The adaptive spatio-temporal clustering method in classifying direct labor costs for the manufacturing industry

Mateusz Kalinowski
Meritus Systemy Informatyczne
Warsaw, Poland
mateusz.kalinowski@meritus.pl

Jakub Baran
Institute of Nuclear Physics
Academy of Sciences,
Cracow, Poland
jakub.baran@ifj.edu.pl

Paweł Weichbroth ✉
Gdansk University of Technology
Department of Software Engineering,
Gdansk, Poland
pawel.weichbroth@pg.edu.pl

Abstract

Employee productivity is critical to the profitability of not only the manufacturing industry. By capturing employee locations using recent advanced tracking devices, one can analyze and evaluate the time spent during a workday of each individual. However, over time, the quantity of the collected data becomes a burden, and decreases the capabilities of efficient classification of direct labor costs. However, the results obtained from performed experiments show that the existing clustering methods have failed to deliver satisfactory results by taking advantage of spatial data. In contrast to this, the adaptive spatio-temporal clustering (ASTC) method introduced in this paper utilizes both spatial and time data, as well as prior data concerning the position and working status of deployed machines inside a factory. The results show that our method outperforms the bucket of three well-known methods, namely DBSCAN, HDBSCAN and OPTICS. Moreover, in a series of experiments, we also validate the underlying assumptions and design of the ASTC method, as well as its efficiency and scalability. The application of the method can help manufacturing companies analyze and evaluate employees, including the productive times of day and most productive locations.

1. Introduction

Wage pressures across major advanced economies have been intensifying, especially if we consider the pre-pandemic period. In the USA, the Employment Cost Index (ECI), measuring the cost of labor changes, has risen from 111.8 points in Q1 2010 to 138.9 points in Q1 2020 [1]. In addition to the tighter job market, the majority of countries and private companies have increased minimum wages [2].

The US Bureau of Labor Statistics estimates that there were 12.7M people employed in the Manufacturing Industry sub-sector in 2018 [3]. Strong domestic manufacturing is vital to the economy since manufactured goods are necessary for trade. Typically, these goods are produced in factories or other large production systems which have become increasingly complicated [4]. Nevertheless, timely and cost-efficient production is of the utmost importance in remaining competitive.

Production efficiency refers to the quality and effectiveness of work. In other words, cost efficiency is the demonstrated ability to execute plant operations using relatively few total input resources [5]. Considering the human factor, the labor efficiency is the number of earned hours of productive work divided by the number of work hours available in a day. Since many of the modern production technologies are still labor-intensive, their efficiency is greatly influenced by the cost of labor [6].

The cost of labor is the sum of all wages paid to the employees, as well as the cost of employee benefits and payroll taxes paid by the employer [7]. Labor costs can be divided into two categories, namely direct and indirect. While the former include wages for the employees who produce the products (including workers on assembly lines), the latter are associated with support labor (such as the employees responsible for maintenance). In principle, accurate measurement of man hours is a must in the manufacturing industry [8].

While micro and small firms usually 'guesstimate' the man hours for a particular job of limited scope, generally achieving a reasonable level of accuracy, for medium and large companies this method is no longer valid, and other, more accurate and reliable techniques must be applied, in particular to estimate direct labor costs. These estimates are often performed by individuals using combinations of personal judgment and analytical techniques [9]. The major drawbacks of such methods are inconsistencies, inaccuracy and uncertainty due to the subjective nature of the process [9, 10, 11]. The consequences of incorrect calculations are far-reaching, influencing issues ranging from production profitability to labor efficiency.

URI: https://hdl.handle.net/10125/70638
978-0-9981331-4-0

Considering manufacturing processes, evidence-based estimation methods operate on collected data regarding workers' locations and the surrounding machines. So far, a variety of technologies such as radio frequency identification (RFID), ultra wideband (UWB), global positioning system (GPS) and multiple sensor systems have been successfully deployed in many scenarios with the aim of labor cost reduction [12], reduced inventory shrinkage [13], and on-site labor consumption analysis and payment negotiations [14], as well as other directly measurable benefits.

However, implementing tracking technologies in the considered application returns a very high volume of data that results in a constant demand for increased storage. Obviously, the processing time increases with the size of the data, which negatively affects the analysis capabilities. One possible solution relies on the possibility to cluster big data in a compact set, preserving the informative value of the entire dataset.

The goal of this study was twofold. Firstly, we aimed to investigate the performance of the existing clustering methods applied to the above-noted problem. Towards this aim, we tested and evaluated three clustering methods, namely DBSCAN, HDBCAN and OPTICS, using experimental data and the corresponding input parameters. Secondly, we aimed to develop a highly efficient and scalable clustering method that is able to outperform its antecedents in the domain of our interest. To this end, we implemented these four methods for the purpose of conducting their comparison.

The paper is organized in the following manner. Section 2 briefly presents the clustering methods selected for this study. Section 3 discusses the results obtained from preliminary research. Section 4 introduces our clustering method, followed by the design of the experimental setup. Section 6 deals with the performance evaluation, followed by the discussion, conclusions and final remarks, given in Sections 7 and 8, respectively.

2. Research Background

The advent of the Internet of Things (IoT) has again sparked the interest of researchers in the fundamental methods of data mining, such as clustering. However, by design clustering algorithms with super-linear computational complexity are, in fact, not well suited to the context of Big Data [15]. As we know, for even two clusters, solving the clustering problem exactly is NP-hard [16]. When dealing with very large data volumes, the clustering problem is one of the most important issues [17].

The definition of a clustering problem is formulated as follows: given a data set $X = \{x_1, x_2, \ldots, x_m\}$ and an integer value *k*, the issue is to find a such mapping $f : X \rightarrow \{1, \ldots, k\}$, where each item x_l, $l \in \{1, \ldots, m\}$ is assigned to one cluster K_j, where $j = 1, \ldots, k$. Cluster K_j contains the mapped items, where $K_j = \{x_l \in X | f(x_l) = j$, for $l = 1, \ldots, m\}$. Each item within a cluster is more similar to the rest of the items within that cluster than to the items from other clusters. Based on the computed similarities among the items, the objective is to find a coherent and valid organization of multivariate data [18]. To put it more informally, clustering can be defined as the search for "natural" groupings.

The clustering methods can be roughly grouped into five types: partitioning, hierarchical, grid-based, model-based and density-based [19]. Since the partitioning methods are claimed to be easy to implement, while using an iterative method to create the clusters, the number of clusters should be predefined and only spherical shaped clusters can be determined. The hierarchical methods easily handle any forms of similarity or distance, however they reveal high complexity, suffering from the ambiguity of termination criteria. The grid-based, claimed to reveal fast processing time and tolerate noise, are rather not valid for high dimensional data. The model-based methods, depending on the hypothesized model or structure, are able to automatically specify the number of clusters. Last but not least, the density-based are able to detect arbitrary shaped clusters, handling noise as well.

Now, let us consider the nature of a manufacturing process in the context of human behavior in a fixed factory area. The collected data from each working day will differ to some extent, unless there are strict passages to move around the plant and fixed places to work. In our opinion, this scenario is much less probable to exist in modern factories due to worker burnout and dissatisfaction.

Having said that, and due to the size limitations of this paper, we chose to test and evaluate three density-based methods, namely DBSCAN, HDBCAN and OPTICS.

Published in 1996 by Ester *et al.* [20], the DBSCAN (Density Based Spectral Clustering of Applications with Noise) algorithm locates regions of high density which are separated from one another by regions of low density. Initially, the method was proposed for clustering spatial data, and the results of the clustering usually indicate acceptable performance [21]. In its set-up, the most time-consuming step is the calculation of the similarity between data items, while the clustering itself requires only a single dataset scan. By design, DBSCAN requires two initial parameters, namely *Eps*

(the radius of the cluster) and *MinPts* (the minimum items required inside the cluster). Choosing appropriate values for both parameters has a significant influence on the clustering results [22]. Moreover, the algorithm is sensitive to the order in which items are processed, and therefore the clustering result depends on the sequence in which the clusters are constructed [21]. Several different improvements have been developed for the DBSCAN algorithm related to the core and noise objects, and the adjacent clusters [21, 23, 24, 25].

Campello *et al*. introduce a hierarchical clustering method, called HDBSCAN [26]. This method theoretically and practically improved its predecessor. It searches the input data space for regions of high density separated by regions of low density, using a cluster stability metric and a mutual reachability distance [27]. Beyond the minimum cluster size (*MinPts*), which is much easier to choose than *Eps*, the method requires no further setting of arbitrary or biasing parameters by a user [28]. Though HDBSCAN is claimed to be robust [29], the algorithm to build the hierarchy runs in quadratic time, in both the worst and the best case [30].

The OPTICS method (abbr. from Ordering Points To Identify the Clustering Structure) [31], actually being the predecessor of HDBSCAN, can detect meaningful clusters in data of varying density. Hence, OPTICS is broadly used to cluster trajectories [32]. However, the method is argued to be inefficient when faced with large datasets and expensive distance measures due to its quadric complexity in terms of both distance function and time calls [33].

3. Preliminary Research

In our preliminary research, the three above methods underwent a series of experiments which had two goals. While the first concerned clustering trajectory data, the second referred to the evaluation of computed clusters. In particular, in the former task each cluster represented a group of signals, while in the later, two distinct classes, namely: positive (P) and negative (N), corresponded to the correctly and incorrectly identified membership for individual signals, respectively. Here, the membership concerned two exclusive categories: productive and unproductive work.

By definition, productive time is the time during which useful work is performed in an operation or process [34]. Therefore, in the context of this study, productive work was associated with the employee's physical presence next to the particular machine, or the employee's path necessary to reach its location. On the contrary, unproductive work denotes the rest of the employee's activity. In other words, the direct linkage between the employee and machine's physical locations was used to categorize working time.

The left panel (white background) of Figure 1, below, depicts the collected trajectory data from an 8-hour day shift of a single employee working in a factory the size of the depicted square. The right panel (black background) presents an image, divided into a grid, where each square represents the relative frequency of the employee's physical position collected inside the factory, with a color assigned from a color palette.

An employee location (event) is a 3-tuple $\{(x, y), timestamp\}$. The parenthesized values refer to two-dimensional coordinates, namely longitude (x) and latitude (y), while the *timestamp* is a record that shows the date and time of the occurred event. Even though the images (see Fig. 1) themselves lack the ability to exhibit the sequence of the events, this simplified view still is able to provide an informative reconstruction of an employee's time spent during the workday.

Nevertheless, being dissatisfied with the results obtained from the testing, and evaluating the bucket of the aforementioned methods (see Table 1), we decided to develop a method which takes advantage of both the spatial and time data. Based on the best of our knowledge, we set up the following list of objectives as necessary to be fulfilled:

- adaptive creation of clusters, depending on the actual data read-out and prior information regarding the positions of the machines;
- automatic identification of areas where the employee's location is productive and not productive;
- self-assignment of employee paths to the corresponding machines.

During the method development, we first focused on exploring the existing approaches to solve similar problems defined by the specifications at an abstract level. Next, we chose Python as the programming language. Offering a number of effective and matured analytics libraries for numerical computing, data analysis, visualization and machine learning [35], Python has been adopted at a tremendous rate recently in data-driven projects [36].

4. The ASTC Method

Due to the nature of the problem, as mentioned above, the method utilizes both time and spatial data, regarding the employee's position and its occurrence in time. As we discussed in the previous section, the

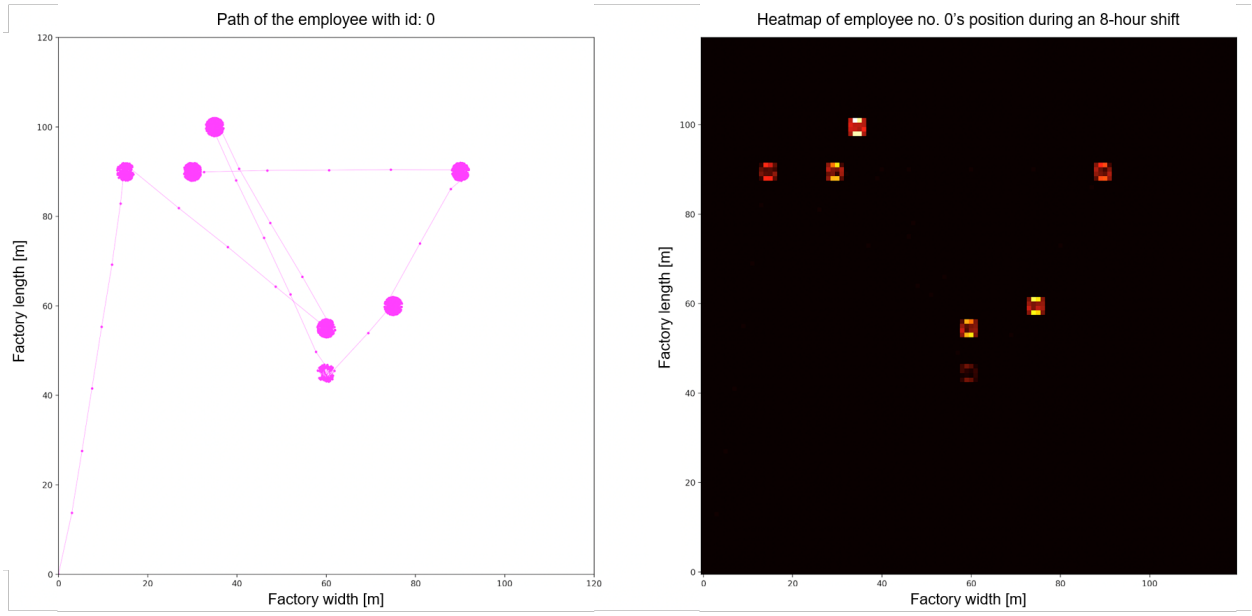

Figure 1. Visualization of a single employee's path (left) and the employee's position frequency map (right).

defined objectives laid the foundations for the following procedure which constitutes the ASTC method.

Every position read-out is analyzed independently in the context of the previous probes. When the algorithm starts, a new cluster is initialized. Afterwards, subsequent points are assigned to the cluster based on the analysis of previous read-out positions. The constraint given to the point to be incorporated into the current cluster is to have at least 1 neighbor in the previous 20 samples. A neighbor is defined as a probe at a distance of no more than 8.5 m from the currently analyzed position.

The presented parameters were carefully investigated (number of analyzed probes backwards, the distance defined by the neighbor, and the minimum number of neighbors considered to determine whether the point is to be inside or outside of the cluster). The initial parameters were evaluated by an expert and then the optimization process was performed (the values were changed by up to $\pm 5\%$ from those indicated by the expert) to ensure they were most likely to achieve the most accurate results (probe assignment to the correct cluster from the two available: P and N).

If the probe does not obey the abovementioned constraints, a new cluster is initialized. The resulting clusters are then processed to allocate them to the appropriate classes (productive time, P) and (unproductive time, N), in order to calculate the cost of the manufacturing.

5. Experimental Setup

A simulation of 50 employees working between 6 a.m. and 2 p.m. was prepared. In order to create a realistic database for testing machine learning and deep learning algorithms, code was developed to simulate a signal recording from geolocation sensors (machine and employee) and current sensors (machine). In total, in the performed simulation, 144,421 samples were generated. The simulation software was developed in Python, currently totaling over 500 lines of code.

It is worth noting here that the share of paths represented only 1.1% of the entire dataset. Therefore, this subset were excluded from the comparison analysis between the introduced method and the bucket of three methods.

Moreover, the following assumptions were made in the simulations:

- a signal from each sensor is recorded every 10 seconds;
- position of all sensors on the z axis is constant, equal to 100 cm;
- factory size is set to 120 x 120 m^2 . Entry and/or exit doors are located at the (0,0) position;

Each machine is configured with the following parameters:

- if the machine is on and running then the sensor indicates a voltage of 230 V,

- if the machine is on and in sleep mode then sensor indicates a voltage of 100 V,
- if the machine is off then the sensor indicates a voltage of 10 V.

The positions of the machines and the operation time in a specific mode are selected arbitrarily and set to 8 h work intervals. The voltage and geolocation sensors transmit the signal at the same time. The order of usage of the machines is randomized for each employee (all seven per shift) and the operation time for each machine depends on their sequence. Each employee moves at a speed of 125–155 cm/s, randomized with the probability characterized by a uniform distribution. Subsequent signals are generated when moving towards another machine based on a directional vector between the current position and the position of the machine sensor, taking into account the randomness of the employee's movement speed.

Additionally, the position of the signal generated from the geolocation sensor is modeled with a Gaussian distribution (sigma = 20 cm) in x and y directions (z direction is constant). The employee's position while working next to the machine is simulated in a 100–200 cm ring around the sensor position in a fully random manner. After completing 8 hours of work, the employee returns directly to the entry or exit position.

For the selected analysis clustering methods, the following values were defined for the required parameters:

1. DBSCAN:
   - Maximum distance between two samples: 75 cm,
   - Minimum number of neighboring samples: 100.
2. HDBSCAN:
   - Minimum number of neighboring samples: 800,
   - Minimum cluster size: 800.
3. OPTICS:
   - Maximum distance between neighbors: 75 cm,
   - Minimum number of neighboring samples: 100.

The same method of parameter tuning was used as for the ASTC method (described in Section 4.) The initial parameters were set by the expert, and then the optimal values were found by tuning the parameters, changing them by up to $\pm 5\%$ from the initial values.

The results of all four experiments, regarding each clustering method, are shown in Figure 2, below. To further investigate the observed clusters, they next served as input data for the binary classification task, the results of which are given in the next section.

6. Results

To compare the performance of the clustering methods, typical classification metrics were calculated, including:

- True Positive (TP),
- True Negative (TN),
- False Positive (FP),
- False Negative (FN),
- Specificity (TNR): TNR = TN / (FP+TN),
- Precision (P): P = TP / (TP+FP),
- Recall (R): R = TP / (TP+FN),
- F1 score (F1): F1 = 2*(P*R) / (P+R).

Table 1 shows the results of the tested metrics for the machine learning methods and the ASTC method. An efficiency of the extended path-based algorithm at the level of 99.5% was achieved for all simulated samples (50 employees). However, the results reveal small discrepancies between the methods. The ASTC method was the only one to show FNs, which indicates the underestimation of the number of points in the cluster.

On the other hand, the machine learning-based methods had significantly higher numbers of FPs with respect to the ASTC approach. Moreover, the DBSCAN, HDBSCAN and OPTICS methods were faultless in terms of the clustering of TPs. An instance of data clustering for one employee is presented in Figure 3.

In spite of its practical features, the original DBSCAN algorithm fails when the border objects of two clusters are relatively close. For spatial types of data, groups of objects are relatively well separated, but for other types of data, this is not always the case.

Our hardware configuration employed one desktop computer with an Intel i9 processor (3.6 GHz, 8 cores), 16 GB RAM, and Radeon Pro Vega 48 with 8 GB. The data processing duration for a sample of 50 employees was 8 seconds for DBSCAN, 14 for HDBSCAN, 170 for OPTICS and 33 for ASTC. Additionally, the memory usage was 0.5 GB, 3 GB, 3 GB and 30 MB, respectively. Nevertheless, if we take into account the factory settings as well as the users' expectations, the durations are acceptable since the overall system is not intended to guarantee a response within specified time constraints.

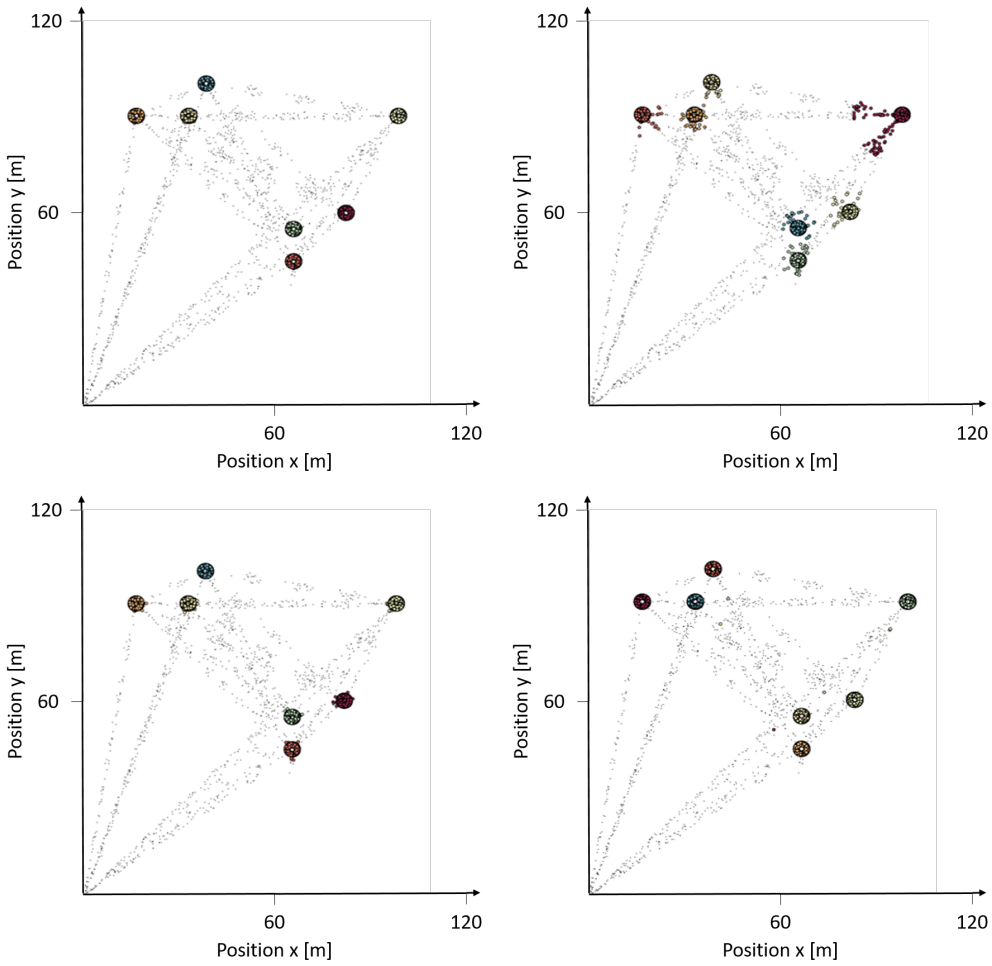

Figure 2. Clustering results for: DBSCAN (top left), HDBSCAN (top right), OPTICS (bottom left), and ASTC (bottom right)

Table 1. The results of the clustering methods performance

Metric/Algorithm	DBSCAN	HDBSCAN	OPTICS	ASTC
TP	142838	142838	142838	142718
TN	1474	1303	1464	1576
FP	109	280	119	7
FN	0	0	0	120
Specificity	0.931	0.823	0.925	0.996
Precision	0.999	0.998	0.999	1.000
Recall	1.000	1.000	1.000	0.999
F1 score	0.99962	0.99902	0.99958	0.99956

7. Discussion

Our work is especially valuable for managers for the following reasons. First, our method of grouping the localization data indicates those areas most frequented by workers during the workday. Such information allows a manager to estimate the total cost of production at a high level of accuracy, since the human labor costs are extracted from collected data. On the other hand, the individual awareness of having personal electronic sensors installed might stimulate them to be more productive.

Second, the workers' tracking system can be also considered to be a proactive approach to mitigate

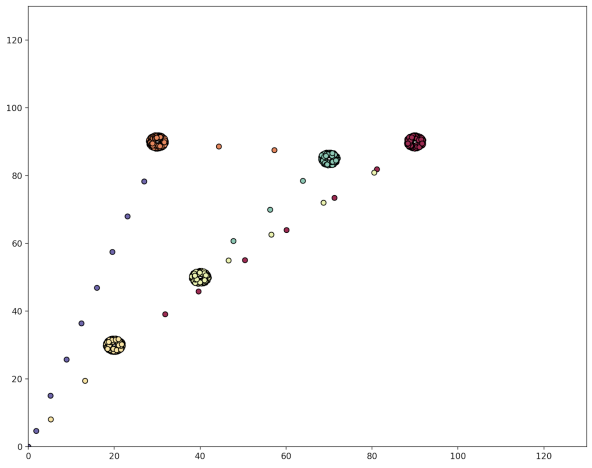

Figure 3. Clustering results for extended path-based algorithm for one employee.

workplace safety threats. For example, in the case of a fire, a manager can check the physical locations of all subordinate staff. If necessary, the proper authorities can be notified of their exact locations, allowing them to prepare and execute suitable evacuation plans. Using this line of thinking, if a worker is present at the beginning of a work period and then later cannot be found, a manager can quickly determine the worker's location and verify his or her present condition. Essentially, one can use the employee's position frequency maps (see Fig. 1) to aid other health, safety and working condition issues (e.g. workplace climate, lighting or noise).

Last but not least, it is worth noting here that the tracking system can be used for a purpose directly opposed to that assumed, such as outside the working area. Therefore, we are giving great consideration to implementing relevant privacy and security mechanisms to prevent such misuses.

8. Conclusions

The results from the performed experiments allow us to conclude that we still need to investigate better values for the parameters, regarding all four methods, during signal recording attempts at intervals differing from 10 seconds. The most intuitive algorithm for selecting the model parameters and giving the best result across the machine learning methods is DBSCAN.

A comparison of the False Negative and False Positive metrics showed that the algorithms using machine learning overestimate the number of points in the cluster, which is expected due to the use of the spatial aspect of the data only, without considering the time component.

It seems that for all methods based on spatial distribution for very large samples, the problem will be a too high density of the data, requiring a change of the input parameters for the models. However, it seems that the solution may be to divide the data into smaller time periods, with the number of samples adjusted to the pre-optimized input parameters. Then it will also be possible to parallelize the algorithm based on the time periods.

The next step is to add information about anomalies (e.g. an employee is in the neighborhood of a machine that is not working or the employee is in a prohibited place), which will allow the algorithms to be tested under conditions even closer to real working conditions. Furthermore, the future research will cover the comparison between improved DBSCAN-related algorithms [21, 23, 25] and our method, as well as parameter optimization based on the data collected from the actual factory unit. However, due to the presence of current and geolocation sensors for the machines, no significant decrease in algorithm accuracy is expected.

References

[1] Statista, "Employment cost index for all employees in the united states from 2010 to 2020, by quarter," 2020. Last accessed 16 April 2020.

[2] World Population Review, "World population review. minimum wage by country 2020," 2020. Last accessed 17 April 2020.

[3] Data USA, "Manufacturing. occupations," 2018. Last accessed 17 April 2020.

[4] D. Armbruster, K. Kaneko, and A. S. Mikhailov, *Networks of interacting machines: production organization in complex industrial systems and biological cells*, vol. 3. World Scientific, 2005.

[5] M. Swink, R. Narasimhan, and S. W. Kim, "Manufacturing practices and strategy integration: effects on cost efficiency, flexibility, and market-based performance," *Decision Sciences*, vol. 36, no. 3, pp. 427–457, 2005.

[6] M. L. Morris, N. Chowdhury, and C. Meisner, *Wheat production in Bangladesh: Technological, economic, and policy issues*, vol. 106. Intl Food Policy Res Inst, 1997.

[7] Investopedia, "Cost of labor," 2020. Last accessed 20 April 2020.

[8] S. Deng and T.-H. Yeh, "Using least squares support vector machines for the airframe structures manufacturing cost estimation," *International Journal of Production Economics*, vol. 131, no. 2, pp. 701–708, 2011.

[9] J. Portas and S. AbouRizk, "Neural network model for estimating construction productivity," *Journal of construction engineering and management*, vol. 123, no. 4, pp. 399–410, 1997.

[10] C.-C. Chou and P.-L. Chang, "Modeling and analysis of labor cost estimation for shipbuilding: The case of china shipbuilding corporation," *Journal of ship production*, vol. 17, no. 2, pp. 92–96, 2001.

[11] H. Nachtmann and K. L. Needy, "Methods for handling uncertainty in activity based costing systems," *The Engineering Economist*, vol. 48, no. 3, pp. 259–282, 2003.

[12] C.-I. Hsu, H.-H. Shih, and W.-C. Wang, "Applying rfid to reduce delay in import cargo customs clearance process," *Computers & Industrial Engineering*, vol. 57, no. 2, pp. 506–519, 2009.

[13] Y. M. Lee, F. Cheng, and Y. T. Leung, "Exploring the impact of rfid on supply chain dynamics," in *Proceedings of the 2004 Winter Simulation Conference, 2004.*, vol. 2, pp. 1145–1152, IEEE, 2004.

[14] H. Jiang, P. Lin, M. Qiang, and Q. Fan, "A labor consumption measurement system based on real-time tracking technology for dam construction site," *Automation in Construction*, vol. 52, pp. 1–15, 2015.

[15] M. Ianni, E. Masciari, G. M. Mazzeo, M. Mezzanzanica, and C. Zaniolo, "Fast and effective big data exploration by clustering," *Future Generation Computer Systems*, vol. 102, pp. 84–94, 2020.

[16] X. Cui, P. Zhu, X. Yang, K. Li, and C. Ji, "Optimized big data k-means clustering using mapreduce," *The Journal of Supercomputing*, vol. 70, no. 3, pp. 1249–1259, 2014.

[17] O. Kurasova, V. Marcinkevicius, V. Medvedev, A. Rapecka, and P. Stefanovic, "Strategies for big data clustering," in *2014 IEEE 26th international conference on tools with artificial intelligence*, pp. 740–747, IEEE, 2014.

[18] J. A. F. Costa and H. Yin, "Gradient-based som clustering and visualisation methods," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2010.

[19] M. Mousavi, A. A. Bakar, and M. Vakilian, "Data stream clustering algorithms: A review," *Int J Adv Soft Comput Appl*, vol. 7, no. 3, p. 13, 2015.

[20] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, pp. 226–231, 1996.

[21] T. N. Tran, K. Drab, and M. Daszykowski, "Revised dbscan algorithm to cluster data with dense adjacent clusters," *Chemometrics and Intelligent Laboratory Systems*, vol. 120, pp. 92–96, 2013.

[22] A. Karami and R. Johansson, "Choosing dbscan parameters automatically using differential evolution," *International Journal of Computer Applications*, vol. 91, no. 7, pp. 1–11, 2014.

[23] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[24] K. M. Kumar and A. R. M. Reddy, "A fast dbscan clustering algorithm by accelerating neighbor searching using groups method," *Pattern Recognition*, vol. 58, pp. 39–48, 2016.

[25] D. Birant and A. Kut, "St-dbscan: An algorithm for clustering spatial–temporal data," *Data & knowledge engineering*, vol. 60, no. 1, pp. 208–221, 2007.

[26] R. J. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Pacific-Asia conference on knowledge discovery and data mining*, pp. 160–172, Springer, 2013.

[27] R. J. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 1, pp. 1–51, 2015.

[28] R. L. Melvin, J. Xiao, R. C. Godwin, K. S. Berenhaut, and F. R. Salsbury Jr, "Visualizing correlated motion with hdbscan clustering," *Protein Science*, vol. 27, no. 1, pp. 62–75, 2018.

[29] A. C. A. Neto, J. Sander, R. J. Campello, and M. A. Nascimento, "Efficient computation of multiple density-based clustering hierarchies," in *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 991–996, IEEE, 2017.

[30] M. de Berg, A. Gunawan, and M. Roeloffzen, "Faster db-scan and hdb-scan in low-dimensional euclidean spaces," *arXiv preprint arXiv:1702.08607*, 2017.

[31] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.

[32] Z. Deng, Y. Hu, M. Zhu, X. Huang, and B. Du, "A scalable and fast optics for clustering trajectory big data," *Cluster Computing*, vol. 18, no. 2, pp. 549–562, 2015.

[33] S. T. Mai, I. Assent, and A. Le, "Anytime optics: An efficient approach for hierarchical density-based clustering," in *International Conference on Database Systems for Advanced Applications*, pp. 164–179, Springer, 2016.

[34] McGraw-Hill Dictionary of Scientific & Technical Terms, 6th Edition, "Productive time," 2003. Last accessed 12 July 2020.

[35] J. Blank and K. Deb, "pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89497–89509, 2020.

[36] F. Amalina, I. A. T. Hashem, Z. H. Azizul, A. T. Fong, A. Firdaus, M. Imran, and N. B. Anuar, "Blending big data analytics: Review on challenges and a recent study," *IEEE Access*, vol. 8, pp. 3629–3645, 2019.