International Journal of Internet, Broadcasting and Communication Vol.14 No.4 88-95 (2022)
http://dx.doi.org/10.7236/IJIBC.2022.14.4.88

IJIBC 22-4-11

An Abnormal Worker Movement Detection System Based on Data Stream Processing and Hierarchical Clustering

Dat Van Anh Duong, Doi Thi Lan, and Seokhoon Yoon*

Postdoctoral Researcher, Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Korea
Ph.D Student, Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Korea
Professor, Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Korea
mrdvad11@gmail.com, doilan151188@gmail.com, seokhoonyoon@ulsan.ac.kr

Abstract

Detecting anomalies in human movement is an important task in industrial applications, such as monitoring industrial disasters or accidents and recognizing unauthorized factory intruders. In this paper, we propose an abnormal worker movement detection system based on data stream processing and hierarchical clustering. In the proposed system, Apache Spark is used for streaming the location data of people. A hierarchical clustering-based anomalous trajectory detection algorithm is designed for detecting anomalies in human movement. The algorithm is integrated into Apache Spark for detecting anomalies from location data. Specifically, the location information is streamed to Apache Spark using the message queuing telemetry transport protocol. Then, Apache Spark processes and stores location data in a data frame. When there is a request from a client, the processed data in the data frame is taken and put into the proposed algorithm for detecting anomalies. A real mobility trace of people is used to evaluate the proposed system. The obtained results show that the system has high performance and can be used for a wide range of industrial applications.

Keywords: *Location Data, Anomaly Detection, Streaming System, Data Frame.*

1. Introduction

In this paper, an abnormal worker movement detection system based on data stream processing and hierarchical clustering is proposed. Anomaly detection is the process of identifying data patterns that deviate from expected behavior. This system focuses on the anomalies in human movement patterns. Detecting anomalies in human movement can benefit a variety of monitoring applications in industrial factories [1, 2]. For instance, when a worker is injured in an accident, the worker's movements become abnormal. If the worker's unusual movement patterns are recognized, the worker can be saved quickly. The proposed system is based on Apache Spark [3] and a hierarchical clustering-based anomalous trajectory detection algorithm is

Manuscript Received: September. 7, 2022 / Revised: September. 11, 2022 / Accepted: September. 14, 2022
Corresponding Author: seokhoonyoon@ulsan.ac.kr
Tel: +82-52-259-1403, Fax: +82-52-259-1687
Professor, Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Korea

also proposed to be integrated into the system.

In particular, Apache Spark, which is an open-source computing platform, is used. It is an appropriate framework for streaming processing, iterative processing, batch processing, and interactive querying. Apache Spark provides a suitable framework for continuously querying data and is optimized for the execution of applications utilizing big data. Additionally, it does not require access to the system storage or disk in order to perform these tasks. Message queuing telemetry transport protocol (MQTT) [4] is used in this research to enable the streaming of the locations of workers to Apache Spark. Specifically, workers update their locations regularly to Apache Spark using the MQTT protocol. Then, Spark receives location information and divides it into batches. The necessary information from batches (such as worker identification, record date and time, and worker location) is processed and stored in a data frame. From the information in the data frame, the anomaly in workers' movements is detected.

A method to detect anomalies in the location data is also proposed in this work. Detecting anomalous trajectories has been strongly developed and attracted considerable attention in recent decades. Many methods have been proposed and achieved high efficiency in this field. For example, methods analyze the "few and different" features of trajectories to identify anomalies [5-6]. The density-based anomaly detection methods calculate the neighbor density of trajectories. Then, a trajectory is detected as an anomaly if its density is smaller than a given threshold [7-8]. The clustering-based methods are also applied to detect anomalies by discovering clusters of similarity trajectories in the data set [9-10]. In this work, we proposed a method for detecting anomalies in workers' trajectories based on hierarchical clustering. First, the hierarchical clustering algorithm is used to group a set of trajectories into clusters of similar trajectories. Then, a trajectory is detected as an anomaly or not, relying on the closeness between it and the extracted clusters. The proposed algorithm is integrated into Apache Spark. Apache Spark extracts the input data for the anomaly detection algorithm from the data frame upon receiving an anomaly detection request from a client. The algorithm is then applied to the input data to detect anomalies. Then, Apache Spark is responsible for returning the result to the client after obtaining it from the algorithm.

In order to validate the proposed system, a real data set of workers' locations is used for obtaining clusters and testing the anomaly detection method. We also design an application to stream workers' locations in the real data set to Apache Spark by using MQTT. Then, the anomaly detection results are obtained with the real data. The results indicate that the system has high performance and is suitable for a variety of industrial applications.

The remainder of the paper is structured as follows. In Section 2, we describe the abnormal worker movement detection system. The experimental results are then presented in Section 3. Finally, we conclude this paper in Section 4.

2. The Abnormal Worker Movement Detection System

In this section, the abnormal worker movement detection system is presented in detail. First, we discuss how the locations of workers are transmitted to Apache Spark via the MQTT protocol [4]. Then, processing worker's location data in Apache Spark and saving necessary information in the data frame is described. An anomaly detection method for workers' trajectories is also presented. Finally, how the proposed system processes a client request is discussed. The abnormal worker movement detection system is shown in Figure 1.

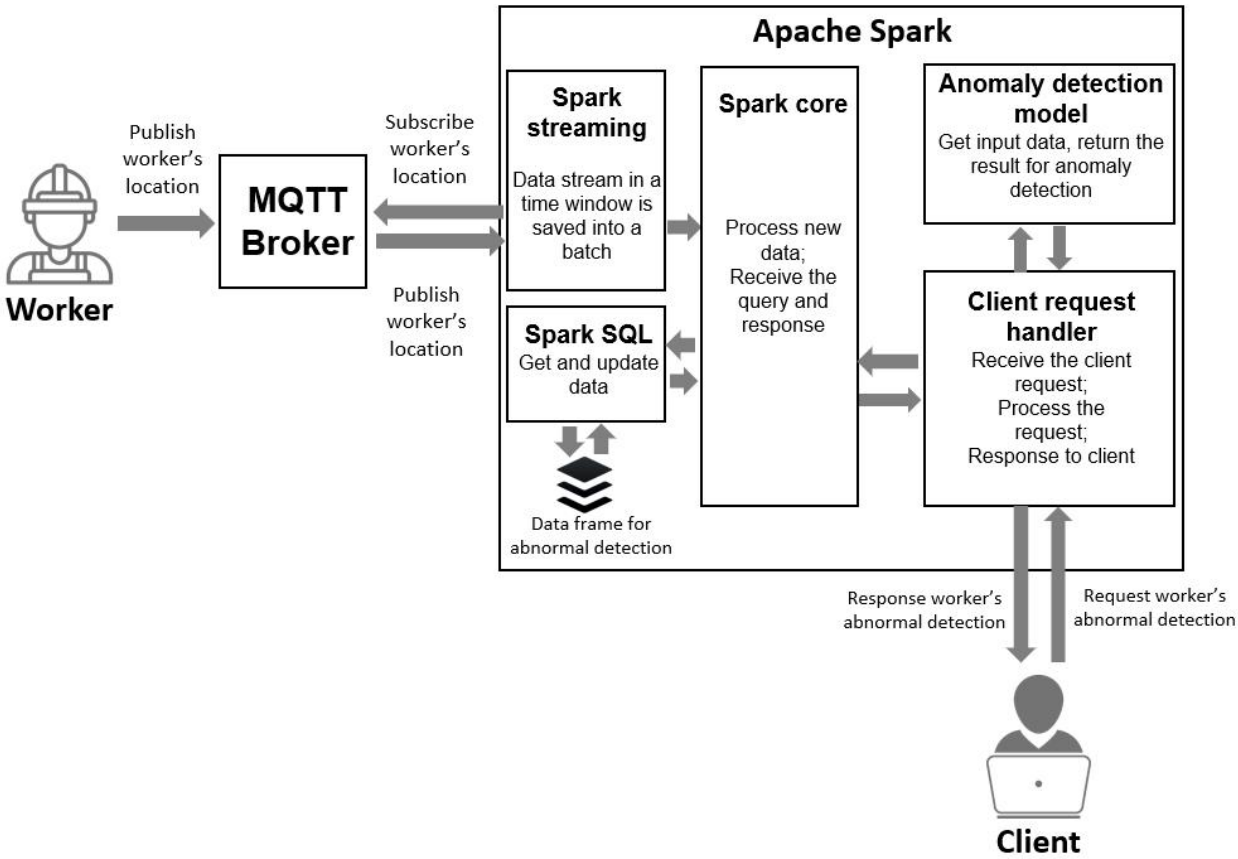

Figure 1. The abnormal worker movement detection system.

2.1 Transmitting Location Information by Using the MQTT Protocol

In the proposed system, workers transmit their location data to Apache Spark over the MQTT protocol. MQTT provides a lightweight method of publish-subscribe network protocol that enables devices to exchange messages with each other. Due to its scalability and support for dynamic application topologies, this protocol is often used in industrial networks. MQTT separates publishers and subscribers, which allows the expansion of new data sources and the replacement of modules. In MQTT, a broker is the one in charge of managing client connections. A client can be either a publisher or a subscriber. It is a subscriber when it wants to subscribe to a topic on the MQTT broker. When it publishes messages on a certain topic to the broker, it will be considered a publisher. When a broker receives data on a topic from the publishers, that data is then sent to all of the subscribers who have subscribed to the topic. MQTT supports varied service quality. Specifically, with quality of service 0 (QoS0), the message is sent only once, and the client and the broker take no additional steps to acknowledge delivery. For QoS1, the message is resent multiple times until the sender receives an acknowledgment. If the quality of service is set to 2, the sender and receiver execute a two-level handshake to ensure that only a single copy of the message is received. As shown in Figure 1, the workers in our system take on the role of publishers and are responsible for publishing their positions to the worker's location topic on the MQTT broker. Apache Spark has subscribed to get updates on the worker's location topic on the broker. In the proposed system, workers transmit their location information to the MQTT broker, and then the broker forwards that information to Apache Spark.

2.2 Handling Data Stream in Apache Spark

As described in the previous subsection, Apache Spark acts as a subscriber to the worker's location topic on the MQTT broker. In this subsection, we discuss how the received data from the broker is processed in Apache Spark.

Apache Spark is a unified analytics engine for large-scale data processing. It provides distributed calculations on big data and enables users to store data in memory and often query it. Therefore, it allows for real-time and iterative processing. Spark's main abstraction is a resilient distributed dataset (RDD), which is a collection of elements that can be executed in parallel and is fault tolerant. RDDs are distributed data structures, and it automatically separates operations into tasks. Spark can persist RDDs in memory, enabling efficient reuse across parallel operations. Each record of a worker's location received from the MQTT broker is considered an RDD in the proposed system. It contains information such as the recording time, the worker's identification, the worker type, the worker team, and the location coordinates.

In Apache Spark, RDDs received in a time window from an MQTT broker are grouped into a batch. Then, processing data is performed on each batch that allows us to use Spark's API in streaming environments. Specifically, in our system, Spark streaming receives the worker's position data as RDDs and saves RDDs during a time window into a batch. Then, steps for extracting data are executed on each batch in Spark core. The necessary information in RDDs (e.g., the worker's identification, the recording time, and the location coordinates) is collected.

Spark SQL is a module that provides operations for structured data processing. It can also serve as a distributed SQL query engine for data. Spark provides a new data structure called the data frame that allows us to process as well as query data with Spark SQL. In this system, the collected data is stored in a data frame. Each attribute, such as the worker's identification, the recording time, and the location coordinates, corresponds to each field in the data frame. The data frame is also updated after each time window when there is a new batch.

2.3 The Anomaly Detection Algorithm

In this subsection, we propose an algorithm for detecting anomalies in workers' trajectories based on the trajectories' clustering. First, the hierarchical clustering algorithm is used to group a set of trajectories into clusters of similar trajectories. Then, a trajectory is detected as an anomaly or not, relying on the closeness between it and the extracted clusters [11].

The framework of our method is shown in Figure 2. Specifically, we find clusters from the training data set using hierarchical clustering. Note that we need to calculate a distance matrix of trajectories in the data set before applying the clustering algorithm. In this work, we chose the edit distance on real sequence (EDR) to determine the distance of trajectories [12]. Besides, the hierarchical clustering algorithm also requires an input parameter, which is the number of clusters. Determining this parameter is an important task of hierarchical clustering. Here, we use the WB-index to determine the number of clusters [13]. This index aims to maximize the compactness within each cluster and the separation between clusters in the data set. In hierarchical clustering, the distance between clusters is calculated to find which two clusters should be merged. The single link method is used in our work. In the single-link method, the distance between two clusters is defined as the minimum distance between two trajectories in the two clusters.

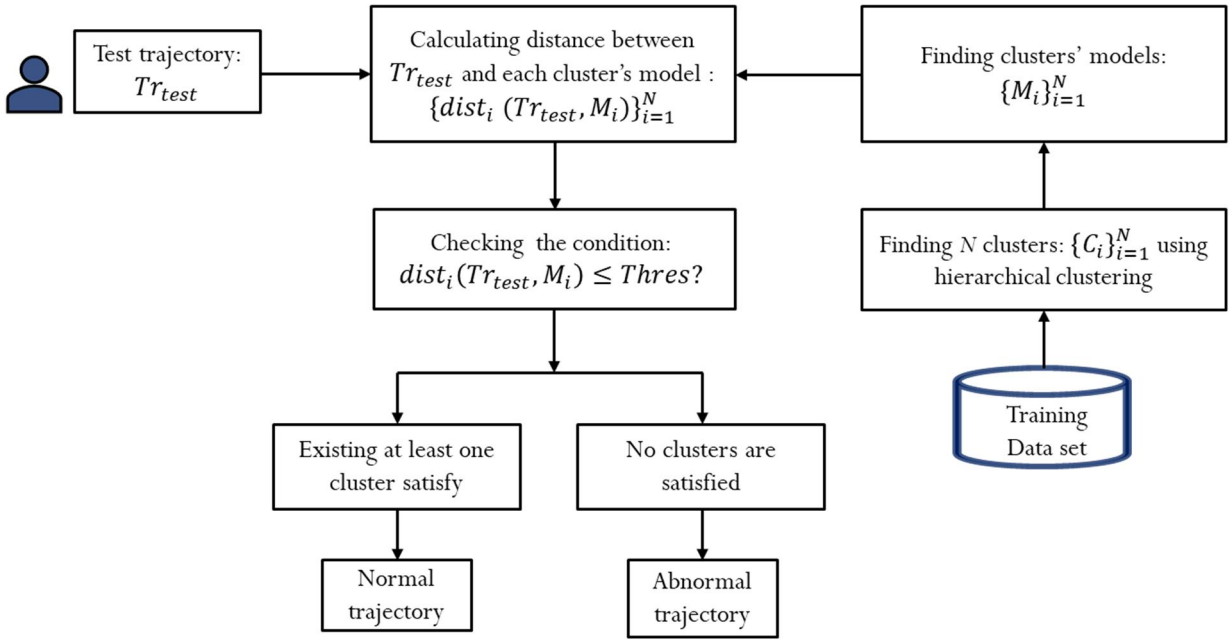

Figure 2. Hierarchical clustering-based anomalous trajectory detection framework.

From N extracted clusters $\{C_i\}_{i=1}^{N}$, a test trajectory Tr_{test} is determined as an anomaly or not. To do this, we need to calculate the closeness between Tr_{test} and each cluster C_i in the data set. First, each cluster C_i is represented by a cluster's model M_i. In this work, M_i is determined as a trajectory that has a minimum average distance with respect to the other trajectories in the cluster [11]. After determining model of N clusters $\{M_i\}_{i=1}^{N}$, the distance between Tr_{test} and each model M_i ($dist_i(Tr_{test}, M_i)$) is calculated. If $dist_i(Tr_{test}, M_i) \leq Thres$, Tr_{test} belongs to the cluster C_i. Tr_{test} is detected as an anomaly if it does not belong any clusters. Here, $Thres$ is a given distance threshold to determine the closeness between a trajectory and a cluster. In our work, $Thres$ is determined as the following equation:

$$Thres = \mu_{Dist} + \beta * \sigma_{Dist} \quad (1)$$

where β is a predefined parameter by user, μ_{Dist} and σ_{Dist} are the mean and standard deviation values of trajectories' distances in the data set, respectively.

As shown in Figure 1, the anomaly detection algorithm is integrated as a component in Apache Spark. It will perform when there are requests from clients.

2.4 Client's Request Handler

In this subsection, we discuss how to handle the requests of clients. As shown in Figure 1, whenever there is a request for abnormal detection, the client sends the request to Apache Spark. In order to communicate between clients and Apache Spark, one of the internet of things protocols can be used. In this system, the TCP socket protocol is used. Apache Spark initiates a new thread called the client request handler, which acts as a server to receive the request from the client. From the client's request, the client request handler obtains the identification of the worker who must verify the movement. This information is transmitted to the Spark core in order to query data in the data frame. Specifically, using the worker identification, Spark SQL queries the

data frame to obtain the input for the anomaly detection algorithm (i.e., the location coordinates of the worker in the last ten minutes). After getting the input from the data frame, Spark core returns it to the client request handler. The data is then input into the model for performing anomaly detection. Finally, the anomaly detection result is returned to the client request handler, which then responds to the client.

3. Experimental Results

In this study, the performance of the system is evaluated using a real dataset of worker locations (the MIT Badge data set [14]). A program was also developed to transmit the location data of workers to the broker via the MQTT protocol. The system will perform the anomaly detection algorithm on MIT Badge data to respond to the client.

3.1 Experimental Setup

The MIT Badge data set contains time-stamped locations for workers at an IT Call Center in Chicago over one month, and the location is updated every six seconds. There are three groups with 36 workers who participated in the data collection: the configuration group has 25 workers; the pricing group has 7 workers; and the coordinator group has 4 workers.

Since the MIT Badge data set does not contain abnormal trajectories of workers, we need to make some assumptions for anomalies. In this data set, the number of configuration group achieves approximately 78 % of the total, while the number of pricing group accounts for only 22 %. Therefore, we can assume that workers in the pricing group are anomalies while workers in the configuration group are normal. The original dataset is divided into two parts for evaluation. The first part accounts for 70% of the total and is used to find clusters and the value of $Thres$. The second part is 30% and is used to estimate the algorithm performance.

The aim of our work is to detect anomalous trajectories of workers as quickly as possible, so we divide workers' trajectories for anomaly detection using a time slide window W. In this work, we choose: $W = 10$ minutes. A working day is considered from 9:00 am to 6:00 pm, divided into 54 timeslots each day. Detecting anomalies is processed following each timeslot. To determine the value of $Thres$, we choose: $\beta = -0.6$.

For testing the system, one machine is used to run Apache Spark and another machine is used to run the program that simulates the streaming location of 36 workers in the MIT Badge dataset. Using a different machine, we send requests from 20 clients to Apache Spark. The configuration of these machines is the same (i.e., CPU: Intel Xeon E3-1240 V2; Memory: 16GB of 1600 MHz DIMM DRAM; Hard Disk Drive: 1TB).

3.2 Experimental Results

Three metrics (i.e., recall, precision, and F1-socre) are used to verify the algorithm performance. Recall is calculated as the true positive over the total of true positive and false positive. Precision is the true positive over the total of true positive and false negative. F1-score is calculated based on the values of precision and recall as in Equation (2).

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2)$$

Table 1. Result of proposed method on the MIT Badge data set.

Recall	Precision	F1-score
89.23%	67.38%	76.76%

The result of the algorithm is presented in Table 1. As shown in the table, our hierarchical clustering-based anomalous trajectory detection method has F1-score achieving 76.76% with recall at 89.23 % and precision at 67.38%. This means that our system has a high possibility of detecting true anomalies. This information is beneficial for numerous industrial applications. The abnormal worker movement detection system runs well with 20 clients and 36 workers.

4. Conclusion

In this study, we proposed an abnormal worker movement detection system based on data stream processing and hierarchical clustering. First, the message queuing telemetry transport protocol (MQTT) is used to send the workers' location from the worker to Apache Spark. Specifically, the workers act as publishers who post the information to the workers' location topic, whereas Apache Spark is a subscriber of that topic. The location data is then processed by Spark core and stored in a data frame using Spark SQL. In order to detect the anomaly, a hierarchical clustering-based anomalous trajectory detection algorithm is proposed. It is integrated as a part of the proposed system. When a client makes a request, the client request handler is responsible for receiving the request, and then Spark core queries to the data frame to extract the input for the anomaly detection algorithm. Finally, the client request handler obtains the anomaly detection result from the algorithm and delivers it to the client. The proposed system is evaluated using a real dataset of worker movements. The obtained results show that our system achieves a high F1-score, performs well with a large number of clients, and is suitable for a wide range of industrial applications.

Acknowledgement

This work was supported in part by the Institute of Information and Communication Technology Planning and Evaluation (IITP) Grant by the Korean Government through MSIT (Development of 5G-Based Shipbuilding and Marine Smart Communication Platform and Convergence Service) under Grant 2020-0-00869, and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2021R1I1A3051364.

References

[1] F. Zhang, H. A. D. E. Kodituwakku, W. Hines, and J. B. Coble, "Multi-layer data-driven cyber-attack detection system for industrial control systems based on network, system and process data." *IEEE Transactions on Industrial Informatics* 15, 7, pp. 4362–4369, 2019.
DOI: 10.1109/TII.2019.2891261

[2] D. Wijayasekara, O. Linda, M. Manic, and C. Rieger, "Mining building energy management system data using fuzzy anomaly detection and linguistic descriptions." *IEEE Transactions on Industrial Informatics* 10, 3, pp. 1829–1840, 2014.
DOI: 10.1109/TII.2014.2328291

[3] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing." *In 9th Symposium on Networked Systems Design and Implementation*, pp. 15-28, 2012.

[4] S. A. Shinde, P. A. Nimkar, S. P. Singh, V. D. Salpe, and Y. R. Jadhav, "MQTT-message queuing telemetry transport protocol." *International Journal of Research*, 3(3), pp.240-244, 2016.

[5] C. Che, D. Zhang, P. S. Castro, N. Li, L. Sun, S. Li, Z. Wang, "iBOAT: Isolation-based online anomalous trajectory detection." *IEEE Transactions on Intelligent Transportation Systems.* Vol. 14(2), pp. 806-818, Feb 2013.
DOI: 10.1109/TITS.2013.2238531

[6] P. Banerjee, P. Yawalkar, and S. Ranu, "Mantra: a scalable approach to mining temporally anomalous sub-trajectories." in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery,*pp. 1415-1424, Aug 2016.
DOI: 10.1145/2939672.2939846

[7] J.-G. Lee, J. Han, and X. Li, "Trajectory outlier detection: A partition-and-detect framework." *in 2008 IEEE 24th International Conference on Data Engineering.* pp. 140–149, Apr 2008.
DOI: 10.1109/ICDE.2008.4497422

[8] Z. Zhu, D. Yao, J. Huang, H. Li, and J. Bi, "Sub-trajectory-and trajectory-neighbor-based outlier detection over trajectory streams." in *Pacific-Asia Conference on Knowledge Discovery and Data Mining,* pp. 551-563, Jun 2018.
DOI: 10.1007/978-3-319-93034-3_44

[9] Y. Wang, K. Qin, Y. Chen, and P. Zhao, "Detecting anomalous trajectories and behavior patterns using hierarchical clustering from taxi GPS data." *ISPRS International Journal of Geo-Information*, *7*(1), p.25, Jan 2018.
DOI: 10.3390/ijgi7010025

[10] Z. Fu, W. Hu, and T. Tan, "Similarity based vehicle trajectory clustering and anomaly detection." in *IEEE International Conference on Image Processing*, vol. 2, pp. II–602, Sep 2005.
DOI: 10.1109/ICIP.2005.1530127

[11] N. B. Ghrab, E. Fendri, and M. Hammami, "Abnormal events detection based on trajectory clustering." In *2016 13th International Conference on Computer Graphics, Imaging and Visualization*, pp. 301-306, Mar 2016.
DOI: 10.1109/CGiV.2016.65

[12] L. M. Chen, T. Ozsu, and V. Oria. "Robust and Fast Similarity Search for Moving Object Trajectories." *In Proc. ACM SIGMOD International Conference on Management of Data,* pp. 491–502, Jun 2005.
DOI: 10.1145/1066157.1066213

[13] Q. Zhao, and P. Franti, "WB-index: A sum-of-squares based index for cluster validity." *Data & Knowledge Engineering,* 92, pp.77-89, Jul 2014.
DOI: 10.1016/j.datak.2014.07.008

[14] D. O. Olguin, B. N. Waber, T. Kim, A. Mohan, K. Ara, and A. Pentland, "Sensible organizations: Technology and methodology for automatically measuring organizational behavior." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39 (1), pp. 43-55, Dec 2008.
DOI: 10.1109/TSMCB.2008.2006638