Determining Case Identifiers in Event Logs from Hoists in Construction Sites to Estimate the Correlation between Usage Rates and BPMNs

ERJAN STEENBERGEN, University of Twente, The Netherlands

The use of process mining is rapidly growing. Even so, not every business has its data log infrastructure ready for process mining. One of the three main components of discovering a process model is a case identifier, which groups events into a sequence. It is often the case that businesses do not have case identifiers to group events in their event log. This is also the case for a hoist rental company, whose goal is to determine the usage rate of their hosts from their event data. If there is a correlation between the usage rates and the process-mined models, this could increase the efficiency of conformance checking. This research investigates if there is a correlation between a hoist's usage rate and its process models by creating cases for an event log that is devoid of case identifiers.

Additional Key Words and Phrases: Process Mining, Hoists, Usage Rate, Case Identifiers, Case Extraction, BPMN

1 INTRODUCTION

Businesses constantly eye on new opportunities to maximise efficiency for their processes. Every system in a business is usually designed beforehand into a coherent model. However, these models do not always comply with the events that occur in real life. This is where Process Mining (PM) is introduced. PM bridges the gap between data mining and business process management by allowing the creation of a process model from an event log of that system. [23] This grants the ability to discover inefficiencies in processes.

A model in PM consists of several types of data fields that bundle up into an event log. These types include but are not limited to, *activities*, *timestamps*, and *case identifiers* (case IDs). An activity is the name of an event that occurs in a process. A timestamp indicates the start or end of an activity. A case ID is a unique identifier that belongs to a certain instance of a process. [21] These three data types are necessary for process discovery, which is a means to translate event data into a model, such as Petri nets, process trees, and BPMNs, where the latter will be primarily used in this research. It is generally a very useful tool to have for your business. [24]

Even though PM is rapidly growing, many businesses have not yet implemented these techniques, which makes it difficult to apply PM to their event data. arious process discovery algorithms already exist, [10, 13, 26, 27, 29] but these algorithms only tailor to data which is perfectly orchestrated for PM, which makes them irrelevant in cases where businesses do not have their data infrastructure ready for PM. An example of this is grouping events to a case identifier, which has been a problem as long as PM exists. [6, 9] The PM community calls this problem the *event correlation* problem. [2, 7, 16]

This problem also occurs in hoist rental company X (name kept anonymous). X hires out hoists, which refer to a machine that vertically transports passengers or freight on the side of a building that is in construction, primarily for construction companies to accompany their workers and equipment on construction sites. At this instant, usage rates cannot be shown from their event data. Allowing to measure usage rates could be useful for predicting maintenance, as well as being able to predict the number of hoists needed for a job. The usage rate is referred to as the percentage of time the lift is active. [22] As there are no case identifiers determined in the data, the utilisation of process mining directly on the given event logs is impractical. This topic of the event correlation problem has not been touched upon in terms of hoist activity and usage rate, therefore it is an interesting topic to discuss. Also, it is possible within the scope of this project to find a solution to integrate PM into a hoist/lift business where the data infrastructure is not viable. This project could find a method to more easily integrate PM into such a business without having to completely change one's data infrastructure.

Conformance checking (CC) is an analysis between a process model and its respective event log to determine whether the process model demonstrates the intended behaviour of the event log. [4, 23] CC takes a considerable amount of time when practising PM, as it is one of the three main components of PM. [8] If it is possible to determine the process model from a value that is simply taken from the event log, it should quicken the process of CC. This is especially helpful when a business starts to integrate PM into its data structure and check for valuable case IDs. In the context of this research, the usage rate will be the value to find a correlation between it and the model. When there is a certain correlation between the usage rate and the process model, it is possible to already determine the shape of the model, by comparing process models from cases that have an equal usage rate. To summarise the hypothesis: if an event log with a certain case and process model exists and there is the same event log with a different case and a similar usage rate, does this process model always look similar to the former?

1.1 Goals

X's ultimate goal for this project is to find a method to discover the usage rate of X's hoists from their existing data log infrastructure. This is to be able to predict the number of hoists needed for a construction site and when maintenance needs to be scheduled. Additionally, X's data retrieval does not account for PM techniques, as there are no clear case IDs indicated in their data. To expand this research, it is possible to try to extract case IDs from the already existing event logs and analyse their effectiveness by calculating the usage rate of the newly generated event logs that have self-determined cases. The discovered models and the usage rates of each case will then determine if there is a correlation between the two metrics. This can be summarised in two goals: The first goal is to discover the usage rate of a hoist (The amount of time that the hoist is in use) from the existing data that is retrieved by the respective

TScIT 41, July 5, 2024, Enschede, The Netherlands

hoist. Secondly, the goal is to generate and discover process models with self-determined case IDs and compare them with each other to determine whether it is possible to determine the shape of the models from the usage rate. These goals can be outlined in a research question in this fashion:

Is there any correlation between the estimated usage rate and the process model taken from an event log of a hoist in a construction site?

This research question can then be segregated into multiple sub-questions:

- What can be said about the difference in usage rate between an estimation directly taken from a dataset and an estimation derived from discovered process mining models?
- What kind of case identifiers result in adequate process mined models on data from hoists on construction sites, such that the usage rate can be compared to the usage rate benchmark that is directly taken from a dataset?
- How can an accurate benchmark for usage rate be determined?
- How is a case defined from a data log with no clearly indicated case IDs?
- How is a strong correlation between the usage rate and the shape of the process model determined?

2 RELATED WORK

There have been numerous works that have researched methods to discover case identifiers from datasets where they are not visible or devoid of them. Therefore it shows that correlating events is not that obvious to determine. There are multiple choices for case IDs, where each case can yield very different results. [7, 12] The most common method is to find sequences that fit into workflow patterns. [25] When case IDs are not present, an event log is just a single line of events. By deducing patterns from the data, it is possible to create full logical sequences. [9, 19]

Below are a few examples of works that have invented new ways to discover processes without any case identifiers.

One of the process discovery algorithms that address the correlation challenge is called the *correlation miner*. This algorithm discovers correlations between events without a case identifier by creating 2 matrices, a *Precede/Succeed matrix* and a *Duration matrix*. These matrices then determine the correlation between the two events. [18] Another solution, found by Ferreria and Gillbad, talks about using an iterative Expectation-Maximization procedure to find case IDs in unlabelled event logs.[9]

Pegoraro uses a *word2vec* (A natural language processing technique) neural model to form cases from a transition model of a whole process. [16] This is achieved by segmenting an execution log from the model into well-formed cases. The data that is used is from click data from a smartphone app.

Walicki and Ferreira [28] extensively note the problem of finding case IDs in unlabelled event logs. Their approach to solving this problem is by using sequence partitioning to find minimal sets of patterns contained in a sequence. This is done by building a *trie* to do a complete search on sets of patterns and then generating a list of possible solutions to those patterns.

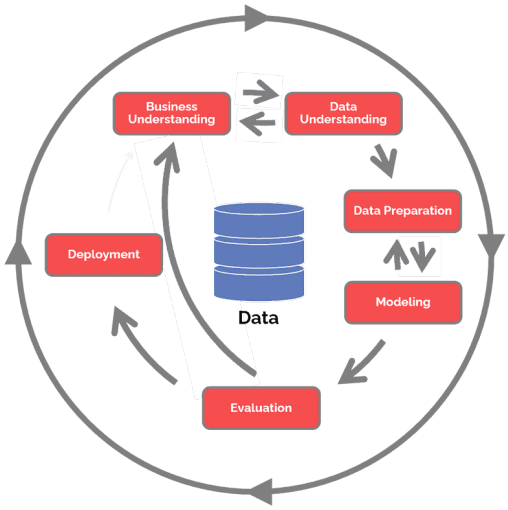

Fig. 1. Process of a CRISP DM approach

Pieters and Schlobach [17] have used PM, together with time series forecasting to predict bed usage in hospitals. Even though the topic of this is in a different working environment, it is possible to translate their methods in a way that is also useful for this branch. For example, Pieters and Schlobach use PM to determine the patient flow of a hospital. To put this into context for a construction hoist, the *patient flow* could be translated into a *passenger flow*, to determine the activity of passengers in a hoist.

3 METHODOLOGY AND APPROACH

To answer the research questions, it is important to adopt a clear approach to the problems. This research will be done by following the *CRISP DM* process model. The reason for this is that it is the most used methodology for data mining (DM) and PM. It is a process that naturally describes a DM/PM cycle by inspecting the next 6 subsections. [20]

3.1 Business Understanding

This subsection talks about the goals of this research. This is primarily discussed in 1.1. The objectives are to discover case identifiers from the event log and then determine the usage rate of hoists in construction sites with the newly case ID-labelled event log and to find whether there is a correlation between the usage rate and the shape of the process models.

3.2 Data Understanding

A significant challenge of this research is the absence of case IDs in the event data received. This project uses a data log made up of approximately 3 million events from around 180 signals, spanning over one month (October 2023). There is an abundance of sensed data acquired from the hoists, ranging from the current floor of the hoist to Bluetooth activity. Each sensor data should be verified by quality in terms of relevance to this project. This can be seen in Table 1, which shows the signals that have been primarily used in

Table 1. Signals from the hoist that are used in this research

Signal	Description
CallBufferDownSent	Floor from which the hoist is called to go down
CallBufferUpSent	Floor from which the hoist is called to go up
CallBufferCabinSent	Destination floor pressed in hoist
ActualFloor	Current floor of hoist
DoorsOpen	Doors of hoist open
DoorsClose	Doors of hoist close
ActiveTime	Time when the hoist is active
RidingTime	Time when the hoist is in motion
PeopleRidingTime	Time between each point of *CallBufferCabinSent*
WaitingTime	Time need to arrive to floor call
ActiveDays	Check whether the hoist has been active that day

this research. The relevance of the signals is based on their ability to describe usage rate and/or the capability for creating cases.

3.3 Data Preparation

In this portion of the research, irrelevant sensor data will be excluded, while the leftover data will be cleaned by potentially correcting or removing erroneous values, and replacing the *signal ids*, that correlate to a certain sensor, to the respective sensor name, as they are mapped one-to-one, to make the dataset more readable. Then, a benchmark for the usage rate will be estimated of the data without using PM to be used as a control value. Since the data received from the hoists includes a sensor that keeps track of the time that the hoist is in use, it is possible to derive usage rate from this sensor. After this step, the data will be extensively researched to find patterns that could be made into case IDs. Some examples of cases could be:

- All the activities of one hoist being one case id
- All the activities of one day being one case id
- An estimation of a person using the hoist [1]
- Every activity being a separate case id
- Every interval of the hoist's activity is one case id

3.4 Modelling

This phase will be run roughly parallel to the previous data preparation phase. When some case IDs have been constructed, it is important to check whether a coherent process model will be discovered. If this is the case, then it will be used for evaluation. If not, it will be discarded. The process discovery software *Apromore* will be used to better visualise the event logs. Python library *pm4py* is used to calculate the usage rate from the case id-labelled data. [3] The average case duration is taken as the active time of the hoist.

[1]An example of this would be: Person presses button for lift to come to floor -> person steps in lift and presses button to go to floor -> hoist reaches floor and person steps out of hoist

3.5 Evaluation

After discovering the models by using the case IDs, the outcome of the usage rate from the model will be compared with each other and with the benchmark created at the start of the research. The models are compared by using the software *BPMNDiffViz*. [11] This program uses Graph-Edit-Distance (GED) to determine the similarity between two graphs. However, its GED score will not be utilised for this research, instead, a percentage of matched elements between the two BPMNs is applied, which is named the 'matching rate' in this project. To elaborate, the GED score is based on the sum of the amount of distinct edges and vertices. When the score is higher, there is a bigger variance between the two graphs. This score favours the comparison between smaller graphs, hence the matching rate is used, which holds a more suitable equality between the graphs. The matching rate is ultimately meant to compare with the difference in usage rate from both cases. The closer the rates are to each other, the stronger the correlation between the two variables. The BPMNs are taken from Apromore, all with an arc of 50 and a parallelism value of 40. These values are chosen based on keeping relevant relations and removing weak relations to ensure detailed models, but removing edge cases that could contaminate the results.

3.6 Deployment

Finally, when all the variables are compared, a conclusion can be made about whether there is a strong correlation between the usage rate and the shape of the BPMN. Company *X* will also be given the research results. They will be able to use the usage rate in their data, together with using PM with the cases that are described in this paper.

4 DATA PREPARATION & MODELLING

4.1 Finding Benchmark

The benchmark is determined by taking the data from one sensor; *ActiveTime*, which keeps track of how many minutes the hoist has been consequently in use (referred to as 'active'), to determine the amount of minutes the hoist was active. The hoist is 'active' when the hoist is moving, or if passengers or goods are loading in or out of the hoist. The value is then divided by the total work hours spanning the dataset. The formula for determining the usage rate resembles this:

$$U = (T_a/T_{tot}) * 100$$

It divides the amount of time the hoist was active, denoted by T_a by the total time the hoist is in operation, denoted by T_{tot}. In this context, the total time equals the amount of work hours the hoist has operated during the event log's time. The benchmark of the event log resulted in a usage rate of 57,19%.

4.2 Finding Case Identifiers

This section explains all of the cases that are used in this research. These cases all showed a moderate to highly accurate representation of the event log during the CC of the process models.

4.2.1 ActiveTime. It is possible to group activities based on an interval every time the lift is active. This model is shown in Figure 2. You can see that a usual sequence follows a pattern of the hoist doors opening (DoorsOpen), a floor button being pressed inside the hoist (CallBufferCabinSent), doors closing (DoorsClose), and then the hoist going to the respective floor and ending unless another person has pressed a button that brings the lift to their floor (CallBuffer(Up/Down)Sent)

4.2.2 Person. The simulation of a person's activity is determined as follows: A button is pressed on a specific floor, which brings the hoist to that floor (CallBuffer(Up/Down)Sent), the hoist reaches this floor and opens the door (DoorsOpen), the person presses a floor button in the hoist (CallBufferCabinSent), the doors close (DoorsClose), the hoist reaches the floor and opens the door (DoorsOpen), the person leaves and the door closes (DoorsClose).

However, this leads to several challenges. Firstly, during the sequence, other people can enter the lift, causing most cases to have multiple instances of buttons pressed in the hoist and the door opening and closing. This is pictured by the process model discovered from this case, shown in Figure 3. Additionally, an event can be grouped into multiple case IDs since there are multiple passengers. The solution to this gives each sequence a version of this event. The event will thus be grouped in every 'person id' that is active at that time.

4.2.3 Day. This case groups events into separate days. Therefore, one day equals one case identifier.

4.2.4 RidingTime. Another interval that is given by the sensors is the *RidingTime*, which gives the amount of time (in seconds) that the lift is in motion, from one destination floor to the other. This is different than *ActiveTime*, since this case does not include the time when the hoist doors are opened, thus creating more cases with lower times.

4.2.5 RidingWaitingTime & ConcRidingWaiting. A sequence of *RidingWaitingTime* starts when a *CallBuffer(Up/Down)Sent* gets called. It also includes the subsequent *WaitingTime* and *RidingTime*, which indicate the time of a passenger waiting for the hoist and the time between two subsequent times the hoist initiated its brakes. However, when two *CallBuffer(Up/Down)Sent* signals get sent shortly after each other, the first case gets cut off and will terminate prematurely. This is not the case for *ConcRidingWaiting*. It follows the same pattern as *RidingWaitingTime*, however, it keeps track of alive cases and only stops the sequence when the *RidingTime* is over.

4.2.6 PeopleRidingTime. The model of *PeopleRidingTime* groups cases into the interval determined by the signal *PeopleRidingTime*, which calculates the time between each point of *CallBufferCabinSent*.

4.3 Concurrency between cases

A challenge during the estimation of usage rates was the introduction of concurrency between cases, which was apparent in cases that simulate a process for a certain person. This is very difficult to do, as several users can use the lift at once, which means that some events from the hoist have to be grouped in more than 1 case ID. This is possible to do by simply duplicating the events, but this will

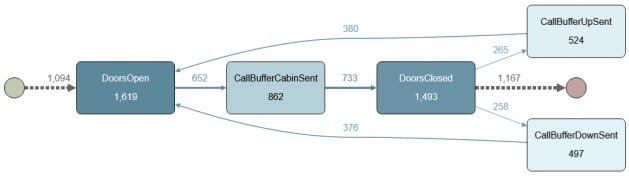

Fig. 2. Process Model if case id is based on every instance that the hoist is active

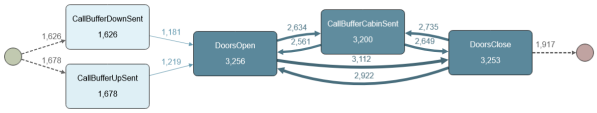

Fig. 3. Process Model if case id is simulated as a person

most likely lead to an inaccurate representation of case variants, activity instances, and, most importantly, the total time duration of events. However, this can be solved by using *Little's Law*, which states that the average number of people in a system is equal to the arrival rate multiplied by the time spent in the system ($L = \lambda W$). [1] If it is possible to calculate the average amount of people using the hoist in a certain time interval, it should be possible to attain an accurate usage rate.

$$U_{pm} = \frac{T_a/T_{tot}}{\lambda W} * 100$$

This version should be used when some events are done in parallel to represent the usage rate accurately. The cases that use this are *Person* and *ConcRidingWaiting*.

5 EVALUATION

5.1 Comparing Usage Rate

In this section, the usage rates of all the event logs are compared and explanations of the results are given. In Table 2, the case ID's usage rate taken from its model and the difference between the benchmark are shown. In Table 3, the difference in usage rate is shown between all cases. Some names of cases have been abbreviated to ensure the compactness of the table. These names include: *RidingTime* (RT), *PeopleRidingTime* (PRT), *RidingWaitingTime* (RWT), and *ConcRidingWaiting* (CRW).

The *Day* case results in a very large usage rate of 98%. This is quite logical since it takes the interval between the first signal of the day and the last signal of the day, which spans almost the entirety of the work hours. In contrast, *RidingTime* and *PeopleRidingTime* have a low usage rate, as they do not account for the time when the hoist is loading in passengers. Although this model is not a good representative for estimating usage rate, it does prove the fact that the hoist takes a considerable amount of time waiting for passengers to load in and out of the hoist, which is estimated to be around 36%.

Table 2. The usage rates and difference from the benchmark of all tested Case Identifiers

Case	Usage Rate	Δ Benchmark
Benchmark	57,19%	0,00%
ActiveTime	56,50%	0,69%
ConcRidingWaiting	52,38%	4,81%
Person	63,38%	6,19%
RidingWaitingTime	66,55%	11,36%
PeopleRidingTime	34,96%	22,23%
Day	98,08%	40,89%
RidingTime	12,66%	44,53%

Table 3. Usage Rate Difference between all cases in %

Case	RT	Day	PRT	RWT	Person	CRW
ActiveTime	43,84	41,58	21,54	10,05	6,88	4,12
ConcRidingWaiting	39,72	45,7	17,42	14,17	14	
Person	50,72	34,7	28,42	3,17		
RidingWaitingTime	53,89	31,53	31,59			
PeopleRidingTime	22,3	63,12				
Day	85,42					

Table 4. Matching rate between two models in %

Case	RT	Day	PRT	RWT	Person	CRW
ActiveTime	54	52	59	57	48	54
ConcRidingWaiting	34	60	65	94	86	
Person	38	60	58	86		
RidingWaitingTime	44	60	61			
PeopleRidingTime	47	65				
Day	47					

ConcRidingWaiting is more accurate than its non-concurrent sibling *RidingWaitingTime*, with a difference of 6,55% to the benchmark. In general, the concurrent cases are relatively very accurate compared to the non-concurrent cases, except for the *ActiveTime* case.

The benchmark is determined from the signal *ActiveTime*. It does then make sense that the process model influenced by *ActiveTime* would be accurate, as it takes the same intervals of activity. Consequently, the difference in percentage from the benchmark ultimately concludes which case has an average case duration that is most representative of the signal *ActiveTime*.

5.2 Comparing Models

When simulating the cases on the hoist or the passengers, both cases result in relatively accurate usage rates. From the naked eye, it seems like they show a slight difference in their process models (See Figure 2 and 3). The models are further compared using *BPMNDiff Viz* to ensure more systematic and robust testing. The program shows the amount of matched elements between the models, called the *matching rate*. The percentage of matched elements between all models is plotted in Table 4.

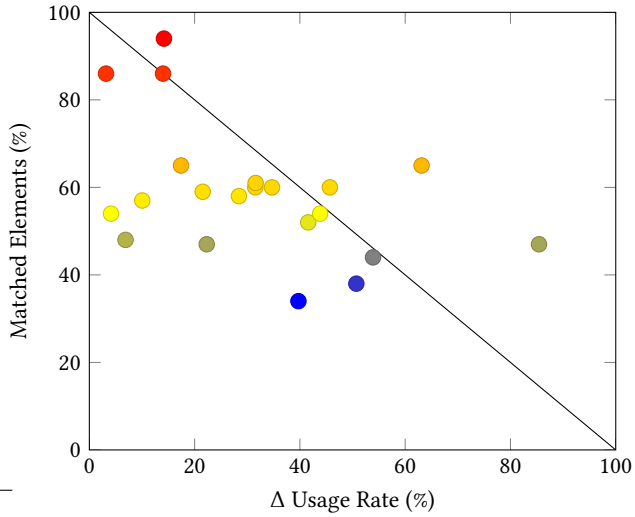

Fig. 4. Usage rate and matching rate correlation of case pairs

It shows that *RidingWaitingTime*, *ConcRidingWaiting*, and Person all show a high matching rate to each other, having an average matching rate of approximately 89%. This can be explained, as all three simulate the process of a passenger in some way, causing their models to be comparable. This group of cases will be named the "passenger cases". Interestingly, the case *PeopleRidingTime* does not show a high matching rate against the previously mentioned cases, despite its name. This is most likely because, as explained in point 4.2, *PeopleRidingTime* does not account for the time a passenger is waiting for the hoist to reach their floor.

The rest of the cases do not show any significant similarity between each other, with most values being within 40-60%, reaching at most a 65% matching rate, bringing both the average and median of the percentage of matched elements to roughly 58%. The case RidingTime scores exceptionally low, with an average matching rate of 44%. This implies that its BPMN is the least similar to the rest of the cases. The theory behind this could be that, since *RidingTime* also has a very low usage rate, it does not pick up most edge activities that most other cases do pick up, causing start and stop events to be different.

The measurements of all model pairs are plotted in Figure 4. A pair that has an equal matching rate and difference in usage rate shows that the usage rate can be correlated to the model shape. This is indicated by the black line in Figure 4. The closer a point is to this line, the more likely it is that the usage rate and model shape are correlated to each other. The 'distance' between a point and the black line is shown in Table 5. The passenger cases can be seen as red dots in Figure 4, showing that they are quite an outlier in terms of matching rate.

It is possible to determine the correlation between the two variables by using the *Pearson Correlation Coefficient*: [15]

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

Table 5. Accuracy of the difference in usage rate and matched elements in %

Case	RT	Day	PRT	RWT	Person	CRW
ActiveTime	2,16	6,42	19,46	32,95	45,12	41,88
ConcRidingWaiting	26,28	5,70	17,58	8,17	0	
Person	11,28	5,30	13,58	10,83		
RidingWaitingTime	2,11	8,47	7,41			
PeopleRidingTime	30,70	28,12				
Day	32,42					

This results in a Pearson coefficient of $r = -0,45$, which does not show enough evidence to accept the hypothesis that there is a strong correlation between the usage rate of the hoist, taken from the event log, and the BPMN of the event log.

6 DISCUSSION AND FUTURE WORK

With this event log, the representation of passengers in the hoist cannot be estimated with 100% accuracy. It cannot be stated how many passengers are in the hoist. Situations exist where, for example, two passengers step into the hoist together and have to access the same floor. The cases count these passengers as only one passenger. Determining this from the weight is an option, though this does not account for goods that are brought into the hoist. Given that these are hoists from construction sites, this is not a trivial point and should be looked upon when primarily focusing on the process models. For the usage rate, however, The inaccuracy is negligible, as due to *Little's law*, the ratio stays roughly the same.

The limited cases in this paper are self-determined. While it is suspected that the most relevant cases have been found in this study, the possibility should not be excluded that there may be more cases that could further be relevant to this research that have not been formed.

The usage rate acquired in this paper is not a completely controlled value. Frankly, the control value (named benchmark) was not a given value, but something that had to be determined during the paper itself, where there were no means to check its accuracy. There was limited to no related work about an accurate formula describing the usage rate of a hoist or lift, where most work was based only on assessing maintenance. [5, 14]

There is another version of the benchmark, which includes the rated load of the hoist as an additional modifier, which looks like this:

$$U_w = (1 + \frac{W_{avg}}{W_{tot}} * 0,5) * (\frac{T_a}{T_{tot}}) * 100$$

This version was more liked by the stakeholders but is not used in this paper, as the added modifier always stays the same across the cases, therefore only a little change will be seen. Additionally, it simplifies the process of calculating the usage rates with *pm4py* drastically. Since this paper has a deadline, this weighted variant was excluded. Nonetheless, this weighted version is still forwarded to the stakeholders for future use.

The approach to estimating the correlation between BPMNs using a 'matching rate' is a self-determined measurement. It has not yet been commonly proven to be an effective method in the PM domain. Therefore, it cannot be said with complete certainty that the method is academically accurate. However, the matching rate is a spur from the GED score, which is a commonly used method in PM.

This study is tested on one event log. To produce a more definite conclusion, the correlation of the two variables should have been measured between multiple event logs of several hoists. In the future, the comparison between the usage rate of hoists can be a follow-up question to the research from this paper. This subject could help in determining occupancy rates for a complete construction building.

Additionally, it is possible to go further in-depth in determining the usage rate of the hoist. For example, a digital twin could be made of the hoist, simulating and calculating possible times of maintenance, correlating to the usage rate. With this, hoist businesses will be able to not only predict maintenance but also deduce what the perfect value of usage rate is for the hoist. Then, something can also be said about the number of hoists needed for construction.

Another possibility for future research is to identify a variable that is applicable for most if not all, event logs across businesses and try to ascertain a correlation between that variable and the respective process models.

7 CONCLUSION

In this paper, the question stated if there is a correlation between the hoist's usage rate and the event log's BPMN. The first phase of the research was to find and elect case identifiers fit for the event log. This was done by analysing the signals from the dataset and transforming certain intervals into groups, which resulted in case IDs.

As concluded, the case *ActiveTime* shows the most accurate usage rate compared to the benchmark. This is most likely since the benchmark is derived from the same signal that *ActiveTime* is also derived from. It can be confirmed that cases that do not possess a usage rate value comparable to the benchmark, carry a case duration that contradicts the *ActiveTime* case duration. Furthermore, the concurrent cases *ConcRidingWaiting* and *Person* scored relatively well in usage rate compared to the benchmark.

The usage rate that is applied is established by researching common methods and adopting a trial-and-error style approach to find a suitable formula. To determine the correlation between usage rate and process models, a simplistic formula was constructed, as this ensured a more robust approach to determining the correlation between these two variables. A distinct formula has been presented for the company *X*, which includes the loaded weight of the hoist, which will be applied by them.

Most case pairs have a matching rate between 40 and 65%, where there were a few outliers, including the passenger cases, which have a matching rate above 85% between each other. In essence, the majority of the cases include the same activities but have a slightly different sequence, which leads to a matching rate of around 50%.

From the results, it is concluded that there is not enough evidence to show that there is a strong correlation between the two variables. If cases have a similar estimation of the hoist's usage rate, it does not mean that the BPMN created from the event log are identical to each other. An example of this can be seen between Figure 2 and 3.

The difference in usage rate is 6,88%, which is, compared to other cases, relatively low. However, the two models are visibly different.

This is furthermore stated by the graph shown in Figure 4. The case pairs do not follow the line of correlation. Plus, according to the Pearson coefficient, there is only a moderate to low correlation between the difference in usage rate and the matching rate between the cases.

8 ACKNOWLEDGEMENTS

I would like to thank my supervisor Faiza Bukhsh for guiding me through my research and giving invaluable feedback. I would also like to thank my supervisor Rob Bemthuis for teaching me all the technical aspects of process mining and taking the time to brainstorm with me. This work has received support from the ECOLOGIC project, which was funded by the Dutch Ministry of Infrastructure and Water Management and TKI Dinalog (case no. 31192090). I would like to thank the stakeholders in this research for supporting me through this project.

REFERENCES

[1] Arnold O. Allen. 1990. *Probability, Statistics, and Queuing Theory (Second Edition).*

[2] Dina Bayomie, Claudio Di Ciccio, Marcello La Rosa, and Jan Mendling. 2019. A Probabilistic Approach to Event-Case Correlation for Process Mining. In *Conceptual Modeling*, Alberto H. F. Laender, Barbara Pernici, Ee-Peng Lim, and José Palazzo M. De Oliveira (Eds.). Vol. 11788. Springer International Publishing, Cham, 136–152. https://doi.org/10.1007/978-3-030-33223-5_12 Series Title: Lecture Notes in Computer Science.

[3] Alessandro Berti, Sebastiaan Van Zelst, and Daniel Schuster. 2023. PM4Py: A process mining library for Python. *Software Impacts* 17 (Sept. 2023), 100556. https://doi.org/10.1016/j.simpa.2023.100556

[4] Josep Carmona, Boudewijn Van Dongen, Andreas Solti, and Matthias Weidlich. 2018. Introduction to Conformance Checking. In *Conformance Checking*. Springer International Publishing, Cham, 3–20. https://doi.org/10.1007/978-3-319-99414-7_1

[5] Barrie Chanter and Peter Swallow. 2007. *Building maintenance management* (2nd ed ed.). Blackwell, Oxford ; Malden, MA. OCLC: ocm76167435.

[6] Roberta De Fazio, Antonio Balzanella, Stefano Marrone, Fiammetta Marulli, Laura Verde, Vincenzo Reccia, and Paolo Valletta. 2024. CaseID Detection for Process Mining: A Heuristic-Based Methodology. In *Process Mining Workshops*, Johannes De Smedt and Pnina Soffer (Eds.). Springer Nature Switzerland, Cham, 45–57. https://doi.org/10.1007/978-3-031-56107-8_4

[7] Jochen De Weerdt and Moe Thandar Wynn. 2022. Foundations of Process Event Data. In *Process Mining Handbook*, Wil M. P. Van Der Aalst and Josep Carmona (Eds.). Vol. 448. Springer International Publishing, Cham, 193–211. https://doi.org/10.1007/978-3-031-08848-3_6 Series Title: Lecture Notes in Business Information Processing.

[8] Sebastian Dunzer, Matthias Stierle, Martin Matzner, and Stephan Baier. 2019. Conformance checking: a state-of-the-art literature review. In *Proceedings of the 11th International Conference on Subject-Oriented Business Process Management*. ACM, Seville Spain, 1–10. https://doi.org/10.1145/3329007.3329014

[9] Diogo R. Ferreira and Daniel Gillblad. 2009. Discovering Process Models from Unlabelled Event Logs. In *Business Process Management*, Umeshwar Dayal, Johann Eder, Jana Koehler, and Hajo A. Reijers (Eds.). Springer, Berlin, Heidelberg, 143–158. https://doi.org/10.1007/978-3-642-03848-8_11

[10] Christian W. Günther and Wil M. P. Van Der Aalst. 2007. Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics. In *Business Process Management*, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Gustavo Alonso, Peter Dadam, and Michael Rosemann (Eds.). Vol. 4714. Springer Berlin Heidelberg, Berlin, Heidelberg, 328–343. https://doi.org/10.1007/978-3-540-75183-0_24 Series Title: Lecture Notes in Computer Science.

[11] S. Ivanov, Anna Kalenkova, and Wil M. P. Van Der Aalst. 2015. BPMNDiffVizz: A Tool for BPMN Models Comparison. (2015). https://www.semanticscholar.org/paper/BPMNDiffViz%3A-A-Tool-for-BPMN-Models-Comparison-Ivanov-Kalenkova/2d2b9a39234c673db77fb694319e7e7049e5625b

[12] Agnes Koschmider, Milda Aleknonytė-Resch, Frederik Fonger, Christian Imenkamp, Arvid Lepsien, Kaan Apaydin, Maximilian Harms, Dominik Janssen, Dominic Langhammer, Tobias Ziolkowski, and Yorck Zisgen. 2023. Process Mining for Unstructured Data: Challenges and Research Directions. http://arxiv.org/abs/2401.13677 arXiv:2401.13677 [cs].

[13] Sander J. J. Leemans, Dirk Fahland, and Wil M. P. Van Der Aalst. 2013. Discovering Block-Structured Process Models from Event Logs - A Constructive Approach. In *Application and Theory of Petri Nets and Concurrency*, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, José-Manuel Colom, and Jörg Desel (Eds.). Vol. 7927. Springer Berlin Heidelberg, Berlin, Heidelberg, 311–329. https://doi.org/10.1007/978-3-642-38697-8_17 Series Title: Lecture Notes in Computer Science.

[14] Roger T.H. Ng, Joseph H.K. Lai, Oscar C.H. Leung, and David J. Edwards. 2023. Assessing lift maintenance performance of high-rise residential buildings. *Journal of Building Engineering* 68 (June 2023), 106202. https://doi.org/10.1016/j.jobe.2023.106202

[15] Karl Pearson. 1895. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London Series I* 58 (Jan. 1895), 240–242. https://ui.adsabs.harvard.edu/abs/1895RSPS...58..240P ADS Bibcode: 1895RSPS...58..240P.

[16] Marco Pegoraro, Merih Seran Uysal, Tom-Hendrik Hülsmann, and Wil M. P. van der Aalst. 2022. Resolving Uncertain Case Identifiers in Interaction Logs: A User Study. https://doi.org/10.48550/arXiv.2212.00009 arXiv:2212.00009 [cs].

[17] Annelore Jellemijn Pieters and Stefan Schlobach. 2022. Combining Process Mining and Time Series Forecasting to Predict Hospital Bed Occupancy. In *Health Information Science*, Agma Traina, Hua Wang, Yong Zhang, Siuly Siuly, Rui Zhou, and Lu Chen (Eds.). Vol. 13705. Springer Nature Switzerland, Cham, 76–87. https://doi.org/10.1007/978-3-031-20627-6_8 Series Title: Lecture Notes in Computer Science.

[18] Shaya Pourmirza, Remco Dijkman, and Paul Grefen. 2015. Correlation Mining: Mining Process Orchestrations Without Case Identifiers. In *Service-Oriented Computing*, Alistair Barros, Daniela Grigori, Nanjangud C. Narendra, and Hoa Khanh Dam (Eds.). Springer, Berlin, Heidelberg, 237–252. https://doi.org/10.1007/978-3-662-48616-0_15

[19] R. P. Jagadeesh Chandra Bose, R. P. Jagadeesh Chandra Bose, Wil M. P. van der Aalst, and Wil M. P. van der Aalst. 2009. Abstractions in Process Mining: A Taxonomy of Patterns. 5701 (Aug. 2009), 159–175. https://doi.org/10.1007/978-3-642-03848-8_12 MAG ID: 1824264389.

[20] Christoph Schröer, Felix Kruse, and Jorge Marx Gómez. 2021. A Systematic Literature Review on Applying CRISP-DM Process Model. *Procedia Computer Science* 181 (2021), 526–534. https://doi.org/10.1016/j.procs.2021.01.199

[21] Arthur H. M. Ter Hofstede, Agnes Koschmider, Andrea Marrella, Robert Andrews, Dominik A. Fischer, Sareh Sadeghianasl, Moe Thandar Wynn, Marco Comuzzi, Jochen De Weerdt, Kanika Goel, Niels Martin, and Pnina Soffer. 2023. Process-Data Quality: The True Frontier of Process Mining. *Journal of Data and Information Quality* 15, 3 (Sept. 2023), 1–21. https://doi.org/10.1145/3613247

[22] TKElevator. 2021. ELEVATOR PERFORMANCE AND DUTY APPLICATION GUIDELINE.

[23] Wil M. P. Van Der Aalst. 2011. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-19345-3

[24] Wil M. P. Van Der Aalst. 2016. *Process Mining*. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-49851-4

[25] Wil M. P. Van Der Aalst, A.H.M. Ter Hofstede, B. Kiepuszewski, and A.P. Barros. 2003. Workflow Patterns. *Distributed and Parallel Databases* 14, 1 (2003), 5–51. https://doi.org/10.1023/A:1022883727209

[26] Wil M. P. Van Der Aalst, T. Weijters, and L. Maruster. 2004. Workflow mining: discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering* 16, 9 (Sept. 2004), 1128–1142. https://doi.org/10.1109/TKDE.2004.47

[27] J. M. E. M. Van Der Werf, B. F. Van Dongen, C. A. J. Hurkens, and A. Serebrenik. 2008. Process Discovery Using Integer Linear Programming. In *Applications and Theory of Petri Nets*, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Kees M. Van Hee, and Rüdiger Valk (Eds.). Vol. 5062. Springer Berlin Heidelberg, Berlin, Heidelberg, 368–387. https://doi.org/10.1007/978-3-540-68746-7_24 Series Title: Lecture Notes in Computer Science.

[28] Michał Walicki and Diogo R. Ferreira. 2011. Sequence partitioning for process mining with unlabeled event logs. *Data & Knowledge Engineering* 70, 10 (Oct. 2011), 821–841. https://doi.org/10.1016/j.datak.2011.05.003

[29] A.J.M.M. Weijters and J.T.S. Ribeiro. 2011. Flexible Heuristics Miner (FHM). In *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. IEEE, Paris, France, 310–317. https://doi.org/10.1109/CIDM.2011.5949453