Measuring the Complexity of Business Process Data for Predictive Process Monitoring*

Yeonsu Kim and Marco Comuzzi

Ulsan National Institute of Science and Technology, Ulsan, Republic of Korea
{yeon17,mcomuzzi}@unist.ac.kr

Abstract. Predictive Process Monitoring (PPM) aims at creating predictive models of aspects of interest of business process execution using historical data logged in event logs. Anticipating the expected performance of a PPM model can be crucial for model owners, for instance, to decide whether or not to embark on the model development in the first place. However, this is hard to do in practice. Each log is generated by a different business process and may or not contain attributes that effectively discriminate across the labels to be predicted. In this work, we investigate the extent to which the complexity measures developed in the machine learning (ML) literature for traditional classification problems can be used to anticipate the model performance in the specific PPM use case of outcome prediction. We found, as we could expect, a negative correlation between the encoded event log complexity and the model performance. This correlation is significant for a specific set of complexity measures when considering the model accuracy as a performance measure.

Keywords: Predictive Process Monitoring · Meta Feature · Complexity Measure.

1 Introduction

Predictive process monitoring (PPM) aims at creating predictive models of aspects of interests of business process executions using historic data contained in so-called event logs [19], mostly using machine learning (ML) techniques. Given an event log, several aspects can be predicted, such as the remaining duration of running cases, the next activities to be executed in a case, or the case outcome, as captured by a categorical label.

PPM has suffered from a general lack of standardization. While standard pipelines and AutoML tools have proliferated during the last few years for traditional ML tasks [9], there is no agreed-upon pipeline for PPM tasks, and related tools are lacking. Few research works have tried to create benchmarks for different PPM tasks [19, 18], but not in a systematic way. The standardization issue is exacerbated by the nature of event logs, each of which originates from a business

* Sponsored by the NRF Korea, Grant Number 2022R1F1A1072843.

process with possibly very diverse characteristics. Such a lack of standardization hinders, on the one hand, the reproducibility of PPM results. On the other hand, it prevents the development of systematic benchmarks through which we could estimate the performance that could be achieved in a given PPM task.

In this paper, we focus on the latter aspect. Having the means to estimate the performance that can be achieved by a model in PPM can be crucial for several reasons, such as saving time and resources during the model development phase, e.g., while tuning the model hyperparameters, or deciding whether it would make sense to train a model in the first place. No process owner would in fact embark on a PPM task knowing beforehand that the maximum accuracy that they are likely to obtain from a model would be below a certain level that they deem necessary to make informed decisions using that model.

There appears to be a consensus that the model performance in a PPM task is related to the complexity of an event log. The higher the complexity of a log, the more complex the *hidden patterns* in the data to be learned by a PPM model, and therefore the lower the performance of a PPM model trained using that log. Log complexity can be measured simply using the number and frequency of trace variants [18], or using more complex measures accounting for the *entropy* of an event log [2, 3]. These measures, however, only account for the complexity of the *control flow* of the process that has generated an event log. They fail to capture the complexity associated with other event log attributes, from which features of a PPM model can be derived.

ML research has investigated the problem of estimating the performance of a trained model based on the characteristics of the training dataset. Lorena et al. [11], in particular, have devised a set of measures of the complexity of a training dataset for classification problems. These measures have been used in different domains, e.g. [5], to show that there is a significant and negative correlation between the complexity of the training dataset and the model performance, i.e., the more complex the data, the lower the performance that can be obtained from a model.

In this work, we propose a general framework for investigating the effect of the complexity of an event log, as captured by the complexity measures of [11], and the performance of a PPM model trained using that log. We also present the application of this framework in the specific case of the outcome prediction PPM task. We consider several public event log datasets and different ways of encoding them to show that, even in the case of event logs, there is a strong and negative correlation between the complexity of a log and the model performance. Specifically, we show that the correlation is stronger when the model accuracy is considered a performance measure and that it is significant only for specific types of complexity measures.

This paper is organized as follows: Section 2 discusses related works. Section 3 introduces the general framework, whereas Section 4 shows its application in the case of outcome-oriented PPM. Section 5 concludes the paper.

2 Related Work

Because event logs originate from different types of processes executed in various domains and contexts, a domain-independent measure of event log complexity has been sought by researchers in recent years. Back et al. [3] proposed five estimators to compute log entropy, while Augusto et al. [2] proposed measures to estimate event log complexity starting from the complexity of the underlying business process. In both approaches, event log complexity is estimated considering mainly the control flow relations among process activities captured in an event log.

In ML, measures of the complexity of a dataset can be used to create features for meta-learning. In a nutshell, meta-learning involves the study of ways to design meta-features that are independent of the particular dataset used and that can be used to train models that can learn efficiently from diverse input data (which share similar values of the meta-features). Rivolli et al. [16] have discussed extensively the concept of meta-features, expanding on the information-theoretic and statistical description of [6].

In the ML literature, Ho and Basu [8] first proposed a set of classification dataset complexity measures, which can be used to create meta-features. These measures capture how complex a dataset for a classification task is from several perspectives, such as the linear separability of the features or the class imbalance. Lorena et al. [11] have systematically reviewed and extended Ho and Basu's work, creating a set of normalized complexity measures, which we use in this paper.

There are many empirical studies that demonstrate the negative correlation between the complexity of a dataset and the performance of a classification model learned from such data, i.e., the more complex the data, the lower the performance attainable by a model. Moran et al. [14] consider microarray datasets capturing gene expressions and use the complexity measures for feature selection. Furthermore, they performed a paired t-test to compare classification performances. Barrella et al. [4] have demonstrated the correlation in the case of imbalanced classification tasks on widely used publicly available datasets.

Lorena et al.'s [11] complexity measures have never been used in PPM. PPM represents an interesting application context for the complexity measures because, as highlighted by the research on event log complexity, each event log originates from a different domain and context. Therefore, it can be crucial to have domain-independent meta-features that could be used to anticipate the level of performance attainable by a classification model in PPM.

3 Method

Fig. 1 shows a general framework for evaluating the correlation between the values of the complexity measures of an event log dataset and the performance of a PPM model in a given PPM task. The input of the framework is a set of event logs. The first step concerns the encoding of the event logs, which generates a set of encoded logs. Then, two independent tasks are performed:

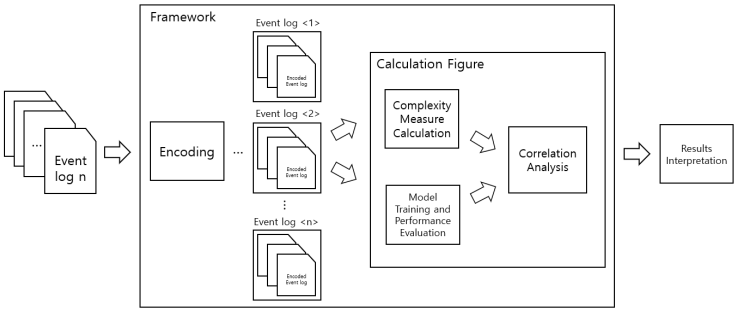

Fig. 1. A framework for assessing the correlation between event log complexity and PPM model performance.

calculating the values of the complexity measures of the encoded event logs and training and testing the classification models for a PPM task. In this paper, we consider outcome prediction as a PPM task. Finally, we analyze the correlation between the values of the complexity measures and the performance obtained by the model. Note that the higher the number of event logs (and, therefore, encoded datasets) considered, the more data will be available for assessing the correlation and, therefore, the more likely to obtain significant results if such a correlation exists. In the last step, the correlation results obtained are interpreted and discussed.

Regarding the assessment of the complexity of a classification problem, we consider the classification complexity measures defined by Lorena et al. [11]. Let us consider a classification problem in which a dataset D contains n pairs of examples $(\mathbf{x}_i, y_i)$, with $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,m})$ the vector of m features and $y_i \in \{0, 1\}$ the value of the binary[1] label.

There are six types of classification measures, which are discussed briefly next. As an illustration, Tab. 1 provides the definition of one specific measure defined in [11] for each type:

Feature-based measures: These measure the discriminative power of the features, that is, how well they separate the two classes. If at least one feature x_i is highly discriminative, then the classification problem is simple. The maximum Fischer's discriminant ratio F1 (see Tab. 1) is an example of this type of measure. Note that r_{f_i} is a discriminant ratio of each feature f_i, for which several definitions have been proposed.

Linearity measures: These measures quantify the extent to which the two classes can be linearly separated. In [11], the problem of linearly separating the classes is formulated as one of finding the hyperplane that separates the two classes with maximum margin while minimizing the training error. This hyperplane is

[1] For simplicity, we assume that the outcome label is binary, which is normally the case in outcome-oriented PPM.

Table 1. Example of classification complexity measures for each of the six types.

Name	**Symbol in [11]**	**Type**	**Definition**
Max. Fischer discriminant ratio	$F1$	Feature-based	$\frac{1}{1+\max_{i=1}^{m} r_{f_i}}$ with $r_{f_i} = \frac{\sum_{j=1}^{n_c} n_{c_j}(\mu_{c_j}^{f_i} - \mu^{f_i})^2}{\sum_{j=1}^{n_c}\sum_{l=1}^{n_{c_j}}(x_{li}^{j} - \mu_{c_j}^{f_i})^2}$
Error Rate of Linear Classifier	$L2$	Linearity	$\frac{\sum_{i=1}^{n} I(h(\mathbf{x}_i) \neq y_i}{n}$
Fraction of Borderline Points	$N1$	Neighborhood	$\frac{1}{n}\sum_{i=1}^{n} I((x_i, x_j) \in MST \wedge y_i \neq y_j)$
Graph Density	*Density*	Network	$1 - \frac{2\|E\|}{n(n-1)}$
Avg. number of features per dimension	$T2$	Dimensionality	$\frac{m}{n}$
Entropy of classes proportions	$C1$	Class imbalance	$-\frac{1}{log(n_{c_i})}\sum_{i=1}^{n_c} p_{c_i} log(p_{c_i})$

found to fit an SVM model on the dataset D. As an example of these measures, the error rate of the linear classifier $L2$ in Tab. 1 computes, given the trained SVM model $h(\mathbf{x})$, the normalized number of times that it predicts a wrong label. Higher values of $L2$ denote more errors and, therefore, a more complex classification problem.

Neighborhood measures: These measures are based on the Minimum Spanning Tree (MST) algorithm to quantify how close observations in a dataset are from their neighbors. The MST algorithm considers each data point to be a vertex and connects the minimum distance to edges. As an example measure of this type, $N1$ simply counts the average number of connections in the MST.

Network measures: These measures assess datasets using a graph, where two observations (nodes in the graph) are linked by an edge if the distance between them is smaller than ϵ. This results in a total of $n(n-1)/2$ interconnected nodes in the dataset. Density is then normalized to $1 - 2|E|/n(n-1)$, where $|E|$ is the number of edges in the graph. When observations are more distant from each other (exceeding the ϵ distance), fewer edges are formed. A decrease in edges implies sparser connections between data points, indicating higher complexity.

Dimensionality measures: These measures address the dimensionality of datasets as the classifier's performance hinges on the dataset's sparsity. The dimensionality measure illustrates the data's sparsity, with the $T2$ measure serving as the foundational metric. It quantifies the ratio between the event count and the number of feature columns. This approach also extends to examining the dataset's characteristics through techniques like PCA.

Class imbalance measures: These measures are related to the entropy of the classes. When the dataset is biased on one side, i.e., one output class is much more frequent than the other one, a model will always tend to select the majority class. Class imbalance measures the extent of such imbalance between classes.

Regarding the correlation analysis, we consider Pearson's product-moment correlation coefficient (PPMCC), which is captured by the correlation coefficient r_{ppmcc}. According to [15] and [17], the correlation between two vectors x and y of size k is calculated as:

$$r_{ppmcc} = \frac{\sum_{j=1}^{k}(x_j - \overline{x})(y_j - \overline{y})}{\sqrt{\sum_{j=1}^{k}(x_j - \overline{x})^2 \sum_{j=1}^{k}(y_j - \overline{y})^2}} \tag{1}$$

where

$$\overline{x} = \frac{\sum_{j=1}^{k} x_j}{k}, \overline{y} = \frac{\sum_{j=1}^{k} y_j}{k}. \tag{2}$$

4 Applying the framework to outcome-oriented PPM

We have applied the framework described in the previous section to the PPM use case of outcome prediction [2]. Briefly, given an event log E, this use case aims at learning the function $\hat{Y} : X \rightarrow \{0, 1\}$, which relates feature vectors encoded from process traces in E to their binary outcome label. Specifically, $\mathbf{x}_i \in X$ is a feature vector encoded from the events of the i-th case in E, whereas $y_i \in \{0, 1\}$ is the value of a categorical label capturing the outcome of the execution of that case (e.g., positive v. negative).

We consider three of the publicly available logs published by the Business Process Intelligence Challenge (BPIC 2012, 2015 and 2017). For the BPIC 2015 log, we consider municipality number 1. For the BPIC 2017 log, we consider the `accepted` outcome label. We use `pymfe` [1][3] to calculate the complexity measures. For BPIC 2017, we consider 100 mini-batches, showing the mean values obtained, i.e., for the log complexity measure. This is because the BPIC 2017 is one order of magnitude larger than the other logs, posing significant computational challenges when calculating the complexity measures when used in its entirety.

For the encoding methods, we consider the following three methods, which can be applied to any event log and have been used in the past in the literature [19]:

– Aggregated encoding (`agg`): In this case, every categorical attribute is encoded using its frequency in the events of a case. The timestamps are aggregated using the average of `datetimes`. We then applied one-hot encoding to the aggregated categorical variables.

[2] The code and datasets to reprodce the experiments are available at `github.com/Yeoonsu/Complexity_measures_for_PPM`

[3] `https://github.com/ealcobaca/pymfe`

- Index-based encoding (`index`): Index-based encoding uses all information available in an event log. Each attribute is encoded as-is. Categorical attributes are encoded using one-hot encoding.
- Index-based encoding excluding categorical variables (`excat`): It is based on index-based encoding, however the categorical data other than activity labels are not encoded. When conducting an initial test, we observed improved accuracy of the model using this encoding and, therefore, we decided to keep this in the experiments.

Combining three event logs and three encoding methods yields nine different datasets. These are labeled `<encoding><YY>` when presenting the results, e.g., `agg12` for the dataset obtained by applying the aggregation encoding method to the BPIC 2012 dataset. As far as classification techniques are concerned, we consider four tree-based classifiers: Decision Tree (DT), Random Forest (RF), LightGBM (LGBM) and XGboost (XGB). Tree-based models are generally well-performing in outcome-oriented PPM, even outscoring standard deep learning-based architectures [19]. We split the input dataset into 70% training and 30% testing.

For the PPM model performance, we consider the accuracy and AUC (Area Under the receiving operator Curve) measures. Accuracy is more intuitive, but it is deemed not appropriate for highly imbalanced classification problems. Finally, a detailed interpretation of the results is provided to gain deeper insights into the implications and underlying patterns inherent in the analyzed data. We anticipate a negative correlation between the model performance (accuracy/AUC score) and the input dataset complexity.

Next, we present the experimental results obtained in the following order:

1. Values of the complexity measures for the considered datasets. These results are used to select a meaningful subset of complexity measures to consider for the correlation analysis.
2. Performance of the PPM models.
3. Analysis of the correlation between the PPM model performance and the input dataset complexity.

Fig. 2 shows the values of the complexity measures of the encoded event logs considered in our experiment. Note that the table already excludes the complexity measures $F1$, $F2$, $N3$, $N4$, $T3$, and $T4$, for which, at least for some event logs, the computation of the complexity measures returned a NaN value. In these cases, it appears that the encoding methods adopted tend to produce, during the calculation of these measures, a sparse matrix, which makes calculations expensive and practically impossible, leading to a NaN result.

Interestingly, the measures $F4$, $L1$, and $L2$ have zero values for all datasets. An $F4 = 0$ indicates that there are no overlapping regions in the data. As illustrated in Figure 2, F4 is calculated by applying the measure $F3$ in T rounds. $F3$ calculates the individual efficiency of each feature in dividing the classes and considers the maximum value found among the m features in a dataset. $L1 = 0$

	agg12	excat12	index12	agg15	excat15	index15	agg17	excat17	index17
c1	0.006585	0.006585	0.006585	0.76761	0.76761	0.76761	0.972072	0.792439	0.793482
c2	0.99893	0.99893	0.99893	0.466788	0.466788	0.466788	0.072907	0.052349	0.051985
cls_coef	0.239084	0.211612	0.334102	0.32434	0.537851	0.573136	0.319229	0.369856	0.29635
density	0.811926	0.817829	0.808387	0.849609	0.864444	0.867406	0.832388	0.980124	0.981351
f3.mean	0.186979	0.39459	0.39459	0.779064	0.989101	0.985468	0.547166	0	0
f4.mean	0	0	0	0	0	0	0	0	0
hubs.mean	0.703682	0.77659	0.75629	0.726348	0.818902	0.798346	0.760223	0.92758	0.925666
hubs.sd	0.313491	0.23859	0.208014	0.254665	0.117455	0.150237	0.327931	0.223925	0.240419
l1.mean	0	0	0	0	0	0	0	0	0
l2.mean	0	0	0	0	0	0	0	0	0
l3.mean	0	0	0	0	0.008401	0.000681	0.000064	0	0
lsc	0.547457	0.730451	0.722171	0.974656	0.999216	0.998495	0.857399	0.996086	0.995388
n1	0.001223	0.001146	0.001605	0.06653	0.419391	0.397593	0.098248	0.923631	0.913121
n2.mean	0.202883	0.167281	0.182587	0.288829	0.456486	0.414374	0.349683	0.637155	0.641872
n2.sd	0.117423	0.180582	0.155839	0.095975	0.115029	0.156372	0.087581	0.158183	0.160594
t1.mean	7.64E-05	7.64E-05	7.64E-05	0.000227	0.000227	0.000227	0.003185	0.003185	0.003185
t1.sd	1.36E-20	1.36E-20	1.36E-20	5.42E-20	3.42E-06	3.42E-06	0	0.00E+00	0.00E+00
t2	1.3437	4.345534	4.37648	1.583333	1.178928	5.829473	1.175159	1.665605	6.146497

Table 2. Complexity values of considered datasets (excluding measures that yield NaN results).

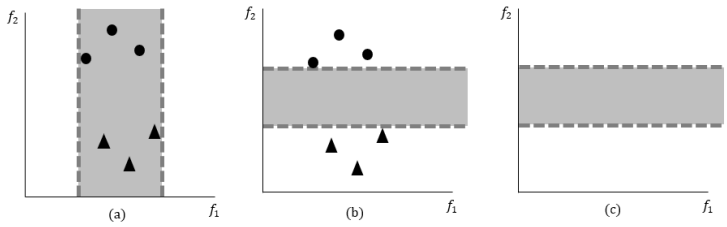

Fig. 2. The case for $F4 = 0$

and $L2 = 0$ indicate that the features in the input dataset can be linearly separated, which is relevant when using Support Vector Machine (SVM). Based on the complexity measure calculation, for the correlation analysis, we utilized the complexity measures $C1$, $C2$, cls_coef, $density$, $F3$, $Hubs$, $L3$, lsc, $N1$, $N2$, $T1$, and $T2$.

Note that other domain applications of the complexity measures have made a similar choice to exclude some of them from the analysis, either because their calculation was infeasible or the values obtained were not helpful in discriminating among the datasets. Table 3 provides an overview of the complexity measures considered by other research works in the literature.

	C1	C2	cls_coef	density	F1	F1v	F2	F3	F4	Hubs	L1	L2	L3	lsc	N1	N2	N3	N4	T1	T2	T3	T4
(Luengo, 2010)[12]							O	O			O	O				O	O	O		O		
(Luengo, 2013)[13]					O		O	O			O	O	O		O	O	O	O	O	O		
(Moran, 2016)[14]					O		O	O			O	O	O		O	O	O	O	O	O		
(Francisco, 2022)[5]					O		O	O			O	O	O		O	O	O	O	O	O		
(Komornicazk, 2022)[10]	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O

Table 3. Complexity measures used by other papers.

		agg12	excat12	index12	agg15	excat15	index15	agg17	excat17	index17
DT	Accuracy	0.8838808	0.809778	0.809778	0.783661	0.790469	0.789713	0.745225	0.631473	0.599958
	AUC	0.442053	0.404992	0.404992	0.524511	0.501799	0.5	0.781736	0.553885	0.5
RF	Accuracy	0.8186911	0.832951	0.777184	0.871407	0.789713	0.789713	0.973154	0.74618	0.758383
	AUC	0.9093225	0.916454	0.888563	0.717844	0.5	0.5	0.970093	0.687487	0.708484
XGB	Accuracy	**0.9997454**	0.995162	0.99287	**0.999244**	**0.837368**	**0.832829**	0.985144	**0.844228**	**0.864283**
	AUC	0.5	0.497708	0.496561	0.998333	0.613309	0.614395	0.982675	0.824617	0.843408
LGBM	Accruacy	**0.9997454**	**0.998727**	**0.998472**	0.998487	0.805598	0.801815	**0.98525**	0.840832	0.861205
	AUC	0.5	0.499491	0.499363	0.997844	0.565482	0.615873	0.982681	0.820727	0.841595

Table 4. Performance of PPM models (best model for each encoded log highlighted in bold).

	C1	C2	cls_coef	density	F3	Hubs	L3	LSC	N1	N2	T1	T2
Pearson's r	-0.581	0.577	-0.744	-0.724	-0.127	-0.766	-0.464	-0.739	-0.833	-0.837	-0.314	-0.243
p-value	0.101	0.104	**0.021**	**0.028**	0.744	**0.016**	0.208	**0.023**	**0.005**	**0.005**	0.411	0.529

Table 5. Pearson's r and p-value of the correlation between complexity measures and model accuracy (significant values at $p<0.05$ highlighted in bold).

Table 4 shows the PPM model performance obtained for the different encoded event logs. It can be noticed that XGB and LGBM exhibit identical values for the dataset `agg12`. This could be attributed to their shared foundation in tree-based models. In the majority of cases, the XGB models show the best performance, which is consistent with what is found in the literature [19].

	C1	C2	cls_coef	density	F3	Hubs	L3	LSC	N1	N2	T1	T2
Pearson's r	-0.243	0.11	**-0.866**	-0.25	-0.455	-0.368	-0.662	-0.451	-0.421	-0.41	0.19	-0.264
p-value	0.529	0.778	**0.003**	0.517	0.219	0.329	0.052	0.222	0.259	0.272	0.624	0.493

Table 6. Pearson's r and p-value of the correlation between complexity measures and model AUC (significant values at $p<0.05$ highlighted in bold).

We have now complexity measures (see Table 1) and model performance (see Table 4) obtained for a total of 9 different datasets. Table 5 and Table 6 show the results of the correlation analysis between the complexity measures and the PPM model performance (accuracy and AUC, respectively), which is the last step of the proposed framework.

When accuracy is considered as a performance measure (see Table 5), the complexity measures cls_coef, $density$, $Hubs$, lsc, $N1$, and $N2$ are strongly, negatively and significantly correlated with the PPM model performance. This result is aligned with our expectations and confirms that a negative correlation between the input dataset complexity and the model performance can be claimed also in the case of outcome-oriented PPM. More specifically, the negative correlation is more evident when using complexity measures that capture the network structure of the dataset (e.g., cls_coef, $Hubs$) and the exclusivity of neighbouring data (e.g., LSC, $N1$, $N2$). Event logs have indeed a network structure, as acknowledged for instance by the recent interest in applying graph neural networks in PPM [7], and the traces in an event log can normally be clustered efficiently based on the attribute data.

When the accuracy is considered as a performance measure (see Table 6) only the measure cls_coef remains significant to characterise the negative correlation. This lack of statistically significant results calls for further analysis. It may be due to the lack of observation points in the correlation analysis (only 9 datasets are considered) and the correlation may become significant if more observation points are generated. Yet, all the correlation coefficients obtained (except $C2$ and $T1$) are strongly negative.

5 Conclusions

We have proposed a framework to investigate the correlation between the complexity of an event log, as captured by classification complexity measures in ML, and the performance of an outcome-oriented PPM model. As anticipated, we observed a negative relationship between the event log complexity and the model accuracy. This aligns with our expectations since the same correlation has been found also in other domains.

Specifically, we found that the network structure complexity measures cls_coef, $density$, $Hubs$, LSC, $N1$, and $N2$ exhibited a significant negative relationship with model performance when accuracy is considered. In contrast, in the case of AUC, only one measure (cls_coef) displayed a significant negative relationship.

The work presented in the paper has several limitations. We considered a limited set of event logs, varying only the way in which they are encoded. More observations for the correlation analysis could be generated by considering more input datasets. The correlation analysis is based on standard correlation metrics, which are not suitable for investigating whether a causal relation between the variables involved exists. Besides addressing these limitations, this work can also be extended by comparing the results obtained with event log complexity measures specifically proposed in process mining research.

References

1. E. Alcobaça, F. Siqueira, A. Rivolli, L. P. Garcia, J. T. Oliva, and A. C. De Carvalho. Mfe: Towards reproducible meta-feature extraction. *The Journal of Machine Learning Research*, 21(1):4503–4507, 2020.
2. A. Augusto, J. Mendling, M. Vidgof, and B. Wurm. The connection between process complexity of event sequences and models discovered by process mining. *Information Sciences*, 598:196–215, 2022.
3. C. O. Back, S. Debois, and T. Slaats. Entropy as a measure of log variability. *Journal on Data Semantics*, 8:129–156, 2019.
4. V. H. Barella, L. P. Garcia, M. P. de Souto, A. C. Lorena, and A. de Carvalho. Data complexity measures for imbalanced classification tasks. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.
5. F. J. Camacho-Urriolagoitia, Y. Villuendas-Rey, I. López-Yáñez, O. Camacho-Nieto, and C. Yáñez-Márquez. Correlation assessment of the performance of associative classifiers on credit datasets based on data complexity measures. *Mathematics*, 10(9):1460, 2022.
6. C. Castiello, G. Castellano, and A. M. Fanelli. Meta-data: Characterization of input features for meta-learning. In *Modeling Decisions for Artificial Intelligence: Second International Conference, MDAI 2005, Tsukuba, Japan, July 25-27, 2005. Proceedings 2*, pages 457–468. Springer, 2005.
7. M. Harl, S. Weinzierl, M. Stierle, and M. Matzner. Explainable predictive business process monitoring using gated graph neural networks. *Journal of Decision Systems*, 29(sup1):312–327, 2020.
8. T. K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE transactions on pattern analysis and machine intelligence*, 24(3):289–300, 2002.
9. S. K. Karmaker, M. M. Hassan, M. J. Smith, L. Xu, C. Zhai, and K. Veeramachaneni. Automl to date and beyond: Challenges and opportunities. *ACM Computing Surveys (CSUR)*, 54(8):1–36, 2021.
10. J. Komorniczak, P. Ksieniewicz, and M. Woźniak. Data complexity and classification accuracy correlation in oversampling algorithms. In *Fourth International Workshop on Learning with Imbalanced Domains: Theory and Applications*, pages 175–186. PMLR, 2022.
11. A. C. Lorena, L. P. Garcia, J. Lehmann, M. C. Souto, and T. K. Ho. How complex is your classification problem? a survey on measuring classification complexity. *ACM Computing Surveys (CSUR)*, 52(5):1–34, 2019.
12. J. Luengo and F. Herrera. Domains of competence of fuzzy rule based classification systems with data complexity measures: A case of study using a fuzzy hybrid

genetic based machine learning method. *Fuzzy Sets and Systems*, 161(1):3–19, 2010.
13. J. Luengo and F. Herrera. An automatic extraction method of the domains of competence for learning classifiers using data complexity measures. *Knowledge and Information Systems*, 42:147–180, 2015.
14. L. Morán-Fernández, V. Bolón-Canedo, and A. Alonso-Betanzos. Can classification performance be predicted by complexity measures? a study using microarray data. *Knowledge and Information Systems*, 51:1067–1090, 2017.
15. F. Z. Okwonu, B. L. Asaju, and F. I. Arunaye. Breakdown analysis of pearson correlation coefficient and robust correlation methods. In *IOP Conference Series: Materials Science and Engineering*, volume 917, page 012065. IOP Publishing, 2020.
16. A. Rivolli, L. P. Garcia, C. Soares, J. Vanschoren, and A. C. de Carvalho. Meta-features for meta-learning. *Knowledge-Based Systems*, 240:108101, 2022.
17. C. Spearman. The proof and measurement of association between two things. *International journal of epidemiology*, 39(5):1137–1150, 2010.
18. B. A. Tama, M. Comuzzi, and J. Ko. An empirical investigation of different classifiers, encoding, and ensemble schemes for next event prediction using business process event logs. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(6):1–34, 2020.
19. I. Teinemaa, M. Dumas, M. L. Rosa, and F. M. Maggi. Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(2):1–57, 2019.