Comparing Process Mining Tools and Algorithms

Conference Paper · June 2022

DOI: 10.23919/CISTI54924.2022.9820570

Carlos J. Costa
ISEG – Lisboa School of Economics and Management. University of Lisbon

Comparing Process Mining Tools

Ana Catarina Ferreira Parente
ISEG – Universidade de Lisboa
Lisboa, Portugal
catarinaparente1@hotmail.com

Carlos J. Costa
ISEG – Universidade de Lisboa
Lisboa, Portugal
cjcosta@iseg.ulisboa.pt

Abstract — **As a result of the data explosion, most companies typically address the need to deal with massive volumes of data about their processes in their information systems. Some of this data, commonly known as event logs, contain valuable information about the activities that happen during the execution of a process in a company. Process mining attempts to extract valuable insights from these event logs. The main goal of this work is to understand the available technologies that may be used to develop process mining algorithms and to compare the results obtained. The generated process models should describe the complete workflow of decision and execution of each process so that their performance can be later evaluated according to a set of metrics.**

Keywords – Process Mining; Process discovery; Python; ProM; Power BI.

I. Introduction

As a result of the growth and development of an organization, it is expected that there will be an increase in demand from its customers and the potential problems associated with this relationship. In addition, the influence of the growth of technologies on each organization's workflow should be considered, as they can simplify these challenges. Concerning these questions, process mining is an approach that ought to be considered as it can help organizations to do the management of these issues. Through the application of these techniques, it would be possible to have a complete perception and a better understanding of each of the processes of a given organization and their interactions. Hopefully, these techniques can extract useful information from the event logs stored in the information systems [1]. In this sense, each organization can apply process mining techniques as long as they record the necessary information related to its processes in their systems, such as which activities were performed, by whom, and when. [12] Nowadays, there is, in fact, a growing tendency for organizations to prefer to store their data in a more or less structured format in their systems, either as a result of the current legislation or to improve the organization's performance [2]. Accordingly, the objective of this work involves the identification and comparison of current technologies that allow processes to emerge. The solution is centered on the integration of the current technologies to create workflows capable of representing the processes and comparing these results. To solve this problem, we propose that, first of all, the available tools aimed at the process mining area are identified. Bearing this information in mind, a set of comparison criteria must be identified so that later it is possible to compare the performance of the different process mining tools. Finally, it is expected that it will be possible to make a final recommendation of the most promising tool and algorithm.

II. Literature Review

A. *Introduction to Process Mining*

As previously introduced, one of the main challenges that companies are facing today is to figure out how to extract information from the data they are saving in their information systems to enhance the company's processes' effectiveness. These data are often records of activities that occur throughout the development of each process and are usually called event logs. Event logs contain various information, with mandatory and optional information about the events. Concerning mandatory components, the following are included: the case, that is, the instance to which the process is associated; and the activity, which corresponds to an event in the process. The extra information that event logs may contain is emphasized in [3]: an associated timestamp, that is the time the activity started and eventually the time at which the activity ended; and a resource, which corresponds to the person who did the activity. Through the analysis of this type of data, the process models are then analyzed to identify and locate bottlenecks throughout the processes, as well as to identify models that can serve as a reference for similar future processes.

B. *Process Mining Techniques*

In the literature, three different process mining techniques are distinguished: process discovery, conformance checking, and

Site:
C. Parente and C. J. Costa, "Comparing Process Mining Tools and Algorithms," *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, 2022, pp. 1-7, doi: 10.23919/CISTI54924.2022.9820570.,

model enhancement. Process discovery techniques are used to convert an event log into a process model [1]. It is usually seen as the starting point for considering deeper analyses [3]. Figure 1 shows how this technique works: a process model is generated as an output from a set of logs. The main objective of using this method is to create a model that reflects all the possible paths of a given process. The model resulting from these techniques may provide different information, according to the characteristics of the algorithm that is used.

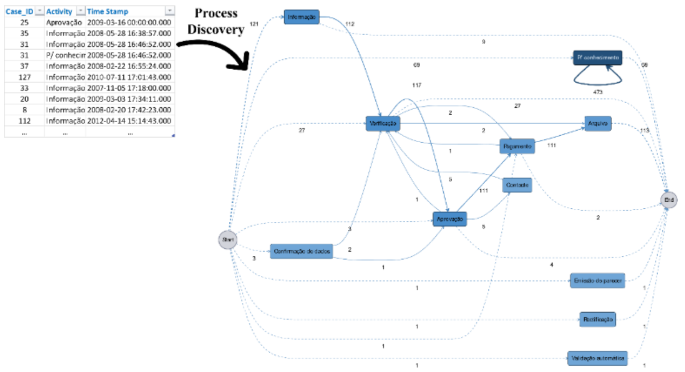

Figure 1. Process Discovery Technique.

Figure 2.

Conformance checking techniques are in charge of verifying whether the registered model corresponds to the reality by comparing both models [1]. When considering this technique, the central question is: is the process model an adequate reflection of the logs? By answering this question, it is possible to understand whether or not there is a good match between the event logs and the existing model [4]. Lastly, the purpose of applying model enhancement techniques is to try to change the original process model by applying the available information, with the ultimate goal of improving it [1].

C. Petri Nets

One of the best-known ways to model business processes is through Petri nets. These graphs include a start and an end state, and each one of these networks includes two types of nodes: places and transitions [5]. As illustrated in Figure 2, the circles represent the places, whereas the squares represent the transitions. A place can be seen as an input if there is an arc connected to the transition, or it can be considered an output otherwise [6]. A transition represents an action or an activity that is being modeled and is responsible for the movement of tokens from the input to the output place connected to [6]. Petri nets are useful to depict two circumstances that frequently occur in daily life: parallelism and choices [5]. In parallelism situations, we try to represent activities that happen in parallel. This is done by connecting a transition with several output places in the case of a parallel split. If it is a parallel join, the different places are connected to a single transition, and that transition can only fire if it receives tokens from all places. On the other hand, in the case of representing an exclusive decision situation, a place appears connected to different transitions, as is the case of the link between place p2 and transitions b and e in Figure 2.

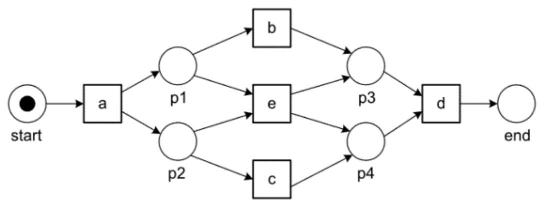

Figure 3. Example of a Petri Net. Adapted from [1].

D. Some process mining algorithms

A set of distinctive process mining algorithms is regularly described in the literature. Next, we will present some of the most common algorithms. The Alpha Miner is seen as one of the first algorithms for process discovery techniques. Implementing this algorithm makes it possible to create a Petri net that represents the underlying process. In Figure 3, it is possible to see the steps that make up the implementation of this algorithm. First (1), the existing traces are identified, i.e., the sequence of activities of a given process. Subsequently (2), the order of these activities is analyzed, and a matrix is created with the respective relationships so that, finally (3), it would be possible to create the final Petri net. Although this algorithm has some advantages, such as the ease of use, it faces particular challenges in some situations, namely: when there is too much data, and therefore there may be some noise included[1]; in the presence of complex data; and in cases of data incompleteness [1].

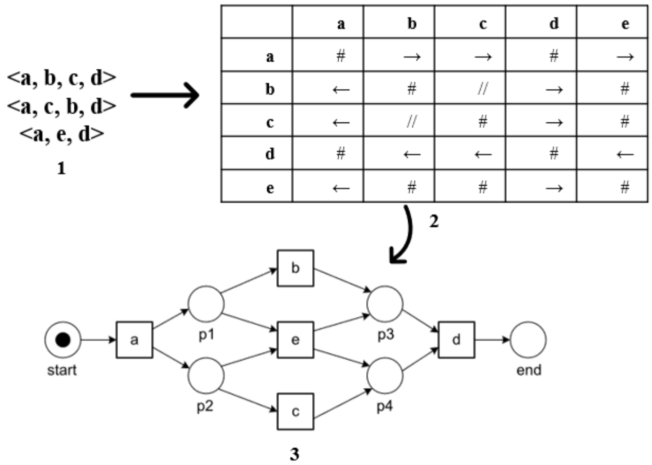

Figure 4. Alpha Miner Algorithm. Adapted from [1].

Figure 5.

The Heuristic Miner algorithm can escape from the problems of the Alpha Miner algorithm by taking into account the frequencies of each event when building the process map, and not just the order of the traces [7]. As a result, this algorithm avoids including infrequent traces in the model [1]. Consequently, this leads to more concise diagrams in the presence of outliers, since it tries to consider only the most frequent paths and not those that may appear occasionally. The last algorithm to be introduced is the Inductive Miner. In [1], this algorithm is introduced as an algorithm that shows good results when dealing with a large number of data, as well as with logs that include infrequent behavior. Typically, the output of this algorithm is a process tree, which is a modeling language characterized by preserving soundness. Even so, the results can also be converted to other representations, such as Petri nets and BPMN models [1].

E. *Tools comparison criteria*

To make a comparative analysis between the process mining tools that will be further applied in the case study, it is essential to describe the criteria that will be used.

TABLE I. COMPARISON CRITERIA

[1] In this case, it is considered that the data is noisy when it includes some outliers, i.e., when it has some records that do not represent the actual behavior.

Criteria	**Description**
License	The tool's type of license has, whether it is open-source or commercial.
Documentation	Check if there is documentation available about the tools and how they work, as it facilitates the developments of the user [8].
Integration of process mining algorithms	Check which process mining algorithms the tool can implement.
Integration of process mining techniques	Check if the tool can provide process discovery and/or conformance checking techniques.
Integration of metrics	Check whether the tool can provide metrics to evaluate the performance of the process.
Types of output models	Check with which modeling languages the tool is capable of producing models.

Several authors have already done similar works, through which they also made a comparison between different Process Mining tools. We can consider as examples the works proposed in [9] and [10], which, as expected, introduce different comparative criteria and different tools for comparison.

III. METHODOLOGY

In this study, it was perfomed an analysis of solutions, asperfomed by several researchers (e.g. [13] [14]). But it is also evaluated in a context of data science, so considering several specificities. The Cross-Industry Standard Process inspired the methodological approach used during the elaboration of this study for Data Mining (CRISP-DM) methodology[15][16], which is organized into a few stages. The first phase included, at first, the understanding of the business environment, as well as the study of the area of process mining and the main conceptual models related to this scientific domain. This study is essential to ensure that the use of data to create process models follows the needs of the business. In addition, it was also necessary to study the characteristics of the process mining tools that are available so that they can later be compared and the most appropriate ones are recommended for the development of models for the case study. Regarding the second stage, it was related to the data understanding to recognize which variables were available and with which characteristics. Subsequently, the third stage of this project consists of preparing the data to be later used to develop process mining models. In this phase, some transformations were made, such as eliminating the null values unnecessary columns, among others. Aiming to compare the tools and algorithms, it is necessary to complete the next phase, which concerns the development of process mining models. To analyze these models, it is also necessary to understand which metrics are integrated into the tools that were chosen to be studied in this project. This is one of the central steps, as these metrics are one of the central criteria for identifying the most appropriate tools.

IV. EMPIRICAL WORK

A. Technologies

Several process mining tools are available on the market, such as Disco, UiPath Studio, RapidProM, PM4PY packages applied through Python, ProM, and Power BI. However, some of these tools only allow a reduced analysis if you don't have a license, as in the first two examples. Since RapidProM is an extension of ProM within RapidMiner, it was decided to choose only ProM for comparison purposes. So, for this work, we will use three of the previous tools for comparison: PM4PY, ProM, and Power BI. It was considered interesting to produce models through these three tools since they offer different advantages that complement the analysis. Thus, with studying these tools, it may be possible to understand which are the most suitable and then provide the final recommendations.

In the first phase, Python was used for data processing and cleaning through the use of the Pandas package. Then, the PM4PY package, which is the package responsible for process mining algorithms, was installed in Jupyter. Furthermore, since the objective is to create and visualize models that represent the flow of the processes, it was also necessary to download and install the GraphViz package.

The use of the ProM tool to conduct analyses was also a must owing to the integration of multiple process mining plug-ins, which allows the creation of models from several perspectives. Besides that, providing visualizations that show the tokens running in real-time makes the analysis somewhat more interactive. It gives some insights concerning how long each event takes to be executed. Finally, this tool also includes several metrics, like the total number of cases and events, which might be helpful to analyze the performance of processes, to identify abnormal situations subsequently.

Finally, Power BI was also included since it allows the development of process mining models. It has the advantage of integrating a set of metrics, such as the average and the sum of the duration of an activity, the frequency of each activity, among others.

B. *Data Preparation*

To build process models, it is first necessary to obtain the event logs. The data extraction was done through a script on Microsoft SQL Server, the relational database management system used. The extracted data were exported to an excel file, and then it was imported into Python through the use of the Pandas library.

Concerning data treatment, the first step included removing the columns that weren't necessary, the null values, and the lines with incomplete data using the same library. After that, it was necessary to check the data types of the values. This step made it possible to perceive that two of the columns containing date values were being saved as objects, which was incorrect. We had to change the data type to date time to solve this issue. Thus, at this point, we already have the data with the desired characteristics and format.

C. *Event Log Preparation*

Depending on the tool used, the file format where the logs are recorded may differ. For example, the default accepted file format in ProM is XES, whereas Python and Power BI both accept Comma-Separated Values (CSV) and Microsoft Excel Open XML Spreadsheet (XLSX) files. In this sense, ProM offers the particularity of implementing a plug-in that enables the conversion of a CSV file to XES, allowing process mining techniques to be applied to data in a format different from that recognized by this tool. However, if this plug-in was not meant to be used, Python may also be applied to convert an event log object to an XES file. Additionally, to perform process mining algorithms, it is necessary that it is possible to distinguish an identifier of each case, the respective activities, and a timestamp associated with each of these activities in the event data. In Python, this correspondence is done through the PM4PY package. With this, it is possible to give the appropriate format to the data frame within process mining.

Regarding ProM, identifying these elements is done almost automatically by each plug-in. In this sense, it is only necessary to ensure that it is done correctly.

Finally, the identification of the case id, the activity, and the timestamp is also a must in Power BI. Nonetheless, before proceeding with this identification, the Power Query Editor carried out the data processing. In particular, the data type of columns representing dates was changed, as it was found as Text/Number and not as Date Time. Furthermore, the null values and the unnecessary columns were eliminated as well. Power BI offers process mining by importing visuals. Hence, with the creation of these visuals, it is then possible to match each of the three critical factors — case id, activity, and timestamp — with the respective columns of the query used.

V. RESULTS

The analysis of this work focuses on creating models for managing document flows. To do so, we develop process models in the three previously mentioned tools, Python, ProM, and Power BI. This analysis aims to understand which algorithms lead to the best results and which tools allow an adequate analysis according to the company's needs.

A. *Comparison between the process mining algorithms used*

We applied three different algorithms for each analysis: Alpha Miner, Heuristic Miner, and Inductive Miner. However, it should be noted that these algorithms were only directly applied in two tools, Python and ProM. In Power BI, it is impossible to employ specific algorithms, only visuals. There are certainly some of these algorithms at the base of the development of these visuals. However, this information is not made explicit.

Therefore, the first algorithm we attempted was Alpha Miner. As noted in the literature review, this algorithm does not perform well on real-life logs, and consequently, it is not recommended. Even so, we tried to create models with this procedure in Python and in ProM, to verify these conclusions. In fact, the algorithm's output is a model with separate activities, i.e., there are no links between the different activities. As such, this analysis enforces the conclusion that the algorithm has serious issues when dealing with this type of data. Then, we created process models through the Heuristic Miner algorithm, which is an improvement over the Alpha Miner. This algorithm allows the specification of some parameters, such as a threshold for the dependency factor. This parameter denotes the dependency relation between the different activities in a process and can range from -1 to 1. A value close to 1 corresponds to a positive dependency, i.e., an activity X always appears after Y; a value close to -1 corresponds to a negative dependency, that is, activity Y is frequently the source of X; and a value close to 0 means that no relationship is detected between two activities. With this, we can see that relationships with negative values for the dependency parameter are not interesting for this case since they do not correctly represent real-life events. Hence, we develop models through Heuristic Miner, taking these conclusions into account, and we verify that the algorithm works well in general, producing process models representing the event logs. However, we conclude that this algorithm reveals some issues for some instances. Namely, process models did not include all the events present in the data, and some models contained activities isolated from the workflow. This might happen because this algorithm has problems in dealing with low-frequency events. Therefore, by including more data, these issues might be addressed. The last algorithm that was applied was the Inductive Miner. The difference between the previous algorithm and this one is related to the fact that the latter does not incorporate the definition of the dependency threshold.

Furthermore, the process models resulting from this algorithm's application describe each process's behavior in a more detailed way as it does not exclude activities from the model. In addition, we can also consider that, according to the obtained process models, the way this algorithm presents the activities gives a much more concrete temporal notion when compared to previous algorithms. With this, we are somehow able to get some insights into which activity follows which.

B. *Comparison between the process mining tools used*

TABLE II. COMPARISON OF PROCESS MINING TOOLS (✓ = YES, **X** = NO)

Criteria	**PM4PY**	**ProM**	**Power BI**
License	Open-source.	Open-source.[2]	Open-source[3].
Documentation	Available.	Available.	There is information available for some visuals, but not as complete.
Integration of process mining algorithms	Alpha Miner: ✓ Heuristic Miner: ✓ Inductive Miner: ✓ Fuzzy Miner: **X**	Alpha Miner: ✓ Heuristic Miner: ✓ Inductive Miner: ✓ Fuzzy Miner: ✓	Process models are created by importing visuals, which follow algorithms but are not made explicit.
Integration of process mining techniques	Process Discovery: ✓ Conformance checking: ✓	Process Discovery: ✓ Conformance checking: ✓	Process Discovery: ✓ Conformance checking: **X**
Integration of metrics	✓	✓	✓
Types of output models	BPMN, Petri nets, heuristics nets, directly-follows graph	BPMN, Petri nets, EPCs, YAWL	-

Considering the comparison result shown in the previous table, both PM4PY and ProM verify the most features. They can therefore be considered as the most suitable tools for process mining. Nevertheless, according to the experience of developing the models for the underlying project, it was concluded that ProM is not a very accessible tool for beginners. Although a set of

[2] However, some plug-ins may be required to be released under a different license than the one currently in use — the Lesser GNU Public License (L-GPL) —, which may not be open-source.

[3] The tool is open-source, however, some visuals for developing process mining models require subscription. For example, you must have a subscription to develop conformance checking analysis.

documentation is available, some issues arose while creating process models. In particular, there is an extensive range of plug-ins and little information about their development, which makes it challenging to choose a suitable algorithm. In addition, some plug-ins do not work as they would be supposed to, either because of input errors or errors in the definition of the algorithm itself.

Concerning Power BI, it proved to be an equally interesting tool. It allows the development of process models including different metrics, such as the average duration of a specific activity and the volume. However, as shown from Table II, we conclude that there are some less positive points, namely the lack of documentation and information about it. One of the most severe issues is the lack of knowledge about the algorithms used to create the process models. As a result, we were somehow unable to conduct a complete analysis in terms of comparison due to the absence of information to perform this task. For example, we cannot determine what kind of output models each process follows.

Furthermore, during the development of the process models, we found that to create models for specific scopes through ProM and Power BI, it becomes necessary to develop a separate input for each document type. This is a very time-consuming task, which proves to be inefficient to meet the needs of any company. In this sense, and taking into account that the same developments made using Python are much more automated, we recommend that, for these types of analyses, Power BI and ProM should be reserved for exceptional processes that require more careful analysis. For example, it may be useful to use these tools to more fully evaluate events or relationships between events identified as being too slow and, therefore, possible reasons that are degrading performance.

VI. Discussion

The previously described results focus on the analysis and the comparison of process mining tools and algorithms. Compared with other related works, such as the ones mentioned above in Section II, this paper provides a differentiating and potentially more complete analysis, as it encompasses a comparison for both factors, tools, and algorithms, rather than focusing on just one.

Furthermore, we did not find related works in this field that included the PM4PY or the Power BI tools in their comparison. Specifically, considering the works already mentioned, we can conclude that there is a practice standard tool to all, namely ProM, and that the findings reached are pretty similar to those reached in this work. Explicitly, both in the conclusions drawn from this work and [9], we see that ProM is considered a significant tool but not recommended for beginners due to its not very easy interface. On the other hand, the conclusions obtained, regarding the algorithms, can be compared with those obtained from the research stated in [11]. However, this study sought to analyze different algorithms from those evaluated in this work. In this sense, the conclusions to be drawn were different. We can find some similarities in the performance of the standard algorithms, namely regarding the conclusions drawn for the Inductive Miner algorithm. As such, in both studies, this algorithm is seen as a model that preserves the soundness property, i.e., that truly reflects all possible paths and thus does not exclude infrequent behavior.

VII. Conclusions And Future Work

This paper aimed to apply process mining techniques to discover a process model from the logs in different tools, namely Python, ProM, and Power BI. Accordingly, the main goal was to compare and analyze the characteristics of each of these tools' results and the algorithms used. Taking into account what was exposed in Section V, the algorithm that showed the best results was the Inductive Miner. Unlike the remaining algorithms, this one did not exclude low-frequency activities. This is especially crucial when studying documents with few activities since little information is available.

Furthermore, this algorithm offers a more realistic temporal perception as it allows us to understand the order of events. On the other hand, in some process models from other algorithms, such as the Heuristic Miner, we see a set of events linked to the initial node and dispersed over several areas of the model, which does not allow such a complete perception of the moment when the event happened. Regarding the tools, to evaluate the results, we defined a set of variables used as criteria for comparison. Nonetheless, the considerations are not so clear and depend on the needs of each user. Thus, given the scope of this work and the company's needs, the use of the PM4PY package through Python proved to be the most appropriate. As the remaining Python libraries can be integrated into the process mining analysis, the models can be created quickly and automatically for various scopes. Hence, not only were we able to make a complete analysis, but we were also able to optimize the time spent.

Additionally, we can provide almost the same metrics and even complement the analysis by including several statistics, such as the event distribution over time and the rework activities, among others. Therefore, we recommend the other tools complement rather than the primary resource. The resulting process models proved to be a beneficial means of understanding the flow of events and which activities and relationships were most critical and resource-consuming. These process models were obtained in different algorithms based on process discovery techniques. Thus, for future work, it could be interesting also to extend these analyzes to conformance checking techniques to check whether the resulting process models conform with the event logs. Additionally, in this work, a more descriptive and qualitative analysis was carried out. For future work, it could also be interesting to move to a more quantitative multi-criteria analysis, according to which each of the dimensional criteria studied would be given weight.

Acknowledgment

We gratefully acknowledge financial support from FCT -Fundação para a Ciência e a Tecnologia (Portugal), national funding through research grant UIDB/04521/2020.

References

[1] W. van der Aalst, *Process Mining*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. doi: 10.1007/978-3-662-49851-4.

[2] W. M. P. van der Aalst *et al.*, "Business process mining: An industrial application," *Information Systems*, vol. 32, no. 5, pp. 713–732, Jul. 2007, doi: 10.1016/j.is.2006.05.003.

[3] W. van der Aalst *et al.*, "Process Mining Manifesto," in *Business Process Management Workshops*, Berlin, Heidelberg, 2012, pp. 169–194. doi: 10.1007/978-3-642-28108-2_19.

[4] A. Rozinat and W. M. P. van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Information Systems*, vol. 33, no. 1, pp. 64–95, Mar. 2008, doi: 10.1016/j.is.2007.07.001.

[5] T. Murata, "Petri Nets: Properties, Analysis and Applications," *PROCEEDINGS OF THE IEEE*, vol. 77, no. 4, p. 40, 1989.

[6] W. M. P. Van Der Aalst, "The application of Petri Nets to workflow management," *J CIRCUIT SYST COMP*, vol. 08, no. 01, pp. 21–66, Feb. 1998, doi: 10.1142/S0218126698000043.

[7] A. Weijters, V. Aalst, and A. K. a. D. Medeiros, "Process mining with the HeuristicsMiner algorithm," 2006.

[8] A. Berti and S. J. van Zelst, "Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science," p. 4.

[9] U. ÇeliK and E. AkçetiN, "Process Mining Tools Comparison," *AJIT-e*, vol. 9, no. 34, pp. 97–104, Nov. 2018, doi: 10.5824/1309-1581.2018.4.007.x.

[10] P. Drakoulogkonas and D. Apostolou, "On the Selection of Process Mining Tools," *Electronics*, vol. 10, no. 4, p. 451, Feb. 2021, doi: 10.3390/electronics10040451.

[11] J. Saint, Y. Fan, S. Singh, D. Gasevic, and A. Pardo, "Using process mining to analyse self-regulated learning: a systematic analysis of four algorithms," in *LAK21: 11th International Learning Analytics and Knowledge Conference*, Irvine CA USA, Apr. 2021, pp. 333–343. doi: 10.1145/3448139.3448171.

[12] C J. Costa and M. Aparicio. "Analysis of e-learning processes." In Proceedings of the 2011 Workshop on Open Source and Design of Communication,. 2011 pp. 37-40. doi:10.1145/2016716.2016726

[13] M. Reixa, C. Carlos and M. Aparicio. "Cloud services evaluation framework." In *Proceedings of the Workshop on Open Source and Design of Communication*, 2012 pp. 61-69. doi: 10.1145/2016716.2016726

[14] P.Antunes, and C.. Costa. "Perceived value: A low-cost approach to evaluate meetingware." In International Conference on Collaboration and Technology, Springer, Berlin, Heidelberg, 2003. pp. 109-125. DOI: 10.1007/978-3-540-39850-9_10

[15] C. J. Costa, J., Aparício. "POST-DS: A Methodology to Boost Data Science" in 2020 15th Iberian Conference on Information Systems and Technologies (CISTI), 2020. pp. 1-6, . IEEE. https://doi.org/10.23919/CISTI49556.2020.9140932

[16] C. Costa, J. Aparicio A Methodology to Boost Data Science in the Context of COVID-19. In Advances in Parallel & Distributed Processing, and Applications, Springer, Cham, 2021, pp. 65-75, https://doi.org/10.1007/978-3-030-69984-0_7