Preliminary Investigation of Unsupervised Factory Activity Recognition with Wearable Sensors via Temporal Structure of Multiple Motifs

Qingxin Xia[1] Joseph Korpela[1] Yasuo Namioka[2] Takuya Maekawa[1]

Abstract: This paper presents a robust unsupervised factory activity recognition technique using accelerometer data collected from wearable sensors. In line-production systems, each worker repeatedly performs an ordered operation series predefined by factories. Because the working process frequently changes, it's impractical to update labeled training data for every workers over time, unsupervised learning technique has been attracting attention. However, prior studies applying unsupervised factory activity recognition methods are vulnerable to outlying activities performed by the workers which usually occur in real manufacturing. In this study, we propose a robust unsupervised learning technique that makes use of two types of sensor data motifs to track the starting time of each operation in every iteration of work periods. A period motif only occurs once in each work period that is used to roughly detect the duration of work periods. An action motif occurs many times in each work period, corresponding to some basic actions in the period. A temporal structure is then constructed based on the temporal distances among motifs in the first period, which is used to improve motif tracking in the following periods as well as roughly detect the location of outliers. We run particle filters to track the starting time of operations and select a best particle series based on the extracted motifs. We evaluate the proposed method using sensor data collected from workers in actual factories and achieved state-of-the-art performance.

1. Introduction

This study aims to recognize working activities done by a worker in factory settings. One common feature found in many factory works is that workers need to repeat a predefined sequence of work processes in a line-production system when a product passes by, such as removing film from a board and then screwing parts onto the board. It is important to recognize a working class performed by a target person during a given period of time, such as detecting an activity class corresponds to a screwing operation in a work process in order to maintain the work efficiency. Having detected time location and timing of the operations, we can then support many useful applications, such as monitoring the status of workers and detecting outlier activities. Because large numbers of workers are often involved in line production systems, it can be difficult for managers to monitor all of their workers simultaneously; therefore, automated monitoring systems for factory-work activities are in high demand by manufacturers, with several previous studies in the ubicomp community having proposed solutions that are based on supervised activity recognition [1, 2, 3, 4, 7]. However, collecting labeled training data for supervised learning can be prohibitively costly when applied to factory-work activities because training data must be collected for every worker, who often performs a different sequence of operations.

[1] Osaka University, Graduate School of Information Science and Technology, Suita, Osaka, 5650871, Japan
[2] Toshiba Corporation, Corporate Manufacturing Engineering Center, Yokohama, Kanagawa, 2350017, Japan

Figure 1 shows an example acceleration data collected from a smartwatch worn by a working labor. In this case, a period, which represents a complete work process, contains 6 operations, with consistent sequence predefined by managers. However, in the second period, there is an undefined data segment called "Outlier" which usually happens in real factory works. Existing studies on unsupervised learning techniques for activity recognition is difficult to deal with outliers. For example, the latest research by Xia et al. [14] uses a single characteristic sensor data segment motif (e.g., m_{11}) to recognize the starting time of operations. However, it is difficult to correctly estimate the operations using a single motif when outliers occur. In our proposed method, we extract multiple period motifs (three in this example) and consider the temporal structure of the period motifs (e.g., temporal distance between motifs) to narrow down the location of outliers and use the occurrences of action motifs, which correspond to frequent actions such as screwing, to calculate the similarity between identified operations in different periods.

The contributions of this research are listed as follows.

(1) We propose a robust unsupervised learning approach that applies two types of motifs (period motif and action motif) and the temporal structure to track the starting time of operations.

(2) Our method shows significant improvement in activity recognition accuracy with data collected in real factory scenario, especially when outlying activities exist, comparing with previous studies using unsupervised learning methods [14].

(3)We measure the similarity between two operations us-

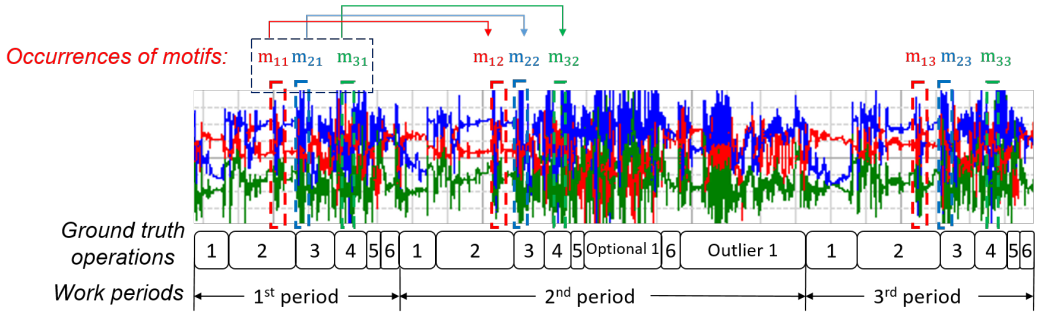


Fig. 1: Example acceleration data collected from a worker. The red, green and blue lines show the x-, y- and z-axis data, respectively. The rectangles below show the fround truth labels of the operations. The duration of each operation period is about 59s. Each worker period consists of six operations, 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.

ing information about the occurrences of action motifs, e.g., the number of occurrences, instead of using time consuming dynamic time warping.

2. Related Work

Techniques for recognizing and supporting factory work using sensor technologies have been actively studied [1, 2, 3, 4, 7] due to the recent growing interest in Industry 4.0 and smart manufacturing [5, 9, 11]. For example, Koskimäki et al. [8] analyzed sensor data from a wrist-worn inertial sensor devices to ensure that all necessary operations were performed. In addition, Stiefmeier et al. [13] analyzed assembly work on automobiles using inertial sensors attached to several locations on workers' bodies, such as the upper and lower arms. They classified sensor data segments by using the computed distance between the collected segments and sensor data templates prepared in advance. Stiefmeier et al. [12] also analyzed work on bicycle repair by recognizing motion sensors and ultrasonic hand tracking employing HMMs. All of these factory activity recognition studies relied on supervised machine learning and so required the creation of labeled training data.

However, supervised machine learning technologies rely on training data in advance. Several studies aim to balance the trade-off between the recognition accuracy and the costs of labeling data by applying unsupervised machine learning. For example, Maekawa et al. [10] measured the duration of each work period on a production line system in an unsupervised manner. This method tracked a motif that appears only once in each work period using a particle filter to estimate the duration of the work periods. Xia et al. [14] estimated the starting and ending times of each operation which makes up of work periods based on the tracked motif with the help of information derived from process instructions. However, as mentioned in the introduction section, these methods are vulnerable to deal with outlier activities, which usually exist in factory work.

3. Unsupervised Factory Recognition with Structure of Motifs

Figure 2 shows an overview of the proposed method. It mainly consists of three phases: (i) discovering motifs, (ii) building motif temporal structure, and (iii) tracking period motifs as well as starting time of operations. In the motif discovery phase, we first preprocess sensor data and then select period and action motifs among motif candidates that randomly generated in an initial segment of the sensor data (i.e., the first t minutes). We build a temporal structure to describe the relationship of each pair of period motifs in the first work period and apply it in the following work periods. Finally, we generate first particle filter to track period motifs and combine it with action motifs to improve the second particle filter for tracking starting time of operations. The starting time of operations is then recognized based on the best particle series of the second particle filter.

3.1 Preliminaries

Before applying this method, we need to build operation flow models to specify the order and standard duration of every operations for each work. This work model is a tree structure, each node corresponds to an operation including with it's standard working duration, the edge represents the every possible sequence of operations. As the figure 3 shows, the first node represents the first operation (e.g., Op.1 is a packing operation) in this work with standard duration of 3.5 seconds, the next Op.2 (e.g., a screwing operation) follows after Op.1 and connects to Op.3 as well as Op.4. Optional operation Op.4 is represented as a branch, which has the same possibility with Op.3 appearing after Op.2.

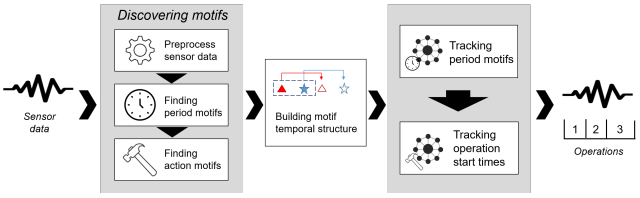


Fig. 2: Overview of the proposed method.

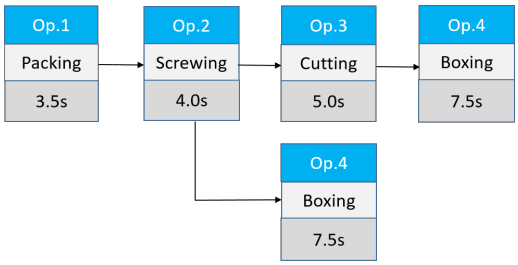


Fig. 3: Example of operation flow model for a work.

3.2 Discovering Motifs

3.2.1 Preprocessing Sensor Data

In order to reduce computation costs, we first symbolize the input acceleration data based on previous factory activity recognition methods [10, 14]. 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 likely starting locations for operations using a Bayesian nonparametric version of HMM called the hierarchical Dirichlet process HMM (HDP-HMM) [6]. 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. Figure 4 shows example output from this segmentation method when applied to acceleration data collected from a factory worker. The overlaid color rectangles show the segments generated by this method. The dash lines corresponds to the ground truth of operations start times for the work. 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.

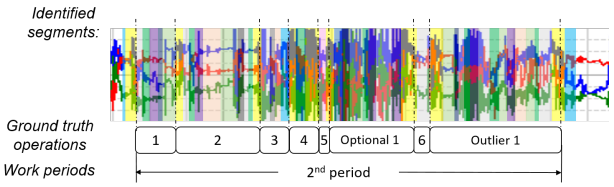


Fig. 4: Example segmentation results for acceleration data when using HDP-HMM to segment the data in an unsupervised manner.

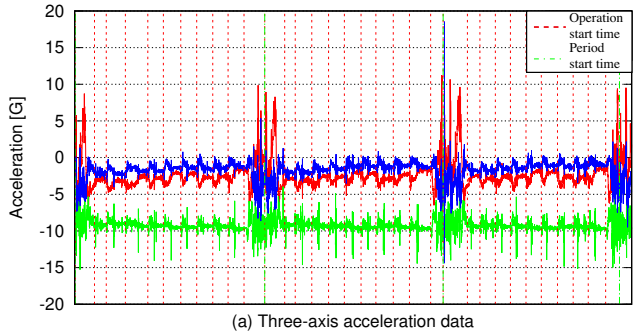
3.2.2 Finding period motifs

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. 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 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 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 5 (b) shows an example of a good candidate for a period motif, with the location of the motif's template indicated by a rectangle background. Meanwhile, Figure 5 (c) shows an example of a bad candidate for a period motif, with the template again indicated by a rectangle background. In the case of Figure 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 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.

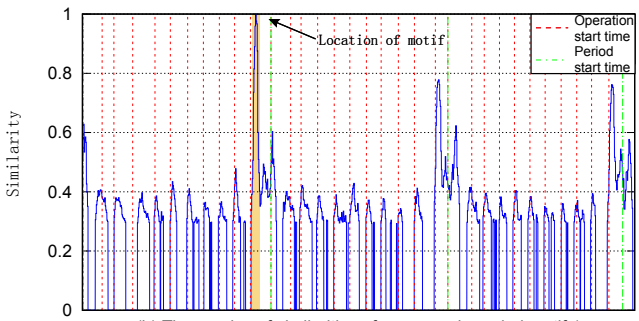
We select the top-k_p best period motifs from amongst the candidates using a combination of two scores: (1) As shown in Figure 5 (b), the similarity values at peaks should be much larger than those at other times in the periods. Therefore, we compute the first score based on the difference between the peak similarity value and similarity values at the other time slices. (2) The occurrence of a period motif in the following work periods should be similar to that of in the first period. Therefore, the second score for candidate motifs is calculated based on the maximum peak values in other period segments.

3.2.3 Finding action motifs

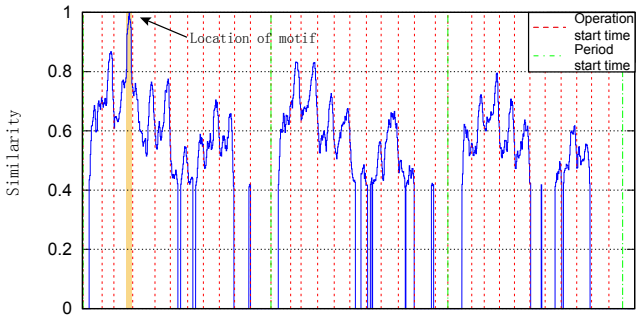
Just as with period motifs, we start our process of selecting a good action motif by first extracting candidate motifs and then computing a similarity time series $\mathbf{S}$ for each candidate using the first t_{dm} minutes of sensor data. Figure 6 shows two examples of such similarity time series, 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 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 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 corresponding actions are similar to the surrounding actions,



(a) Three-axis acceleration data

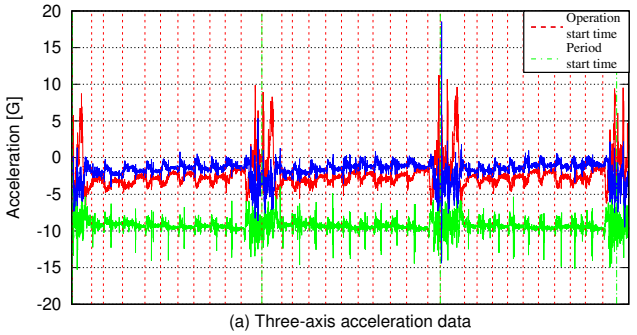


(b) Time-series of similarities of an example period motif 1

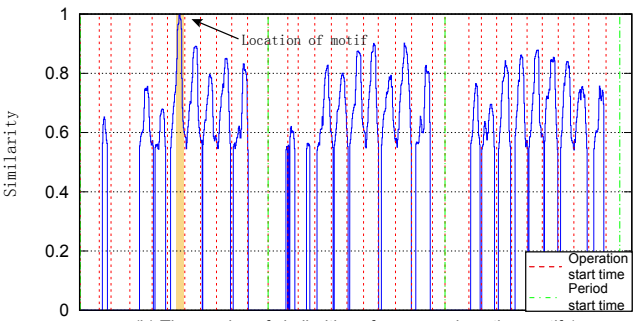


(c) Time-series of similarities of an example period motif 2

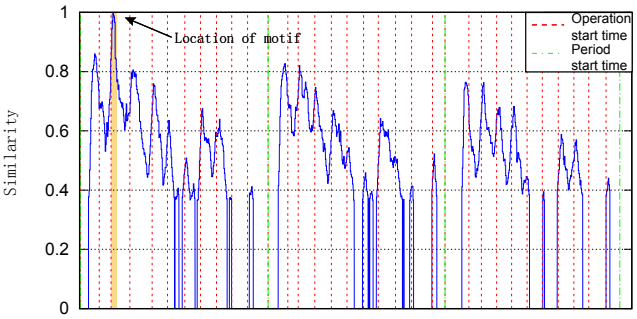
Fig. 5: Comparing the similarity time series for candidate period motifs. (a) The raw three-axis acceleration data. (b) and (c) are the similarity time series for a good and bad candidate for a period motif, respectively.



(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

Fig. 6: Comparing the similarity time series for candidate action motifs. (a) The raw three-axis acceleration data. (b) and (c) are the similarity time series for a good and bad candidate for an action motif, respectively.

which is therefore hard to detect.

We select the best action motif from the amongst the candidates using score based on the average prominence of peaks in the similarity time segment. The peak prominence is the height of the peak's summit above the lowest contour line but containing no higher summit, showing the difference between the action motif and the surrounding data segments.

3.3 Building Temporal Models

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 motifs in the first t_{init} minutes of sensor data. 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.

3.4 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.

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. 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 motif's particle filter. In the resampling procedure, we probabilistically discard particles with low scores. We explain these procedures in detail below.

3.4.1 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.

3.4.2 Sampling

We randomly sample subsequent occurrences for each particle based on $\mathbf{S}$ and the standard lead times from the work model $\mathcal{M}$. 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 by adding a Δt, 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 makes the possible occurrence of the motif only occurs at the location of peaks and therefore formulated as follows.

$$\mathcal{G} = \textsc{Disc}(\text{peak}(\mathbf{S}_{(t(p_i,n)+d_f \cdot \mathcal{M}.min\_lead\_time, t_f)})),$$

where t_f is a time when $\mathbf{S}$ first exceeds th_{pm} after $t(p_i, n) + d_f \cdot \mathcal{M}.min\_lead\_time$ plus a margin, d_f is a hyperparameter that defines the possible minimum lead time ($0 < d_f < 1$), $\mathbf{S}_{(t(p_i,n)+d_f \cdot \mathcal{M}.min\_lead\_time, t_f)}$ is a segment between $t(p_i, n) + d_f \cdot \mathcal{M}.min\_lead\_time$ 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.

3.4.3 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: (1) Similarity score: A sensor data segment corresponding to p_j should be similar to the segment corresponding to the first occurrence of the motif. Therefore, we simply use the similarity value as a score. (2) 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 differences between $\mathcal{T}$ and the temporal distance of target period motifs in other work periods.

3.4.4 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. 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.

3.5 Tracking Operation Start Times

Now, we have the following information: (1) The particle filters for period motifs output timestamps for the occurrences of the peiod motifs. (2) We have timestamps for trend changes $\mathbf{C}$ that are output by HDP-HMM. (3) We have information about the standard duration for each operation which is described in the work model $\mathcal{M}$.

We 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.

3.5.1 Initialization

As shown in Figure 1, 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.

3.5.2 Sampling

For 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 es-

timate 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 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 **??** (b), we can assume that there are some outlier segments between the two occurrences. Therefore, t_{ol} is calculated as follows.

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

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

3.5.3 Weighting

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: (1) Trend-based score: As was mentioned in the introduction section, operation start times usually exist close to trend changes in the sensor data. Therefore, this score is computed based on the temporal distance between the timestamp of a particle and the timestamp of the closest trend change that discovered by HDP-HMM. (2) Period-motif score: The temporal distance between the start time of the $n+1$-th operation and an adjacent period motif should be consistent throughout work periods. Therefore, we calculate the consistency of the temporal distances within the period of interest. (3) Action-motif score: A similarity time series is given for each action motif, which represents the occurrences of that action motif. The similarity segments corresponding to a given operation should be similar with each other across work periods. Therefore, we calculate the consistency of each operations segments within the period of interest. To reduce the computation costs related to the similarity calculation, we extract a feature vector consisting of simple statistical features such as: (i) standard deviation, (ii) variance, (iii) kurtosis, (iv) skewness, (v) #peaks divided by the duration of the segment, (vi) the largest peak value, (vii) average of peak values, and (viii) relative location of the largest peak within the segment. The result of Euclidean distance is then used as the score.

3.5.4 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 1, 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 of operations finally selected based on the particle that has the highest weight at the end of the period.

4. Evaluation

We evaluated the proposed method using 6 datasets collected from different workers in real factory. Each worker wears a Sony SmartWatch3 SWR50 on his right wrist to record acceleration data with an approximate 60Hz sample rate. The ground truth data was collected using video cameras. The data processed offline in our laboratory after data collection. Table 1 shows an overview of all the datasets. The number of operation in each work period is shown in the "# operations" row, the number of optional operations is added in parentheses. The " standard duration" row describes the standard duration of a period, the parentheses shows another standard duration when optional operations occur. The number of periods over a whole work is shown in the "# periods" row. The "data duration" row means the overall duration of the observed sensor data for each work. The "outlier duration" row describes the overall duration of outlier activities for each work.

Our method measures the starting time of each operation in every period. The duration of an operation starts from its own starting time and ends at the starting times of the next operation, all outlier activities are labeled as "outlier" operations. We evaluate every sensor data point by measuring if it belongs to the operation. Therefore, we can calculate the macro-averaged F-measure for the classification result of every sensor data points over each dataset. We designed 6 comparing methods to evaluate the effectiveness of the proposed method:

- **Proposed**: The proposed method.

Table 1: Overview of our dataset

	work A	B	C	D	E	F
# operations	8	6 (1)	7 (1)	7	11	11
standard duration [s]	130	59 (87)	53 (57)	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
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

- **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.
- **W/o-Period**: The proposed method without the use of the period-motif scores.
- **W/o-Action**: The proposed method without the use of the action-motif scores.
- **Xia2019**: This is the proposed method from Xia2019.

4.1 Recognition Accuracy

Figure 7 shows the classification accuracy for each method. Overall, the Proposed method achieved the highest average F-measure over all six works. In addition, the accuracies of Xia2019 in many works are lower than the Proposed, especially when the work contains outliers (e.g., work B, C, D and F). Comparing with Xia2019, this result confirms the robustness of the proposed method when used on real-world factory work sensor data with outliers exist.

4.2 Effect of scores

Trend-based score: The f-measure of the Proposed outperforms the W/o-Trend method in work D, E and F, many of the estimated starting times of operations in the W/o-Trend method shifted from the corresponding operations in the Proposed method. Since the ground truth starting times of operations are very close to the trend changes generated by HDP-HMM, we can use trend-based score to reduce the appearances of shifted tracking results.

Period-motif score: Figure 8 shows an example of W/o-Period in work F, which does not employ the period-motif score. The accuracy of this work is poor, where the number of outliers is large, indicating that the temporal relationship between the occurrences of periods motifs and operation start times was important. Without period motifs, the W/o-Period method can not recognize the starting time of outlier in this example, resulting in a low recognition accuracy in the first period. In addition, as can be seen in the result of Only-Base method, which employs only a base period motif, the accuracy is the lowest among all comparing methods. This result means tracking multiple period motifs is an important aspect of Proposed, as it is difficult to recognize the fine-grained structure of the work only with a single period motif.

Action-motif score: The action-motif score is another important score in this study, this score plays a similar role in sensor-based score in Xia2019 (refers to [14]), but requires less computation costs. As can be seen in the figure 9, the estimated sensor data segment of the same operations in the 3-rd and 4-th periods are different. Because the working duration of operations are various in this case (e.g., the duration of the first operation in the third period is about two times longer than the corresponding operation in the forth period), the period-motif score that relies on the consistency distances between period motifs becomes unstable, action motif score is able to select good tracking results with high data similarity of same operations in different periods.

5. Conclusion

In this study, we proposed a robust unsupervised factory activity recognition method to recognize the starting times of operations by using multiple motifs in conjunctions with motif temporal structures, which achieves state-of-the-art performance even when outliers exist.

Acknowledgment

This work is partially supported by JST CREST JPMJCR15E2, JSPS KAKENHI Grant Number JP16H06539 and JP17H04679. The first author is supported by China Scholarship Council.

References

[1] Mario Aehnelt and Sebastian Bader. Information assistance for smart assembly stations. In *the 7th International Conference on Agents and Artificial Intelligence (ICAART 2015)*, volume 2, pages 143–150, 2015.

[2] Mario Aehnelt and Bodo Urban. The knowledge gap: providing situation-aware information assistance on the shop floor. In *HCI in Business*, pages 232–243. 2015.

[3] Mario Aehnelt and Karoline Wegner. Learn but work!: towards self-directed learning at mobile assembly workplaces. In *the 15th International Conference on Knowledge Technologies and Data-driven Business*, page 17, 2015.

[4] Sebastian Bader, Frank Krüger, and Thomas Kirste. Computational causal behaviour models for assisted manufacturing. In *the 2nd international Workshop on Sensor-based Activity Recognition and Interaction*, page 14, 2015.

[5] Martin Bauer, Lamine Jendoubi, and Oliver

Fig. 7: F-measure of 6 comparing methods for each work.

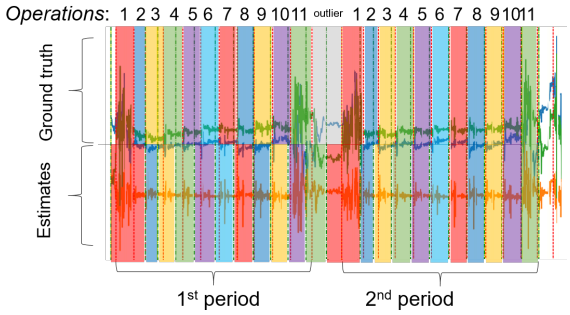


Fig. 8: Example output of estimated starting time of operations of work F by using W/o-Period method.

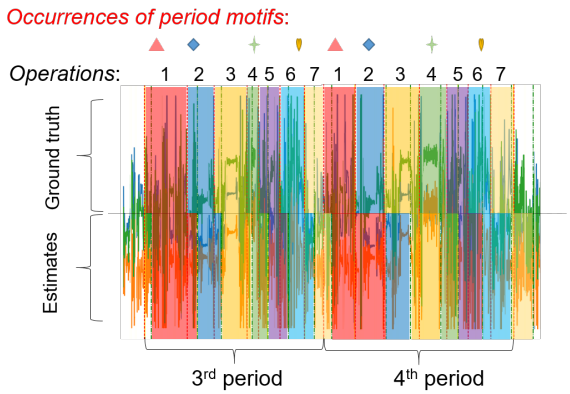


Fig. 9: Example output of estimated starting time of operations of work C by using W/o-Action method.

Siemoneit. Smart factory–mobile computing in production environments. In *the MobiSys 2004 Workshop on Applications of Mobile Embedded Systems (WAMES 2004)*, 2004.

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

[7] Oliver Korn, Albrecht Schmidt, and Thomas Hörz. The potentials of in-situ-projection for augmented workplaces in production: a study with impaired persons. In *CHI'13 Extended Abstracts*, pages 979–984, 2013.

[8] Heli Koskimäki, Ville Huikari, Pekka Siirtola, Perttu Laurinen, and Juha Röning. Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines. In *17th Mediterranean Conference on Control and Automation (MED 2009)*, pages 401–405, 2009.

[9] Dominik Lucke, Carmen Constantinescu, and Engelbert Westkämper. Smart factory-a step towards the next generation of manufacturing. In *Manufacturing systems and technologies for the new frontier*, pages 115–118. Springer, 2008.

[10] Takuya Maekawa, Daisuke Nakai, Kazuya Ohara, and Yasuo Namioka. Toward practical factory activity recognition: unsupervised understanding of repetitive assembly work in a factory. In *UbiComp 2016*, pages 1088–1099, 2016.

[11] Agnieszka Radziwon, Arne Bilberg, Marcel Bogers, and Erik Skov Madsen. The smart factory: Exploring adaptive and flexible manufacturing solutions. *Procedia Engineering*, 69:1184–1190, 2014.

[12] Thomas Stiefmeier, Georg Ogris, Holger Junker, Paul Lukowicz, and Gerhard Tröster. Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. In *10th IEEE International Symposium on Wearable Computers (ISWC 2006)*, pages 97–104, 2006.

[13] Thomas Stiefmeier, Daniel Roggen, and Gerhard Tröster. Fusion of string-matched templates for continuous activity recognition. In *11th IEEE International Symposium on Wearable Computers (ISWC 2007)*, pages 41–44, 2007.

[14] Qingxin Xia, Atsushi Wada, Joseph Korpela, Takuya Maekawa, and Yasuo 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):60, 2019.