013.

[27] J. Korpela, R. Miyaji, T. Maekawa, K. Nozaki, and H. Tamagawa. Evaluating tooth brushing performance with smartphone sound data. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, page 109–120, 2015.

[28] J. Korpela, K. Takase, T. Hirashima, T. Maekawa, J. Eberle, D. Chakraborty, and K. Aberer. An energy-aware method for the joint recognition of activities and gestures using wearable sensors. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, ISWC '15, page 101–108, 2015.

[29] H. Koskimaki, V. Huikari, P. Siirtola, P. Laurinen, and J. Roning. Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines. In *Proceedings of the 17th Mediterranean Conference on Control and Automation*, MED '09, pages 401–405, June 2009.

[30] F. Krüger, M. Nyolt, K. Yordanova, A. Hein, and T. Kirste. Computational state space models for activity and intention recognition. a feasibility study. *PLOS ONE*, 9(11):1–24, 11 2014.

[31] T. Kudo, K. Yamamoto, and Y. Matsumoto. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, EMNLP '04, pages 230–237, July 2004.

[32] J. Lester, T. Choudhury, and G. Borriello. A practical approach to recognizing physical activities. In *Proceedings of the 4th International Conference on Pervasive Computing*, Pervasive '06, pages 1–16, 2006.

[33] J. Lin, E. Keogh, S. Lonardi, and P. Patel. Finding motifs in time series. In *Proceedings of the 2nd Workshop on Temporal Data Mining*, pages 53–68, 2002.

[34] B. Lotter and H.-P. Wiendahl. *Montage in der industriellen Produktion: Ein Handbuch für die Praxis*. Springer-Verlag, 2013.

[35] D. Lucke, C. Constantinescu, and E. Westkämper. Smart factory - a step towards the next generation of manufacturing. In *Manufacturing Systems and Technologies for the New Frontier*, pages 115–118. 2008.

[36] P. Lukowicz, H. Junker, M. Stäger, T. von Büren, and G. Tröster. Wearnet: A distributed multi-sensor system for context aware wearables. In *Proceedings of the 2002 ACM Conference on Ubiquitous Computing*, UbiComp '02, pages 361–370, 2002.

[37] P. Lukowicz, J. A. Ward, H. Junker, M. Stäger, G. Tröster, A. Atrash, and T. Starner. Recognizing workshop activity using body worn microphones and accelerometers. In *Proceedings of the 2nd International Conference on Pervasive Computing*, Pervasive '04, pages 18–32, 2004.

[38] T. Maekawa, Y. Kishino, Y. Sakurai, and T. Suyama. Activity recognition with hand-worn magnetic sensors. *Personal and Ubiquitous Computing*, 17(6):1085–1094, Aug. 2013.

[39] T. Maekawa, Y. Kishino, Y. Yanagisawa, and Y. Sakurai. Recognizing handheld electrical device usage with hand-worn coil of wire. In *Proceedings of the 10th International Conference on Pervasive Computing*, Pervasive '12, pages 234–252, 2012.

[40] T. Maekawa, Y. Kishino, Y. Yanagisawa, and Y. Sakurai. Wristsense: Wrist-worn sensor device with camera for daily activity recognition. In *Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 510–512, March 2012.

[41] T. Maekawa, D. Nakai, K. Ohara, and Y. Namioka. Toward practical factory activity recognition: Unsupervised understanding of repetitive assembly work in a factory. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, page 1088–1099, 2016.

[42] T. Maekawa, K. Ohara, Y. Zhang, M. Fukutomi, S. Matsumoto, K. Matsumura, H. Shidara, S. J. Yamazaki, R. Fujisawa, K. Ide, N. Matsumura, K. Yamazaki, S. Koike, T. Miyatake, K. D. Kimura, H. Ogawa, S. Takahashi, and K. Yoda. Deep learning-assisted comparative analysis of animal trajectories with DeepHL. *Nature Communications*, 11(1):1–15, 2020.

[43] T. Maekawa and S. Watanabe. Unsupervised activity recognition with user's physical characteristics data. In *Proceedings of the 2011 ACM International Symposium on Wearable Computers*, ISWC '11, pages 89–96, June 2011.

[44] T. Maekawa, Y. Yanagisawa, Y. Kishino, K. Ishiguro, K. Kamei, Y. Sakurai, and T. Okadome. Object-based activity recognition with heterogeneous sensors on wrist. In *Proceedings of the 8th International Conference on Pervasive Computing*, Pervasive '10, pages 246–264, 2010.

[45] M. J. Mathie, A. C. F. Coster, N. H. Lovell, B. G. Celler, S. R. Lord, and A. Tiedemann. A pilot study of long-term monitoring of human movements in the home using accelerometry. *Journal of Telemedicine and Telecare*, 10(3):144–151, 2004. PMID: 15165439.

[46] F. E. Meyers and J. R. Stewart. *Motion and Time Study for Lean Manufacturing*. Pearson College Division, 2002.

[47] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, volume 2 of *NIPS '13*, page 3111–3119. 2013.

[48] D. Minnen, T. Starner, I. Essa, and C. Isbell. Discovering characteristic actions from on-body sensor data. In *Proceedings of the 2006 ACM International Symposium on Wearable Computers*, ISWC '06, pages 11–18, Oct 2006.

[49] S. Mirjafari, K. Masaba, T. Grover, W. Wang, P. Audia, A. T. Campbell, N. V. Chawla, V. D. Swain, M. D. Choudhury, A. K. Dey, S. K. D'Mello, G. Gao, J. M. Gregg, K. Jagannath, K. Jiang, S. Lin, Q. Liu, G. Mark, G. J. Martinez, S. M.

Mattingly, E. Moskal, R. Mulukutla, S. Nepal, K. Nies, M. D. Reddy, P. Robles-Granda, K. Saha, A. Sirigiri, and A. Striegel. Differentiating higher and lower job performers in the workplace using mobile sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(2), June 2019.

[50] F. J. O. n. Morales and D. Roggen. Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, ISWC '16, page 92–99, 2016.

[51] F. Moya Rueda, R. Grzeszick, G. A. Fink, S. Feldhorst, and M. Ten Hompel. Convolutional neural networks for human activity recognition using body-worn sensors. *Informatics*, 5(2), 2018.

[52] M. Müller. Dynamic time warping. *Information Retrieval for Music and Motion*, pages 69–84, 2007.

[53] V. S. Murahari and T. Plötz. On attention models for human activity recognition. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, ISWC '18, page 100–103, 2018.

[54] K. Murphy and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Sequential Monte Carlo Methods in Practice*, pages 499–515. Springer, 2001.

[55] A. Ng et al. Sparse autoencoder. *CS294A Lecture Notes*, 72(2011):1–19, 2011.

[56] S. Omidi, F. Schreiber, and A. Masoudi-Nejad. Moda: an efficient algorithm for network motif discovery in biological networks. *Genes & Genetic Systems*, 84(5):385–395, 2009.

[57] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '12, pages 2847–2854, June 2012.

[58] T. Plötz, P. Moynihan, C. Pham, and P. Olivier. Activity recognition and healthier food preparation. In *Activity Recognition in Pervasive Intelligent Environments*, pages 313–329. Springer, 2011.

[59] A. Radziwon, A. Bilberg, M. Bogers, and E. S. Madsen. The smart factory: Exploring adaptive and flexible manufacturing solutions. *Procedia Engineering*, 69:1184–1190, 2014.

[60] J. Ranjan and K. Whitehouse. Object hallmarks: Identifying object users using wearable wrist sensors. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, page 51–61, 2015.

[61] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. Activity recognition from accelerometer data. In *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence*, volume 3 of *IAAI'05*, page 1541–1546, 2005.

[62] C. Reining, F. Niemann, F. Moya Rueda, G. A. Fink, and M. ten Hompel. Human activity recognition for production and logistics—a systematic literature review. *Information*, 10(8), 2019.

[63] C. Reining, M. Schlangen, L. Hissmann, M. ten Hompel, F. Moya, and G. A. Fink. Attribute representation for human activity recognition of manual order picking activities. In *Proceedings of the 5th International Workshop on Sensor-Based Activity Recognition and Interaction*, iWOAR '18, 2018.

[64] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, February 1978.

[65] A. R. Sanabria and J. Ye. Unsupervised domain adaptation for activity recognition across heterogeneous datasets. *Pervasive and Mobile Computing*, 64:101147, 2020.

[66] B. Sherafat, C. R. Ahn, R. Akhavian, A. H. Behzadan, M. Golparvar-Fard, H. Kim, Y.-C. Lee, A. Rashidi, and E. R. Azar. Automated methods for activity recognition of construction workers and equipment: State-of-the-art review. *Journal of Construction Engineering and Management*, 146(6):03120002, 2020.

[67] T. Stiefmeier, G. Ogris, H. Junker, P. Lukowicz, and G. Troster. Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a

maintenance scenario. In *Proceedings of the 2006 ACM International Symposium on Wearable Computers*, ISWC '06, pages 97–104, Oct 2006.

[68] T. Stiefmeier, D. Roggen, and G. Troster. Fusion of string-matched templates for continuous activity recognition. In *Proceedings of the 2007 ACM International Symposium on Wearable Computers*, ISWC '07, pages 41–44, Oct 2007.

[69] N. L. Stokey. Human capital, product quality, and growth. *The Quarterly Journal of Economics*, 106(2):587–616, 1991.

[70] Y. Tanaka, K. Iwamoto, and K. Uehara. Discovery of time-series motif from multi-dimensional data based on MDL principle. *Machine Learning*, 58(2-3):269–300, 2005.

[71] W. Tao, Z.-H. Lai, M. C. Leu, and Z. Yin. Worker activity recognition in smart manufacturing using IMU and sEMG signals with convolutional neural networks. *Procedia Manufacturing*, 26:1159–1166, 2018.

[72] E. Thomaz, I. Essa, and G. D. Abowd. A practical approach for recognizing eating moments with wrist-mounted inertial sensing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, page 1029–1040, 2015.

[73] J. A. Ward, P. Lukowicz, and G. Tröster. Gesture spotting using wrist worn microphone and 3-axis accelerometer. In *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages and Technologies*, sOc-EUSAI '05, page 99–104, 2005.

[74] J. A. Ward, P. Lukowicz, G. Troster, and T. E. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1553–1567, Oct. 2006.

[75] Q. Xia, J. Korpela, Y. Namioka, and T. Maekawa. Robust unsupervised factory activity recognition with body-worn accelerometer using temporal structure of multiple sensor data motifs. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3), Sept. 2020.

[76] Q. Xia, A. Wada, J. Korpela, T. Maekawa, and Y. Namioka. Unsupervised factory activity recognition with wearable sensors using process instruction information. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(2), June 2019.

[77] Q. Xia, A. Wada, T. Yoshii, Y. Namioka, and Maekawa.Takuya. A supporting technique for comparative analysis of factory work by skilled and unskilled workers using neural network with attention mechanism. Dicomo '21, July 2021 (to appear).

[78] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, volume 48 of *ICML'16*, page 478–487, 2016.

[79] D. Yankov, E. Keogh, J. Medina, B. Chiu, and V. Zordan. Detecting time series motifs under uniform scaling. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, page 844–853, 2007.

[80] J. Ye, S. Dobson, and F. Zambonelli. XLearn: Learning activity labels across heterogeneous datasets. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(2):1–28, 2020.

[81] K. Yordanova. Extracting planning operators from instructional texts for behaviour interpretation. In *Proceedings of the 2018 Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 215–228, 2018.

[82] M. Zeng, H. Gao, T. Yu, O. J. Mengshoel, H. Langseth, I. Lane, and X. Liu. Understanding and improving recurrent networks for human activity recognition by continuous attention. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, ISWC '18, page 56–63, 2018.