Master's Thesis

Title

Anomaly detection using incremental clustering and correlation coefficients in industrial wireless sensor networks

Supervisor
Professor Naoki Wakamiya

Author
Ryosuke Kanajiri

February 5th, 2020

Graduate School of Information Science and Technology
Osaka University

Master's Thesis, 2020

Anomaly detection using incremental clustering and correlation coefficients in industrial wireless sensor networks

Ryosuke Kanajiri

Abstract

Industrial wireless sensor networks (IWSNs) are used for monitoring, controlling, and other various industrial tasks including factory automation and system monitoring. Since communication failure in IWSNs such as continuous communication interruption can cause a huge economic loss, an effective framework detect, identify, and deal with unusual communication situations in real-time. In our research group, a method has been proposed for the detection of unusual events in a radio wave propagation environment using RSSI (Received Signal Strength Indicator). It detects an unusual event using the difference between the current mean RSSI (i.e., the RSSI that is time-averaged in a time window) and the exponential moving average of RSSI in the past. However, because taking a time-average can hinder important information such as temporal fluctuations, the mean RSSI is not enough to detect unusual situations. For this reason, it is of practical and urgent importance to develop a methodology for detecting unusual radio propagation patterns using real-time RSSI data.

In this thesis, we propose a real-time anomaly detection method to detect unusual situations in the radio propagation environment using RSSI. To perform real-time detection, we first propose an incremental clustering method to aggregate the incoming sequence of RSSI data. Based on the obtained clusters, we then define the following two anomaly measures for individual links: the anomaly measure for the current RSSI histogram, and the one for the recent transition. Specifically, the first measure is based on the number of data in the cluster while the latter one is based on the transition probability between clusters. In order to further enhance the quality of anomaly detection, we introduce the third metric based on link-correlation. Based on these anomaly scores, we determine whether the current situation is usual or not. We verify that the proposed method can detect RSSI time series data as an anomaly when artificially blocking WSN nodes with obstacles, and show the analysis results of events that can be detected with the proposed anomaly scores.

Keywords

Wireless sensor network, anomaly detection, RSSI, incremental clustering

Contents

List of Figures

List of Tables

1 Introduction

Industrial wireless sensor networks (Industrial WSNs) [1], [2] are used for monitoring and controlling various industrial tasks including factory automation [3] and system monitoring [4]. A primary reason for expansion of WSNs in the industrial context is that WSNs have several advantages over traditional wired sensor networks [5]; a typical WSN requires no cables and, therefore, allows its low-cost and easy implementation as well as repairing [6]. In order to enjoy this superiority of Industrial WSNs, we need to guarantee the reliability of wireless communication on which we build our WSNs. However, the occurrence of communication failures is inevitable due to the harshness of industrial indoor environments. For example, steel, metals and rotating machinery often cause short-term or long-term shadow fading. Such communication interferences can cause a sudden drop in the received signal strength indicator (RSSI), and the drop can be as much as 40 dB [7]. For another example, vehicles, such as trucks and forklifts parking in front of wireless nodes, may temporarily eliminate the communication completely [7]. Because such communication failures are often not predictable, there is an urgent need for the development of an effective framework to automatically detect failures in node-to-node wireless communications.

Toward the development of such a framework, in this thesis we make use of both usual and unusual situations to detect communication failures. If unusual situations occur, then its effect should appear in the RSSI time series data. For this reason, we propose that we use RSSI for anomaly detection. We remark that, however, the temporal fluctuations in the RSSI are caused not only by the communication failures but also other events considered to be normal. Examples of such normal events include the movement of people or vehicles crossing between nodes or the reflection of radio waves by metal objects. For this reason, in this thesis, we focus on detecting unusual situations using RSSI distributions. The reason we use RSSI distributions instead of RSSI time series is that obtained RSSI changes largely.

In order to achieve the aforementioned objective, in this thesis, we propose an anomaly detection methodology based on multiple anomaly scores. Namely, the proposed anomaly scores are differences in RSSI distribution, differences in transition probabilities between RSSI distributions, and differences in the correlation of RSSI time series data between links. Using these three anomaly scores, we propose a real-time anomaly detection method to detect communication failures. The proposed method uses incremental clustering and correlation coefficients.

We use the incremental clustering to detect differences in RSSI distribution and differences transition probabilities between RSSI distributions. The incremental clustering is a method for grouping data in real-time. We devise an incremental clustering method that performs two-stage processing of data assignment and updating of existing clusters for real-time clustering. To determine the best clustering results in multiple-choice update results, we use a clustering metric. We also use correlation coefficients of RSSI Subsequence time series data between links to detect

differences in the correlation of RSSI time series data between links.

The rest of this thesis is organized as follows. In Section 2, we explain proposed anomaly detection method. In Section 3, we evaluate the proposed method. In Section 4, we discuss the evaluated results. Finally, Section 5 concludes this thesis and gives the future work.

2 Anomaly detection using incremental clustering and correlation coefficients

In this section, we first explain the proposed anomaly detection method in Section 2.1. Next, we explain the incremental clustering which is used the proposed anomaly detection method in Section 2.2. In Section 2.3, we explain anomaly scores which are calculated by clustering results and correlation coefficients. Finally, we explain the method of reducing the amount of stored data in Section 2.4.

2.1 Overview of the proposed anomaly detection method

The proposed anomaly detection method use three anomaly scores to detect unusual situations. The three anomaly scores are an anomaly score based on cluster size, an anomaly score based on transition probability, and an anomaly score based on correlation coefficient. The anomaly score based on cluster size is calculated to find differences in RSSI distributions. The anomaly score based on transition probability is calculated to find differences transition probabilities between RSSI distributions. The anomaly score based on correlation coefficient is calculated to find differences in the correlation of RSSI time series data between links. The three anomaly scores are calculated using RSSIs which are obtained sequentially for each link.

The jumping window algorithm is used to obtain the time series subsequence of RSSI per link. The jumping window algorithm requires a parameter of a window size w. For example, let $[-55, -56, -56, -57]$ be a RSSI time series data, let $w = 2$. As a result of using the jumping window algorithm, we obtain the time series subsequence $[-55, -56]$ and $[-56, -57]$.

RSSI distributions per link are created using the time series subsequence of RSSI per link. The distributions means the normalized histogram. The making distribution algorithm requires two parameters, a bin size b_s and the domain of definition of input data. For example, let $[-55, -56, -55, -54, -55]$ be a RSSI time series subsequence data, let $b = 1$, let the domain of definition of input data be the interval $[-57, -53)$. As a result of using the making distribution algorithm, we obtain a distribution $[0.0, 0.2, 0.6, 0.2]$. The first element of the distribution means that there are no data that satisfy $\{x| -57 \leq x < -56\}$. The second element of the distribution means that there are 20% data that satisfy $\{x| -56 \leq x < -55\}$. The third element of the distribution means that there are 60% data that satisfy $\{x| -55 \leq x < -54\}$. The fourth element of the distribution means that there are 20% data that satisfy $\{x| -54 \leq x < -53\}$.

To calculate the anomaly score based on cluster size and the anomaly score based on transition probability, incremental clustering with RSSI distributions for each link is used. The incremental clustering independently performs per link. We explain the incremental clustering in Section 2.2.

To calculate the anomaly score based on correlation coefficient, correlation coefficients between the time series subsequence of RSSI per link are used. We explain the detail of calculating anomaly

scores in Section 2.3.

2.2 Incremental clustering

Clustering is the process of grouping a set of objects into clusters so that objects within a cluster are similar to each other but are dissimilar to objects in other clusters [8][9]. The similarity of data is determined by distance function. As typical clustering methods, K-means [8] and K-medoids [9] are commonly used for data mining. These clustering methods uses all target data. On the other hand, incremental clustering is the method to update clusters whenever new data is obtained for time series data obtained sequentially [10]. The advantage of incremental clustering is that the computational complexity of incremental clustering is smaller than non-incremental clustering such as K-means or K-medoids.

We propose an incremental clustering based on K-medoids. In the proposed incremental clustering, each cluster has a medoid which is a representative point. The medoid of cluster i is selected by the following condition

$$\arg\min_{x \in C_i} \sum_{y \in (C_i - x)} d(x, y), \tag{1}$$

where C_i is cluster i, d is an arbitrary distance function, $d(x, y)$ is a distance between x and y. See Section 2.2.1 for the distance function used in the incremental clustering.

Incremental clustering can perform by the following process that new data is assigned to a cluster i which is belonged a medoid closest to the data and the medoid of the cluster i updates with Equation 1. However, this approach must determine the number of clusters in advance, thus it cannot deal with changes in data trends. Therefore, we modify the process so that the new data is assigned in the existing cluster or a new cluster which consists of only the new data. This process shows in Figure 1. In Figure 1, filled red circle is new data, filled blue circles are data obtained in the past, oval blue frames are cluster boundaries. In the case of A, new data is assigned in the existing cluster. In the case of B, new data is assigned in a new cluster. In addition, in order to flexibly respond to changes in data trends, we use the following process. When new data is assigned to the existing cluster i, the cluster i and the cluster j which is closest to the cluster i are integrated into one cluster, or the cluster i is split into two clusters, or the cluster i is not updated. This process shows in Figure 2. In Figure 2, filled red circle is new data, filled blue circles are data obtained in the past, oval blue frames are cluster boundaries. In the case of C, two clusters integrate into a cluster. In the case of D, a cluster split into two clusters. In the case of E, clusters do not update.

The reason that the above process is not performed when a new cluster is created is that the process of integrating the clusters and the process of being assigned the data to the existing cluster have the same result, and the process of dividing the cluster cannot perform because the cluster consists of one data. Figure 3 shows the flowchart of the proposed incremental clustering. In

Figure 3, A, B, C, D, and E mean the cases shown in Figure 1 and 2.

The detail of integrating two clusters is the following process. First, if obtained data is assigned to the existing cluster i, finds the cluster j which is closest to the cluster i (The Distance between the cluster i and the cluster j means the distance between the medoid of the cluster i and the medoid of the cluster j). Next, all data which is assigned to cluster i and all data which is assigned to cluster j are moved into a new cluster k. Finally, the new cluster k determines its own medoid by Equation 1.

The detail of dividing a cluster is the following process. First, if obtained data is assigned to the existing cluster i, finds a data pair which are the maximum distance in cluster i. Next, let these data pair be x and y, creates new two clusters which are the one consisted of data in the cluster i closer to y than x and another consisted of data in the cluster i closer to x than y. Finally, the new two clusters determine its own medoid by Equation 1 respectively.

When new data is obtained, several clustering results can be considered as described above, but one of them must be selected. Thus, we use clustering metrics which represents goodness for clustering result, and determine the best clustering result. See Section 2.2.2 for the clustering metrics used in the incremental clustering.

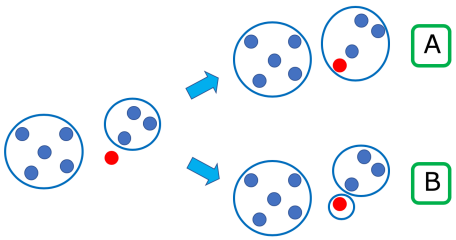

Figure 1: Assigning new data to clusters.

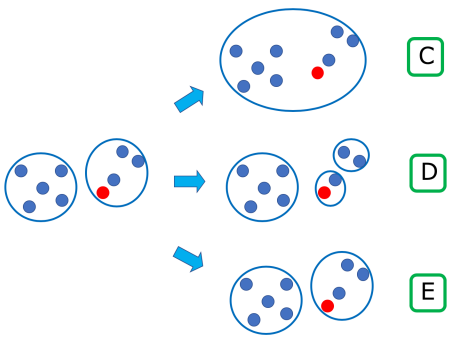

Figure 2: Updating clusters using integration and split.

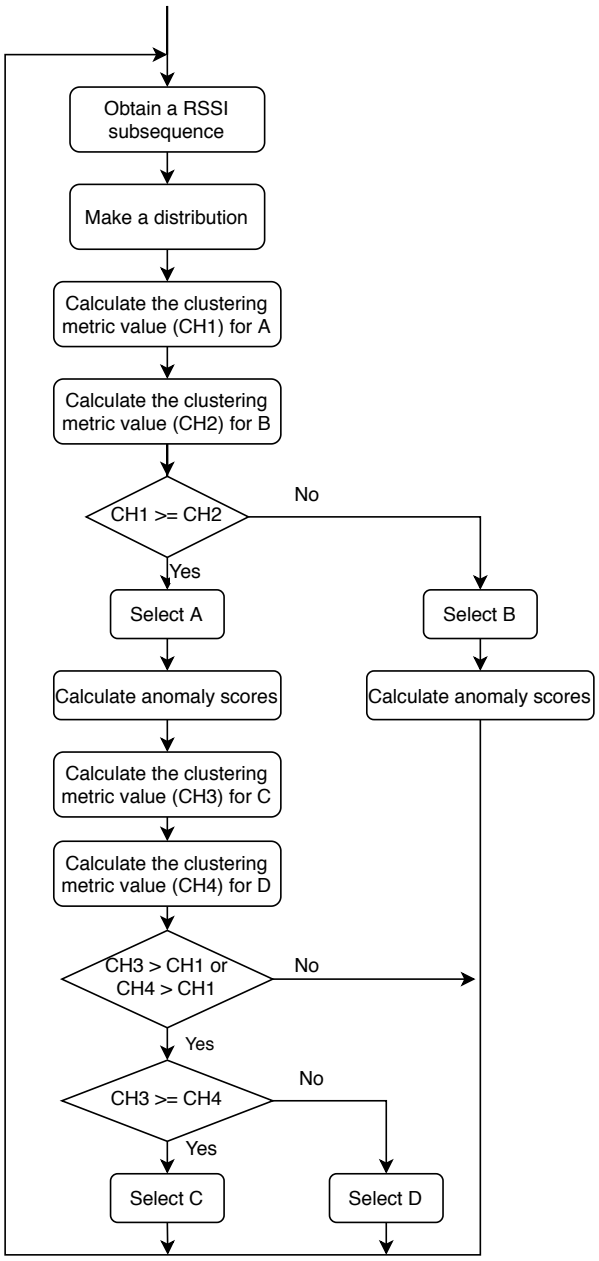

Figure 3: Flowchart of incremental clustering.

2.2.1 Distance functions

It is necessary to select an appropriate distance function depending on the characteristics of the data for clustering. The data for clustering is RSSI distributions, thus a distance function representing dissimilarity between distributions is suitable.

The Kullback-Leibler divergence (KL divergence) [11] is generally used to calculate dissimilarity

between two distributions. The KL divergence is defined as

$$D_{KL}(P\|Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}, \tag{2}$$

where P and Q are the probability distribution, X is the domain of definition of P and Q. However, The KL divergence is an asymmetric and unbounded measure and also cannot calculate it to the distribution which includes zero value. RSSI distributions includes zero value, thus we cannot use the KL divergence.

Another distance function to calculate dissimilarity between two distributions is the Jensen-Shannon divergence (JS divergence) [12]. The JS divergence is a symmetric and bounded measure and also can calculate it to the distribution which includes zero value. The JS divergence is defined as

$$D_{JS}(P\|Q) = \frac{1}{2} D_{KL}(P\|M) + \frac{1}{2} D_{KL}(Q\|M), \tag{3}$$

where P and Q are the probability distribution, X is the domain of definition of P and Q, $M = \frac{1}{2}(P + Q)$. As another characteristic of the JS difference, If the base of the logarithm is 2, the JS divergence satisfies $0 \leq D_{JS}(P\|Q) \leq 1$.

We use the JS divergence as the distance function for incremental clustering because of these characteristics.

2.2.2 Clustering metrics

Various clustering metrics have been proposed [13]. Among them, the Pseudo F (Calinski-Harabasz index) [13] can be applied to K-medoids with a small change. The Pseudo F is defined as

$$\text{Pseudo F} = \frac{\sum_{i=1}^{k} n_i \; d(m_i, m)^2/(k-1)}{\sum_{i=1}^{k} \sum_{x \in C_i} d(x, m_i)^2/(n-k)}, \tag{4}$$

where C_i is cluster i, $d(x, y)$ is a distance between x and y, k is the number of clusters, n_i is the number of data in cluster i, m_i is the medoid of cluster i, m is the mean data of dataset, n is the number of data in dataset.

A clustering metric that is changed the numerator of Equation 4 to the distance between each medoid can use as a metric that represents the goodness of clustering results. This clustering metric is defined as

$$\text{Pseudo F with Mean} = \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} n_i \; d(m_i, m_j)^2/k}{1 + \sum_{i=1}^{k} \sum_{x \in C_i} d(x, m_i)^2}. \tag{5}$$

Furthermore, a clustering metric that is changed the numerator of Equation 4 to the distance between each medoid to the closest medoid can use as a metric that represents the goodness of clustering results. This clustering metric is defined as

$$\text{Pseudo F with Min} = \frac{\sum_{i=1}^{k} n_i d(m_i, m_{\arg\min_j d(m_i, m_j)})^2}{1 + \sum_{i=1}^{k} \sum_{x \in C_i} d(x, m_i)^2}. \tag{6}$$

We determine the clustering metrics to use in incremental clustering by experimental results.

2.3 Anomaly scores

We explain the three anomaly scores in this Section. The anomaly score based on cluster size is explained in Section 2.3.1. The anomaly score based on transition probability is explained in Section 2.3.2. The anomaly score based on correlation coefficient is explained in Section 2.3.3.

2.3.1 Anomaly score based on cluster size

The anomaly score based on cluster size is calculated using the number of elements in the clusters. If the obtained RSSI distribution x is assigned to the cluster i then the anomaly score based on cluster size of x is calculated as

$$a_n = r^{n_i - 1}, \tag{7}$$

where n_i is the number of data in the cluster i, r is the hyperparameter satisfying $0 < r < 1$. The anomaly score a_n satisfies $0 < a_n \leq 1$. The high anomaly score means that the distribution x is rarely obtained in usual. Thus, this anomaly score can find rarely obtained distributions.

Even with a distribution that is rarely obtained, the number of data increases over time, thus the number of data in the cluster to which the distribution is assigned increases. Thus, if this distribution obtained, the anomaly score is a small value. To deal with this, we use the method in Section 2.4 that removes clusters that are not assigned data during a certain period.

2.3.2 Anomaly score based on transition probability

The anomaly score based on transition probability is calculated using transition probabilities between clusters. The transition from cluster i to cluster j mean that the obtained distribution is assigned to cluster j and the previous distribution is assigned to cluster i. The number of all transitions is stored, and the transition probability is calculated using this. If the number of clusters is k and the number of transitions from cluster i to cluster j is M_{ij}, the transition probability from cluster i to cluster j defined as

$$P_{ij} = \frac{M_{ij}}{\sum_{l=1}^{k} M_{il}}.$$

If the obtained RSSI distribution x is assigned to the cluster j and the previous distribution is assigned to the cluster i, the anomaly score based on transition probability of x is calculated as

$$a_t = \frac{\max_l P_{il} - P_{ij}}{\max_l P_{il}} = 1 - \frac{P_{ij}}{\max_l P_{il}}, \tag{8}$$

where P_{ij} is the transition probability from cluster i to cluster j.

When the obtained distribution x is assigned to a new cluster n, the anomaly score of x is 1 because the transition probability to the new cluster is 0. If m is any cluster, the number of the transitions M_{nm} and M_{mn} are initialized 1.

Updating the number of transitions when two clusters are integrated is explained following. Let a and b be two clusters before integration, and c be a cluster after integration. If λ is any cluster other than the cluster c, the number of the transitions $M_{c\lambda}$ is calculated as $M_{c\lambda} = M_{a\lambda} + M_{b\lambda}$. The number of the transitions $M_{\lambda c}$ is calculated as $M_{\lambda c} = M_{\lambda a} + M_{\lambda b}$. The number of the transition M_{cc} is calculated as $M_{cc} = M_{aa} + M_{ab} + M_{ba} + M_{bb}$.

Updating the number of transitions when a cluster is split is explained. Let p and q be two clusters after the split. For each data in p, we can find the cluster to which the next data belongs by stored all transition. Thus, M_{pv} is calculated by the number of transitions from p to any cluster v. M_{qv} is also calculated by the number of transitions from q to any cluster v. Similarly, for each cluster other than p and q, M_{wp} (w is any cluster except p) and M_{wq} (w is any cluster except q) is calculated.

2.3.3 Anomaly score based on correlation coefficient

We then propose our third score for anomaly detection. The third anomaly score is based on correlation coefficients and is calculated using correlation coefficients between RSSI of all links. The correlation coefficients between x and y is calculated as

$$r(x,y) = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y})^2}},$$

where n is the length of x and y, $\overline{x}$ and $\overline{y}$ are the mean of x and y, respectively. Each time an RSSI subsequence is obtained for a link pair, the correlation coefficients is calculated, and the correlation change is calculated from those correlations. If the the current correlation coefficient is r_t and the previous one is r_{t-1}, the change in correlation coefficients is calculated by $rt - r_{t-1}$. To calculate the anomaly score based on correlation coefficient, a normalized histogram X is created using the changes in correlation coefficients obtained in the past. The bin size b_c for creating X require to set in advance. If the change in the obtained correlation is Δr, the anomaly score based on the correlation is calculated by $1 - X(\Delta r)$.

2.4 Reducing the amount of stored data

The proposed anomaly detection method stores RSSI distributions of each link. However, the number of data that can be stored is limited, it is desirable not to store the RSSI distribution when the same distribution obtained in the past is obtained. Therefore, in order to reduce the amount of data to be stored, we explain two procedures which are the procedure for determining whether the distribution is the same or not and the procedure to be performed when the same distribution

is obtained. We also explain the procedure of removing unused clusters to reduce the amount of data to be stored.

Whether the distributions are the same or not is equal to whether the JS divergence is 0 or not 0, so it can be determined by JS divergence. However, calculating the JS divergence between the new obtained distribution and all the distributions obtained in the past requires a large amount of calculation. To deal with this, we use hash and calculate JS divergence with distributions which are the same hash value. We use the mean of the distribution as the hash value. The procedure to be performed when the obtained same distribution is obtained is to assign the obtained distribution to the cluster which is assigned the same distributions, and update the medoid. Furthermore, update clusters using integrating clusters or dividing clusters.

When the same distribution is obtained, the denominator calculation in Equation 4, 5, 6 does not include the JS divergence of the new obtained distribution. Thus, consider the number of the same distribution obtained w for each distribution, and deal with it by multiplying w by JS divergence.

The procedure of removing unused clusters removes clusters which are not assigned data during a certain period T_r. These clusters were not updated during period T_r, thus we consider that the clusters are stored anomalous distributions or distributions which are not occurred because of changing data trends. Therefore, distributions that are contained in these clusters must be detected as anomalous distributions.

Even if clusters that satisfy the above conditions are deleted, the distribution such as be assigned in these clusters is assigned in a new cluster. Therefore, the anomaly score based on cluster size becomes a high value and can be detected as an anomalous distribution. Therefore, removing unused clusters does not affect anomaly detection.

3 Evaluation

First, we explain the RSSI measurement dataset used for evaluating the proposed anomaly detection algorithm. Next, we show the analysis results for determining the clustering metric used for incremental clustering. Finally, we show the results of anomaly detection for the RSSI measurement dataset.

3.1 Datasets

As a dataset used for evaluation of the proposed method, we use RSSI time series data between the nodes which are measured by arranging sensor nodes in indoor environments. Figure 4 shows the environment of RSSI measurements. Figure 5 shows the sensor node used for RSSI measurements. The sensor node specifications conform to IEEE 802.15.4, with a frequency band of 2.4 GHz and a transmission power of 1 mW. The number of sensor nodes is 19. The specific measurement method is as follows. The node 0, the base unit, broadcasts a packet. Other nodes i ($1 \leq i \leq 18$) receive the packet from node 0, wait for $i \times \Delta t$ seconds, and then broadcast the packet to measure the RSSI of the transmitted radio wave. Δt is adjusted so that the measurement cycle is 2 seconds.

We use two dataset to evaluate the proposed method.

Dataset(1) is the dataset of RSSI measurement when a link obstructed by a shield sheet that attenuates radio waves. The measurement period is from 9:55 on December 31th, 2018 to 11:55 on December 31th, 2018. Nodes 3 and node 5 are shielded six times every 10 minutes from 10:00, and nodes 9 and node 16 are shielded six times every 10 minutes from 11:00, and each shielded period is 1 minute. Figure 3.1 shows examples of the RSSI measurement data of a link between node 3 and node 5 in dataset(1). The horizontal axis is time, the vertical axis is the RSSI value.

Dataset(2) is the dataset of RSSI measurements in long term. We use the measurement data for five days.

3.2 Analysis of clustering metrics

To determine the clustering metrics used for incremental clustering, we used an artificial dataset and the RSSI measurement dataset. We applied k-medoids to each dataset and calculated three clustering metrics given in equations 4 – 6. The number of clusters used in k-medoids was $k = 1$ to $k = n$, where n is the number of data in the dataset. K-medoids was performed 100 times for each k because the result depending on the initial medoids.

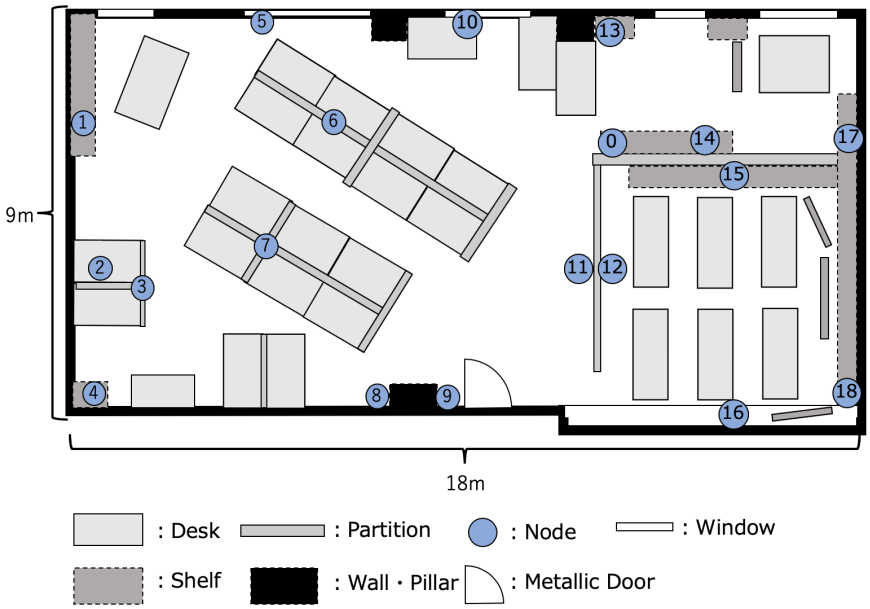

Figure 4: RSSI measurement environment.

3.2.1 Clustering metrics for the artificial dataset

We show the results of three clustering metrics with artificial dataset. The artificial dataset size is 300. Every 100 samples in 300 samples was sampled by the three-dimensional Dirichlet distributions [14] with different parameters. three-dimensional Dirichlet distribution is defined as

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{3}\alpha_i)}{\Pi_{i=1}^{3}\Gamma(\alpha_i)}\theta_1^{\alpha_1-1}\theta_2^{\alpha_2-1}\theta_3^{\alpha_3-1},$$

where θ is a three-dimensional random variable, the parameter α is a three-dimensional vector with components $\alpha_i > 0$, and $\Gamma(x)$ is the Gamma function. The parameters that we used for sampling were $\alpha = (5, 10, 40)$, $\alpha = (7, 30, 8)$, and $\alpha = (24, 5, 8)$. Figure 7 shows the artificial dataset.

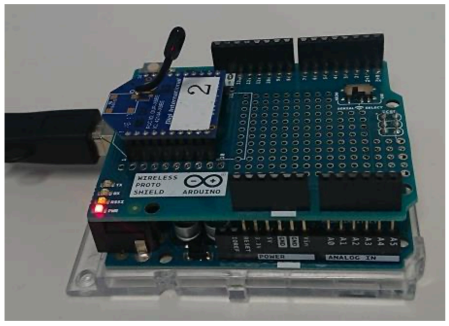

Figure 5: Sensor node used for RSSI measurements.

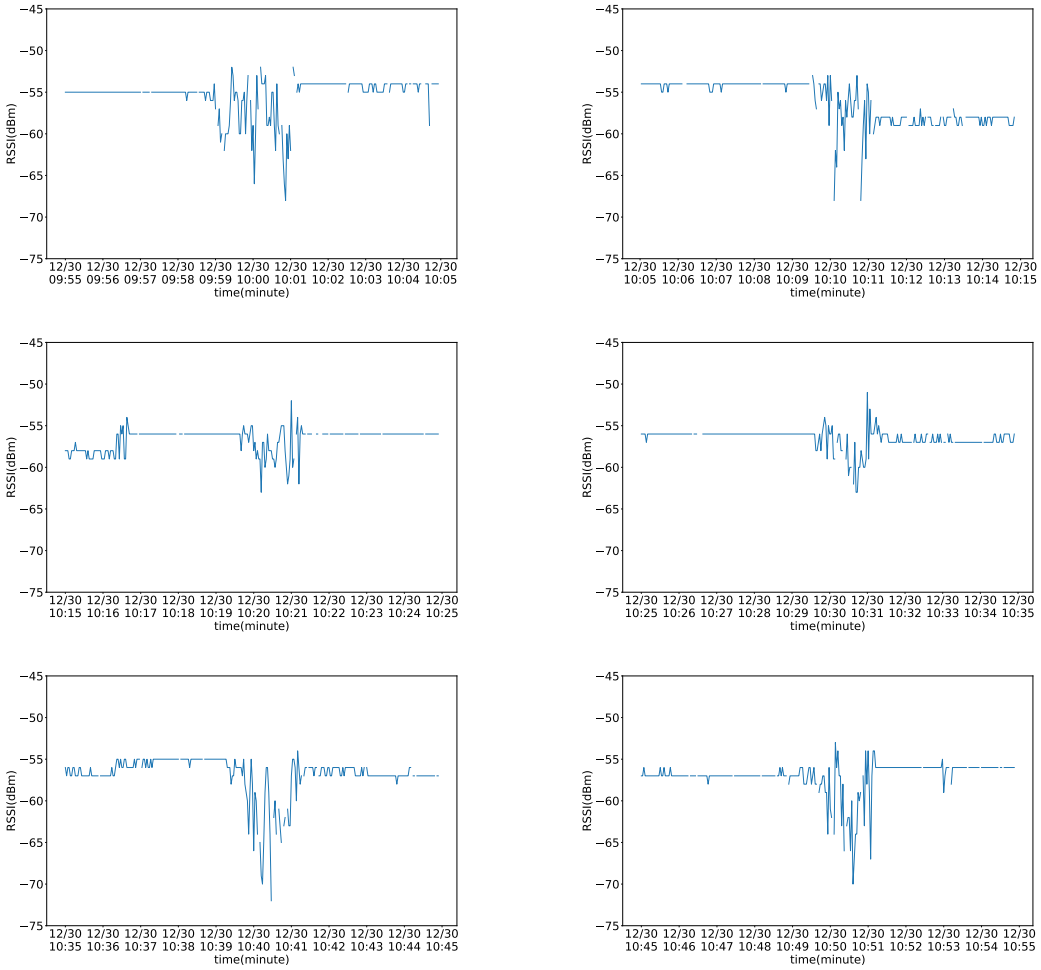

Figure 6: Examples of RSSI of a link between node 3 and node 5.

The clustering metric with the highest value when the number of clusters is three is evaluated as the best clustering metric. Figures 8, 9, 10 show the values of the three clustering metrics given in 4 – 6, respectively. The vertical axis is the clustering metric value, the horizontal axis is the number of clusters k. In Figures 8, 9, 10, the highest clustering metric value among the clustering results for each k was shown. The value of Equation 4 increased as the number of clusters increased (Figure 8). The value of Equation 5 was almost the same when the number of clusters was 10 or more (Figure 9). The value of Equation 6 had the maximum value 18.252 when the number of clusters was three (Figure 10). From this observation, we find that the best clustering metric was for this data set is given by the metric (6).

We then use the clustering metric (6) to perform an incremental clustering for the artificial data set. We present the results in Figure 11. The clustering result split the dataset into three clusters. Each cluster included all data sampled by the Dirichlet distribution of one parameter. The value of Equation 6 was 18.252. This was the same value as the maximum value in Figure 10.

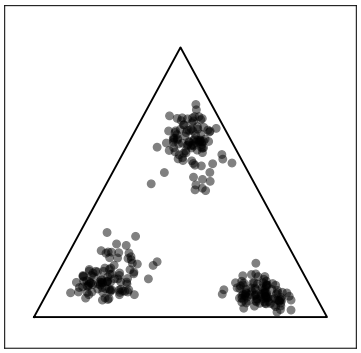

Figure 7: The artificial dataset.

3.2.2 Clustering metrics for the RSSI measurement dataset

We show the results of three clustering metrics with the RSSI measurement dataset. The dataset used to evaluate are the RSSI of a link between node 3 to node 5 in dataset(1). The parameters used to obtain the RSSI distributions were the window size of 1 minute and a bin size of 1 dBm.

Figures 12, 13, 14 shows the values of 4, 5, 6, respectively. The horizontal axis is the clustering metric value, the vertical axis is the number of clusters k. In Figures 12, 13, 14, the highest clustering metric value among the clustering results for each k is shown.

The value of Equation 4 increased as the number of clusters increased (Figure 12). The value

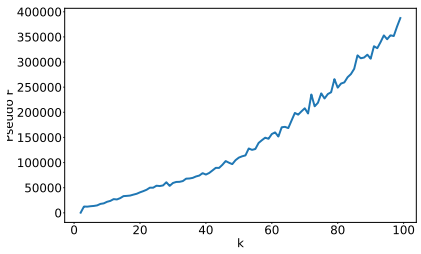

Figure 8: Equation (4) for the k-medoids result of the artificial dataset.

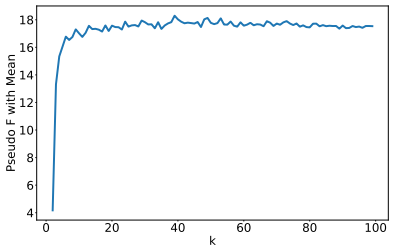

Figure 9: Equation (5) for the k-medoids result of the artificial dataset.

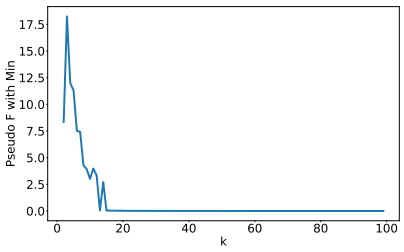

Figure 10: Equation (6) for the k-medoids result of the artificial dataset.

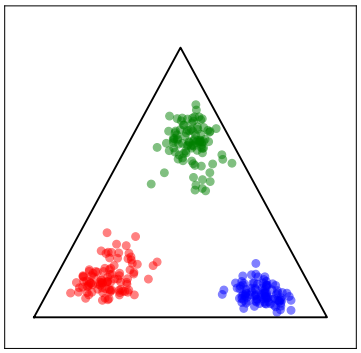

Figure 11: The incremental clustering result of artificial dataset.

of Equation 5 was almost the same when the number of clusters was 20 or more (Figure 13). The value of Equation 6 had the maximum value 18.403 when the number of clusters was nine, and decreased the value when the number of clusters was greater than nine (Figure 14).

3.2.3 Determination the clustering metric used in incremental clustering

From Section 3.2.1 and Section 3.2.2, we use clustering metric (6) as the clustering metric used for the incremental clustering. If we use the clustering metric other than clustering metric (6), each data will be assigned to each new cluster because the clustering metric value increases as the number of clusters increases. Thus, clustering metrics (4) and (5) are not appropriate for incremental clustering.

3.3 Anomaly detection of datasets

3.3.1 Anomaly detection of dataset(1)

We show the anomaly detection results for RSSI time series data of a link between node 3 and node 5 in dataset(1). Clusters are created using incremental clustering with the data for one day before the experiment, and then anomaly detection was performed. Table shows parameters for anomaly detection. In Table , w is the window size for creating RSSI subsequences, b_s is the bin size for creating RSSI distributions, r is the parameter used for calculating anomaly score based on cluster size, b_c is the bin size for calculating anomaly score based on correlation coefficient, T_r is the period used for removing unused clusters. Figure shows anomaly scores and event labels. The horizontal axis is data number. True label represents events that are obstructing a link using a shield sheet or not. Cluster size anomaly represents the anomaly score based on cluster size. Transition anomaly represents the anomaly score based on transition probability. Correlation

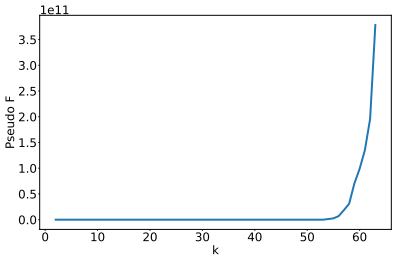

Figure 12: Equation (4) for the k-medoids result of the RSSI measurement dataset.

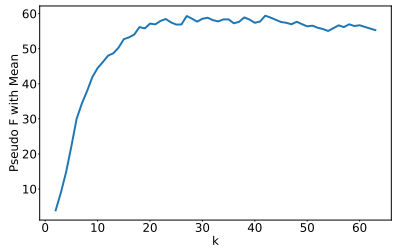

Figure 13: Equation (5) for the k-medoids result of the RSSI measurement dataset.

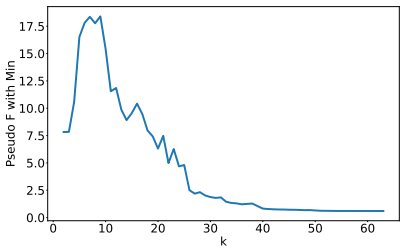

Figure 14: Equation (6) for the k-medoids result of the RSSI measurement dataset.

anomaly represents the mean of the anomaly scores based on correlation coefficient between the link between nodes 3 and 5 and other links. The color depth represents the magnitude of the anomaly score. Figure 16 shows the detection result using anomaly score based on cluster size and threshold θ_a. This detection result is 1 if the anomaly score exceeds the threshold, and 0 otherwise. Figure 17 shows the detection result using anomaly score based on transition probability and threshold θ_b. Figure 18 shows the detection result using anomaly score based on correlation coefficient and threshold θ_c. Figure 19 shows the detection result using three anomaly scores with thresholds $\theta_a = 0.9$, $\theta_b = 0.8$, $\theta_c = 0.6$.

Table 1: Parameters used for anomaly detection for dataset(1).

w	b_s	r	b_c	T_r
1 minutes	1dBm	0.9	0.5	1440 minutes

We show the anomaly detection results for RSSI time series data of a link between node 9 and node 16 in dataset(1). Clusters are created using incremental clustering with the data for one day before the experiment, and then anomaly detection was performed. The parameters used anomaly detection are the same value in Table 1. Figure shows anomaly scores and event labels. The horizontal axis is data number. True label represents events that are obstructing a link using a shield sheet or not. Cluster size anomaly represents the anomaly score based on cluster size. Transition anomaly represents the anomaly score based on transition probability. Correlation anomaly represents the mean of the anomaly score based on correlation coefficient between the link between nodes 9 and 16 and other links. The color depth represents the magnitude of the anomaly score. Figure 21 shows the detection result using anomaly score based on cluster size and threshold θ_a. This detection result is 1 if the anomaly score exceeds the threshold, and 0 otherwise. Figure 22 shows the detection result using anomaly score based on transition probability and threshold θ_b. Figure 23 shows the detection result using anomaly score based on correlation coefficient and threshold θ_c. Figure 24 shows the detection result using three anomaly scores with thresholds $\theta_a = 1.0$, $\theta_b = 1.0$, $\theta_c = 0.5$.

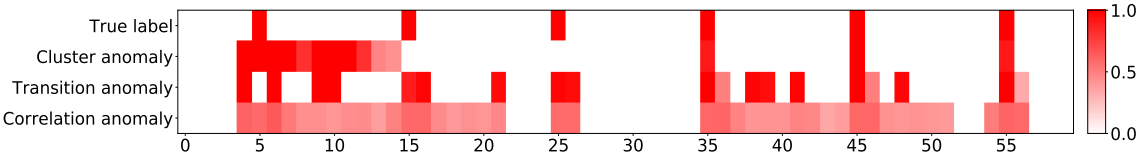

Figure 15: Anomaly scores and events for RSSI of a link between node 3 and node 5 in the dataset(1).

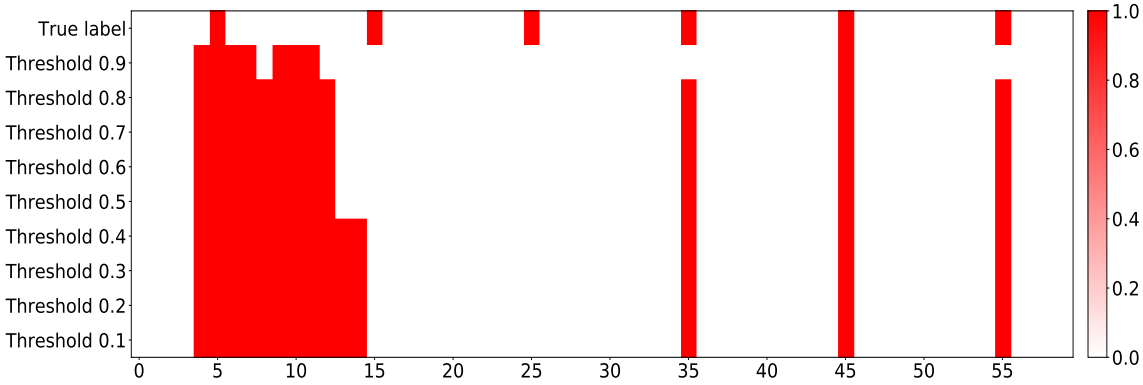

Figure 16: Detection results using anomaly score based on cluster size for RSSI of a link between node 3 and node 5 in the dataset(1).

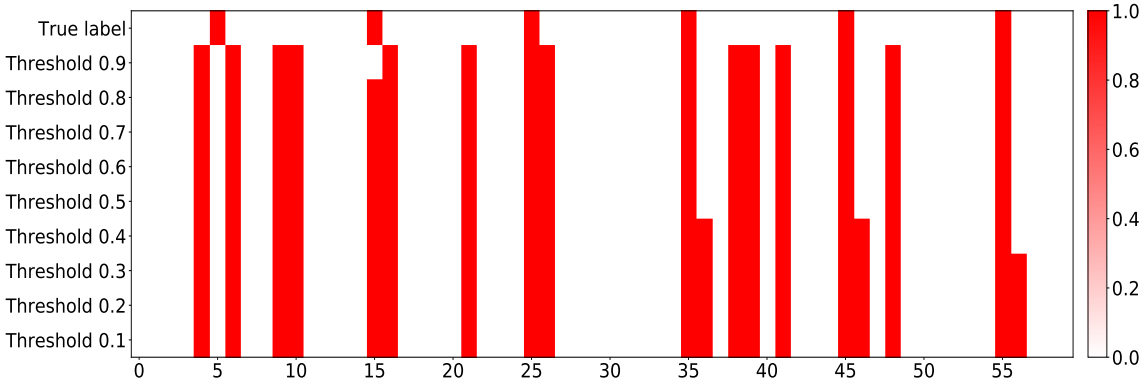

Figure 17: Detection results using anomaly score based on transition probability for RSSI of a link between node 3 and node 5 in the dataset(1).

3.3.2 Anomaly detection of dataset(2)

We show the anomaly detection results for RSSI time series data in dataset(2). In dataset(2), we defined as occurring an event when more than 10 people cross the link per minute. Table 4 shows the anomaly detection results. TP represents true positive. FP represents false positive. TN represents true negative. FN represents false negative. The thresholds used anomaly scores are $\theta_a = 0.9$, $\theta_b = 0.9$, $\theta_c = 0.619$.

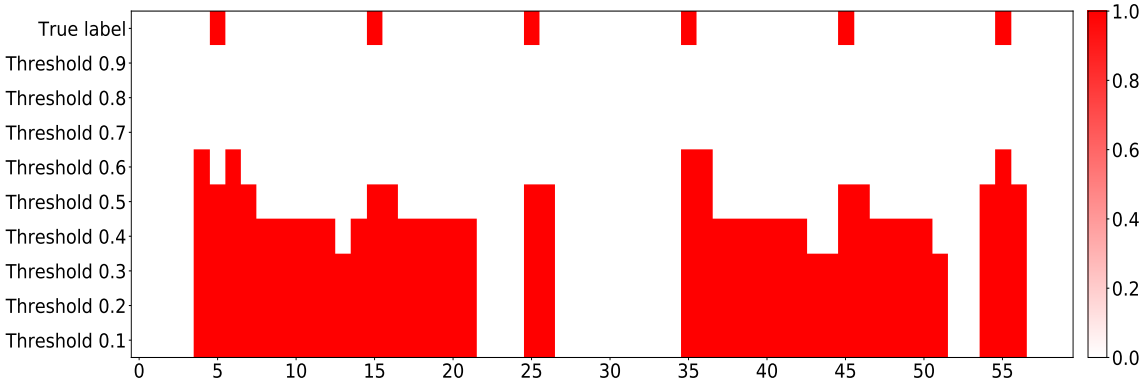

Figure 18: Detection results using anomaly score based on correlation coefficient for RSSI of a link between node 3 and node 5 in the dataset(1).

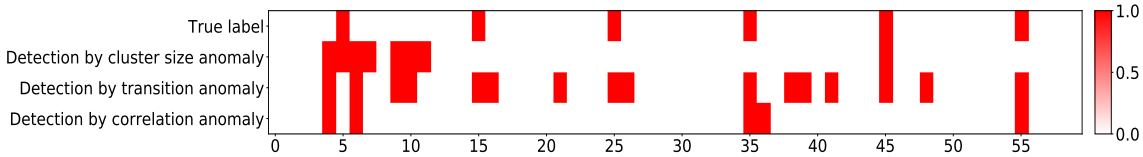

Figure 19: Detection results using the three anomaly scores for RSSI of a link between node 3 and node 5 in the dataset(1).

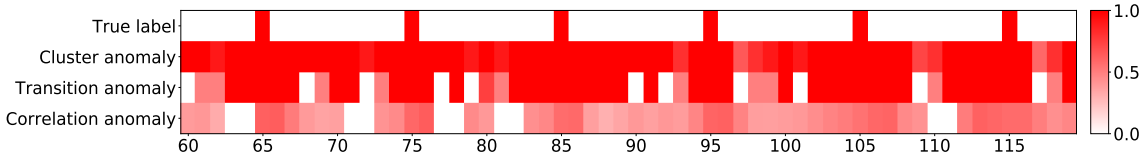

Figure 20: Anomaly scores and events dataset(1) for link 9 to 16.

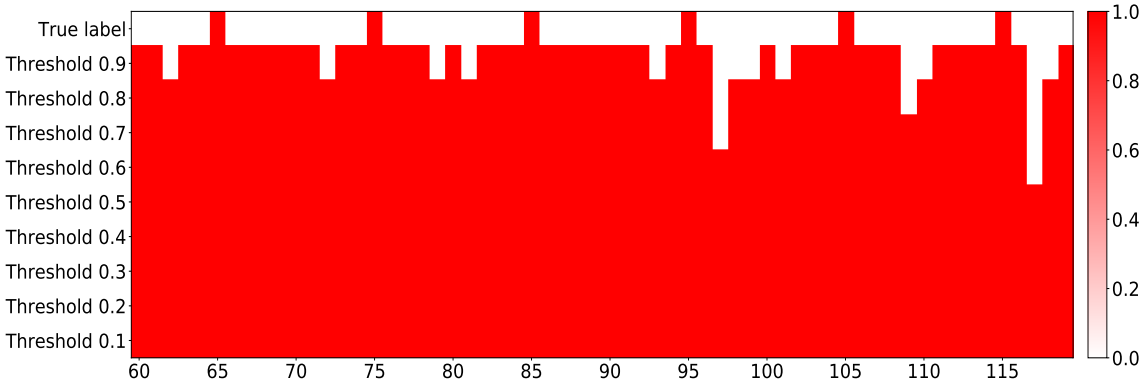

Figure 21: Detection results using anomaly score based on cluster size for RSSI of a link between node 9 and node 16 in the dataset(1).

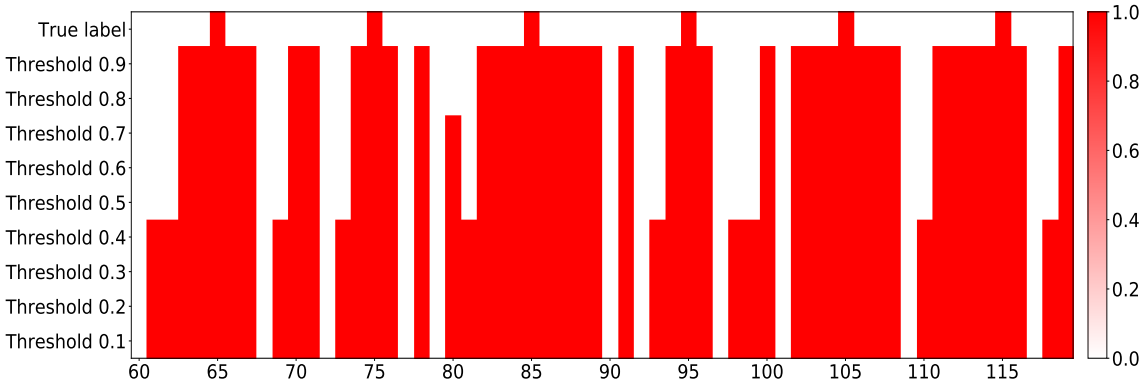

Figure 22: Detection results using anomaly score based on transition probability for RSSI of a link between node 9 and node 16 in the dataset(1).

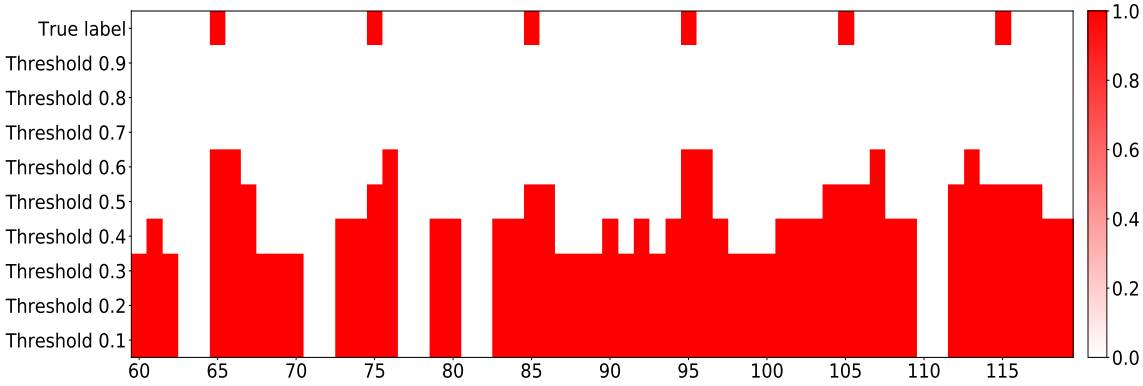

Figure 23: Detection results using anomaly score based on correlation coefficient for RSSI of a link between node 9 and node 16 in the dataset(1).

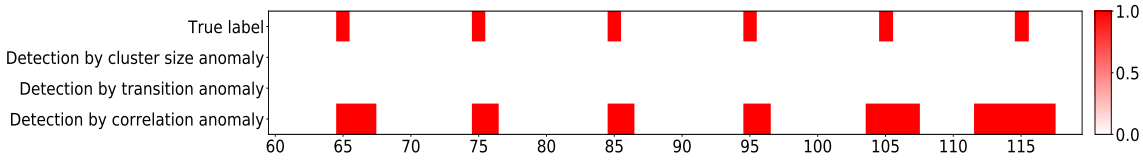

Figure 24: Detection results using the three anomaly scores for RSSI of a link between node 9 and node 16 in the dataset(1).

Table 2: Anomaly detection results for dataset(2).

	Cluster anomaly	Transition anomaly	Correlation anomaly
TP	2	2	3
FP	458	419	541
FN	2	2	1
TN	978	1017	895

4 Discussion

Figures 10, 11, 14 show that clustering metric 6 is an appropriate metric for incremental clustering. Figure 19 shows that the detection is possible by combining three anomaly scores with an appropriate thresholds. In Table , FP is very large, thus the proposed method must improve to reduce false positives.

5 Conclusion

In this thesis, we proposed the anomaly detection method using RSSI to detect unusual situations in industrial wireless sensor networks. The proposed method split RSSI time series data into RSSI subsequences using the jumping window algorithm. To calculate anomaly scores for each RSSI subsequence, we use the incremental clustering and correlation coefficients.

First, we studied the appropriate clustering metric for incremental clustering. Second, we investigate anomaly detection accuracy. We showed that anomaly detection was possible by combining three anomaly scores with an appropriate thresholds. However, the anomaly detection results for the dataset of RSSI measurements in long term showed many false positives.

As future work, we will evaluate the proposed method using the RSSI time series data measured in an industrial environment. This is because these data changes complex by many metal objects, cars, and humans than the dataset used in this thesis.

Acknowledgements

I would like to express the greatest appreciation to my supervisor, Professor Naoki Wakamiya of Osaka University. With his advice and valuable feedback, I learned so much more from him. I am deeply grateful to Associate Professor Masaki Ogura for helpful advice and warm guidance. I also express my appreciation to Assistant Professor Masafumi Hashimoto for various comments. Finally, I thank my colleagues in the Bio-system analysis laboratory, Graduate School of Information Science and Technology of Osaka University for their support.

References

[1] V. C. Gungor and G. P. Hancke. Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Transactions on Industrial Electronics*, 56(10):4258–4265, Oct 2009.

[2] Kay Soon Low, W. N. N. Win, and Meng Joo Er. Wireless sensor networks for industrial environments. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 2, pages 271–276, Nov 2005.

[3] X. Lu, I. H. Kim, A. Xhafa, and J. Zhou. WSN for machine area network applications. In *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, pages 31–36, Sep. 2016.

[4] P. S. Sandra, C. M. Sandeep, V. Nair, M. V. Vindhuja, S. S. Nair, and M. P. Raja. WSN based industrial parameter monitoring using smartwatch. In *2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT)*, pages 1–6, April 2017.

[5] J. Åkerberg, M. Gidlund, and M. Björkman. Future research challenges in wireless sensor and actuator networks targeting industrial automation. In *2011 9th IEEE International Conference on Industrial Informatics*, pages 410–415, July 2011.

[6] Z. Zhang, A. Mehmood, L. Shu, Z. Huo, Y. Zhang, and M. Mukherjee. A survey on fault diagnosis in wireless sensor networks. *IEEE Access*, 6:11349–11364, 2018.

[7] X. Gong, J. Trogh, Q. Braet, E. Tanghe, P. Singh, D. Plets, J. Hoebeke, D. Deschrijver, T. Dhaene, L. Martens, and W. Joseph. Measurement-based wireless network planning, monitoring, and reconfiguration solution for robust radio communications in indoor factories. *IET Science, Measurement Technology*, 10(4):375–382, 2016.

[8] Lior Rokach and Oded Maimon. *Clustering Methods*, pages 321–352. Springer US, Boston, MA, 2005.

[9] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2, Part 2):3336 – 3341, 2009.

[10] Markus Wurzenberger, Florian Skopik, Max Landauer, Philipp Greitbauer, Roman Fiedler, and Wolfgang Kastner. Incremental clustering for semi-supervised anomaly detection applied on log data. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, pages 1–6, 2017.

[11] Goldberger, Gordon, and Greenspan. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 487–493 vol.1, Oct 2003.

[12] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, Jan 1991.

[13] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. Understanding of internal clustering validation measures. In *2010 IEEE International Conference on Data Mining*, pages 911–916, Dec 2010.

[14] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.