Evaluating Stability of Post-hoc Explanations for Business Process Predictions

Mythreyi Velmurugan[1(✉)], Chun Ouyang[1], Catarina Moreira[1], and Renuka Sindhgatta[2]

[1] Queensland University of Technology, Brisbane, Australia
{m.velmurugan,c.ouyang,catarina.pintomoreira}@qut.edu.au
[2] IBM Research, Bangalore, India
renuka.sr@ibm.com

Abstract. Predictive process analytics uses advanced machine learning techniques to accurately predict future states of running business processes. Given the complexity of these predictive models, explainable AI techniques are also required to enable informed decision-making. However, few studies evaluate the quality of explanations provided by existing methods to explain business process predictions. In this paper, we attempt to evaluate the consistency of explanations produced for process predictions by two popular explainable methods. We propose that methods and metrics to assess feature selection algorithms can be used to evaluate explanation stability. We use these metrics to assess explanations produced by LIME and SHAP. Our findings indicate that explanation stability may depend on dataset characteristics, feature construction methods and predictive model characteristics. In addition, we also find that, though stable explanations are needed for informed decision-making, unexpected behaviour in explanation stability can act as a diagnostic tool to determine model quality.

Keywords: Predictive process analytics · Explainable AI · Evaluation metrics · Explanation stability

1 Introduction

Predictive process analytics (PPA) attempts to predict some future state of a business process [6]. It uses event logs, which capture process execution data, to train predictive models. As these models require advanced machine learning algorithms to create accurate predictions, their internal workings are complex, and thus opaque to a human audience. The research field of explainable AI (XAI) provides methods to interpret these opaque, *"black box"* predictive models [3]. Recent studies in PPA have applied existing XAI methods to explain process predictions [2,13] or evaluate process predictive models [11]. However, few works have attempted to evaluate the quality of explanations generated by these methods for process predictions.

H. Hacid et al. (Eds.): ICSOC 2021, LNCS 13121, pp. 49–64, 2021.
https://doi.org/10.1007/978-3-030-91431-8_4

This has motivated us to conduct functionally-grounded evaluation, in which some inherent property of the explanation is evaluated, without input from a human user. Though such evaluations do not reveal the usefulness of the explanation to humans, they are often an essential step in determining the fitness of an explainable method to a dataset and context [1]. A key evaluation measure is *explanation stability*, which is used to assess the consistency of explanations generated for an opaque predictive model [15]. Few methods to measure stability have been proposed in XAI literature, most of which are specific to a single explainable method (such as in [15]), and do not enable comparison between explainable methods. In addition, to the best of our knowledge, no studies have attempted to evaluate explanation stability for process predictions.

In this paper, we aim to use methods and metrics from the field of feature selection to evaluate explanation stability for business process predictions. The evaluation focuses on the stability of *local, post-hoc* explanations, which are provided to individual predictions by an explainable method *after* a predictive model is trained. We apply the proposed metrics to LIME [10] and SHAP [5] in the context of process predictions using real-life event logs.

Furthermore, since event log data is both temporal and case-based, extensive feature construction methods are required to make this data machine readable [14]. Therefore, of particular interest in PPA are not only the dataset and predictive model, but also feature construction techniques used. We aim to understand how the characteristics of this pipeline affect explanation stability. Hence, we design experiments by varying the event log datasets, feature construction methods and classification algorithms used to train a business process predictive model along the pipeline.

Thus, our contributions are two-fold. Firstly, we propose and demonstrate that metrics to evaluate the stability of feature selection algorithms can be used to evaluate the stability of explanations for tabular data such as event logs. Secondly, we apply these metrics to explanations of process predictions to determine the PPA-specific characteristics that affect explanation stability, and in doing so, derive insights into the use of explainable methods for PPA.

2 Background and Related Work

2.1 Process Execution Event Logs

Process execution event logs (or simply *event logs*) are a form of sequential data in tabular format. During business process execution, information is recorded in information systems in the form of event logs. Event log data include the activities that were undertaken (*event*), and the actors, systems and data involved in each event. Events are linked to a particular execution of the process (*process instance* or *case*) through some specific case identifier such as patient ID or order ID. Events form the rows of an event log, in order of occurrence, and attributes associated with an event (*event attributes*), such as case identifier, actors participating in the event, event name or timestamp, form the columns. These attributes may be *static* and unchanging over the course of the case, such

as the case identifier, or *dynamic*, such as the timestamps of events. A *trace* is a sequence of events for the same case, and *prefixes* are the features constructed for each trace using both events and event attributes.

2.2 Explainable AI

The field of explainable AI (XAI) has arisen as a means to provide transparency into otherwise opaque predictive models. Although more complex and sophisticated predictive models may be more accurate, this internal complexity also reduces the ability of human agents to understand their decision-making processes, thus requiring interpretation [3]. In this work, we are interested in local, post-hoc explanations – i.e. explanations provided to individual predictions or small data neighbourhoods (*local explanations*) by explainable methods after the predictive model is trained (*post-hoc*) [3]. A variety of explainable methods exist within this category, among which LIME and SHAP well-known and popular. Both provide feature attribution explanations, wherein they determine the contribution of each feature to the final outcome, though they use different mechanisms to determine feature importance. LIME creates a surrogate model to mimic the black box model's behaviour within a particular data neighbourhood, and uses this surrogate model to determine local feature importance [10]. SHAP's approach is based on game theory and attempts to identify the marginal contribution of each feature to the final output of the predictive model for a single instance [5].

2.3 Explainable Predictive Process Analytics

PPA attempts to predict a future state of process instances using prefixes. Common prediction targets in PPA include case outcome prediction, remaining time prediction, next activity prediction and risk prediction, among others [6]. Given the complexity of machine learning models needed for process predictions, as well as the extensive processing required to extract algorithm-readable features from event log data, process predictive models are highly opaque to human agents [13]. Most attempts at explaining or refining process prediction black boxes in literature have generally attempted to use existing post-hoc methods, including LIME [11,13], SHAP [2,11] and Partial Dependence Plots, a method to generate global explanations capturing the overall model behaviour [7].

2.4 Evaluating Explanation Stability

Explanation stability measures the consistency of explanations generated for identical or similar instances in the data [15]. Since explainable methods attempt to provide insight into otherwise "black box" models, the provided explanations must be reliable. But, when the explainable method is subject to randomness, there may be variations in the explanation, calling its reliability into question [4]. Though stability metrics have been proposed for post-hoc explainable methods,

these are often specific to a particular explainable method (for example, the metrics proposed in [15] for LIME).

We propose that measures and metrics to assess the stability of feature selection algorithms can be adapted for explainable methods. Feature selection algorithms are used to reduce the dimensions of high-dimensional datasets by determining feature relevance [9]. The outputs of these algorithms – feature subsets, feature rankings or quantification of feature relevance [8] – are similar to feature attribution explanations. Thus, we suggest that approaches and metrics to evaluate feature selection algorithms can be applied in XAI, particularly when evaluating feature attribution explanations.

3 Methods and Metrics

3.1 Evaluation Method

We propose an approach to evaluate the stability of explanations generated by post-hoc explainable methods for business process predictions. Figure 1 depicts an overview of this approach, as well as the standard workflow for building process predictive models using machine learning algorithms [14].

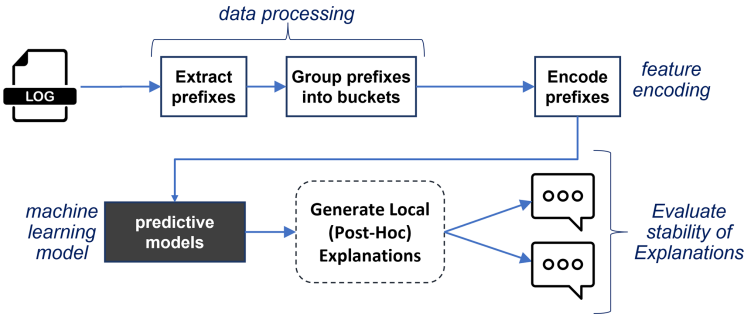


Fig. 1. Approach for evaluating explanation stability for process predictions

Firstly, prefixes are extracted for each trace in the event log, then grouped into *buckets* based on their similarities, such as length or last completed event. The prefixes in each bucket are then encoded into algorithm-readable feature vectors of equal length, and one model is trained per bucket. Once the predictive model/s have been created, local explanations are generated using post-hoc explainable methods for a sample of data. The sample of data for evaluating stability are randomly chosen, primarily from the testing set, but also from the training set when the testing set is small. Around 50 samples are chosen at each prefix length used, though fewer were chosen for the smaller datasets.

During evaluation, we measure the stability of the subset of most important features (*stability by subset*) and the stability of weights applied to each feature (*stability by weight*). Ten explanations are generated for each instance in the

sample of data used for evaluation (i.e. $M = 10$ for each instance, see Eqs. 1 and 2 in Sect. 3.2). This follows the general approach in [15], where the stability of variables used by LIME's surrogate models, and the coefficients applied to them, were measured across 10 surrogate models.

We do not specify a certain number of features to measure stability by subset. Rather, we use the features with feature weights that fall into the top quartile of the feature weight distribution. For example, if the feature weights in an explanation range from 0 to 1, only features with feature weights greater than 0.75 are used to evaluate *stability by subset* (see Eq. 1). *Stability by weight* is evaluated using the weights for all features (see Eq. 2).

3.2 Evaluation Metrics

We propose two explanation stability evaluation metrics. Both are applied to test explanation stability for a single instance, but can be averaged out to understand stability at the dataset level.

Stability by Subset. This metric was proposed in [9] to determine the stability of feature selection algorithms, based on the presence or absence of each feature across a number of feature subsets. We calculate the stability of feature subsets ($\phi(\mathcal{Z})$) for a single process instance in an *event log* as follows:

$$\phi(\mathcal{Z}) = 1 - \frac{\frac{1}{d}\sum_{i=1}^{d} s_{f_i}^2}{\frac{\overline{k}}{d}(1 - \frac{\overline{k}}{d})} \tag{1}$$

where:

- d = number of features encoded from event attributes in the log
- M = number of explanations generated for the process instance
- $\mathcal{Z}$ = binary matrix of size M x d. Each row of the binary matrix represents a feature subset from a single explanation, where a 1 at the i^{th} position means feature f_i is among the most relevant and a 0 means it is not.
- k = number of most relevant features, where relevance or level of importance is determined by an explanation generated for the process instance, for a single explanation
- $\overline{k}$ = average of k across all M explanations for the process instance
- $s_{f_i}^2$ = sample variance of the presence of feature f_i across all M explanations for the process instance (i.e. the the variance of column i in $\mathcal{Z}$)

This measure is bounded between 0 and 1, where 0 indicates no similarity in the feature subsets, and 1 indicates that all subsets are identical.

Stability by Weight. Pearson's correlation coefficient is generally used to measure stability of feature weights in feature selection algorithms [8], but this measures the similarity of trendlines and does not calculate the degree by which a feature's weight may vary. As such, we specify the measure *stability by weight*

– adapted from the statistical measure of relative variance – and calculate the stability of feature weights ($\phi(\mathcal{W})$) for a single process instance in an *event log* as follows:

$$\phi(\mathcal{W}) = 1 - \frac{1}{d}\sum_{i=1}^{d}\frac{\sigma_{w_i}^2}{|\mu_{w_i}|} \tag{2}$$

where:

- d = number of features encoded from event attributes in the log
- M = number of explanations generated for the process instance
- $\mathcal{W}$ = matrix of size M x d. Each row of the matrix records the weight of each feature as quantified by a single explanation
- μ_{w_i} = mean of the weights of feature f_i across all M explanations for the process instance (i.e. the mean of column i in $\mathcal{W}$)
- $\sigma_{w_i}^2$ = variance of the weights of feature f_i across all M explanations for the process instance (i.e. the variance of column i in $\mathcal{W}$)

This measure also has an upper bound of 1 (indicating perfect stability), but no lower bound. The suitability of these metrics will be assessed through comparison to previous results in literature and known behaviours of the explainable methods used.

4 Experimental Design

4.1 Predictive Models

The chosen prediction target for the experiments was the process outcome. This is a common prediction problem in PPA and a typical example of a classification problem. Two algorithms were used to create the predictive models. One is XGBoost which generally produces the most accurate models for outcome-oriented prediction [14]. Given that an aim of this work was to understand the effects of predictive model on explanation stability, a second prediction algorithm of different characteristics was also chosen. Logistic regression (Logit) is simpler in comparison to the significantly more complex models created by XGBoost, but generally produces less accurate models for outcome prediction.

Three combinations of bucketing and encoding were used to construct features when creating the classifiers:

- Aggregate encoding for dynamic attributes with prefix-length bucketing
- Index-based encoding for dynamic attributes with prefix-length bucketing
- Aggregate encoding for dynamic attributes compiled in a single bucket

In the single bucketing method, all data is compiled as one and a single classifier is trained on this bucket. When prefix-length bucketing is used, data is grouped based on the number of activities that have already been completed in a process instance (the prefix length), and one model is trained for each bucket.

Three different types of encoding are used. Static encoding, where numeric attributes are used as-is and categorical attributes are one-hot encoded, was

applied to static attributes in all combinations of bucketing and encoding. Aggregate and index-based encoding were applied for dynamic attributes. Aggregate encoding summarises each case, with a single feature indicating frequency of occurrence for each categorical attribute and four features (mean, maximum, minimum and standard deviation) for each numeric attribute. If index-based encoding is used, numeric attributes are encoded as-is and categorical attributes are one-hot encoded at each index (prefix in the process trace). As such, out of the three methods used, combining prefix-length bucketing with index-based encoding best preserves the temporal information in event logs, while using single buckets with aggregate encoding preserves the least.

Two explainable methods are evaluated in this work. SHAP and LIME, two popular post-hoc interpretation methods, were chosen given their relative popularity in explaining process predictions [2,11,13].

We will assess the suitability of the described metrics based on past stability evaluation results in literature. Instability is a known issue of LIME. To generate instances to train the surrogate model, LIME randomly samples the neighbourhood of the input instance to derive a set of perturbed inputs [10]. This random sampling results in a different set of perturbed instances for every explanation, and so the surrogate model and the resulting explanation lack stability, a problem compounded as the length of the input increases [12]. On the other hand, SHAP optimises the interpretation mechanism for certain categories of predictive models, such that they examine the model directly [5]. We will use two such optimisations (TreeSHAP and LinearSHAP). The lack of randomisation in the interpretation mechanism should result in little to no instability in the explanation. Therefore, the metrics can be judged to be appropriate if the following are observed:

1. LIME's explanations will become more unstable as the length of the input increases; and
2. SHAP's explanations show little to no instability.

4.2 Datasets

We use three open-source, real-life event logs. Each event log is from a different domain and has different characteristics (see Table 1 for summary of the three event logs used).

The Production dataset[1] is derived from a manufacturing process. This event log has the fewest cases and the shortest traces out of the three event logs. When using this dataset, we attempt to predict whether at least one work order in the case will be rejected (which occurs in around 55% of cases). This dataset also has a substantial number of attributes, more dynamic that static.

The Sepsis Cases dataset[2] records patients' journeys in a hospital. Using this dataset, we attempt to predict whether a patient returns to the ER within 14 days of discharge, which only 16% do. As such, this dataset was balanced through down-sampling before model training. This dataset also contains a relatively

[1] https://doi.org/10.4121/uuid:68726926-5ac5-4fab-b873-ee76ea412399.
[2] https://doi.org/10.4121/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460.

Table 1. A summary of statistics of three event log datasets

Event log		Production	Sepsis cases	BPIC2012
Description		A manufacturing process	Hospital event log showing sepsis cases	Loan application process
No. of cases (before prefix extraction)		220	782	4,685
Proportion of positive cases		55.0%	16.0%	53.4%
Maximum prefix length		23	29	40
Prefix lengths used		1–20	1–20	1–20
Feature vector length	Single bucket & aggregate encoding	166	272	133
	Prefix-length buckets & aggregate encoding	Min: 144 Max: 164	Min: 175 Max: 212	Min: 43 Max: 133
	Prefix-length buckets & index-based encoding	Min: 110 Max: 964	Min: 146 Max: 495	Min: 11 Max: 1257

large number of static attributes, but fewer dynamic attributes, so it produces comparatively longer feature vectors when using aggregate encoding, but shorter feature vectors at higher prefix lengths when using index-based encoding.

The BPIC2012 event log[3] follows a loan process. When using this event log, we attempt to predict whether the loan application is accepted (roughly 53% are rejected). This event log only has one static attribute and several dynamic attributes for each event. As such, it will have comparatively short feature vectors when using aggregate encoding, but comparatively long feature vectors at higher prefix lengths when using index-based encoding.

As a summary, each combination of the above bucketing methods, encoding methods, predictive models and explainable methods are evaluated for each dataset. Only a maximum of 20 prefixes are used to train and explain a predictive model. Each event log was split into training and testing sets (80-20 ratio) prior to feature construction. The split was temporal, such that the cases that finished the earliest were used for model training and the remaining 20% was used as the testing set.

All relevant code associated with the experiments, including the feature construction methods, hyperparamter optimisation, model training and explanation generation and evaluation, are available at https://git.io/Jc9Az.

5 Results and Analysis

5.1 Results and Observations

For SHAP, all experiments return 1.0000 for each stability metric. It is by far the more stable explainable method, both by subset and by weight, producing *perfectly* stable explanations regardless of the dataset, feature construction methods or classification algorithm used. On the other hand, LIME's stability was more variable, and often poor (see Tables 2 and 3).

[3] https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f.

Table 2. Stability by Subset results for LIME (averaged over the dataset)

Classifier	Data encoding	Production	Sepsis cases	BPIC2012
XGBoost	Single bucket & aggregate encoding	0.3959	0.2166	0.8135
	Prefix-length buckets & aggregate encoding	0.6660	0.4067	0.3790
	Prefix-length buckets & index-based encoding	0.5010	0.3520	0.1987
Logit	Single bucket & aggregate encoding	0.8417	0.7260	0.8734
	Prefix-length buckets & aggregate encoding	0.9789	0.7906	0.8155
	Prefix-length buckets & index-based encoding	0.8124	0.7977	0.6598

Table 3. Stability by Weight Results for LIME (averaged over the dataset)

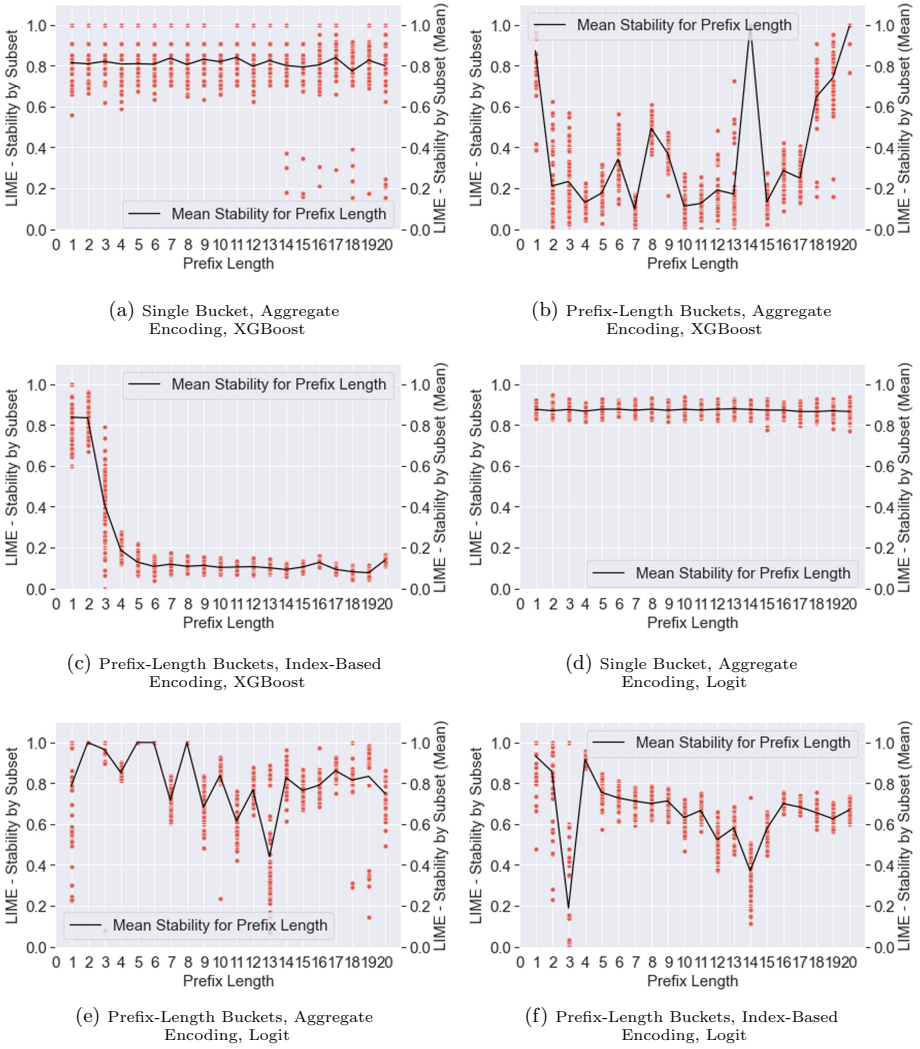
Classifier	Data encoding	Production	Sepsis cases	BPIC2012
XGBoost	Single bucket & aggregate encoding	0.5507	−0.2961	0.5415
	Prefix-length buckets & aggregate encoding	0.5682	0.4694	0.4722
	Prefix-Length buckets & index-based encoding	0.2668	0.1595	−0.1645
Logit	Single bucket & aggregate encoding	−0.0825	0.6926	0.9687
	Prefix-length buckets & aggregate encoding	0.9751	0.7915	0.9450
	Prefix-length buckets & index-based encoding	0.9415	0.8177	−0.1644

The combination of prefix-length bucketing and index-based encoding generally seems to produce the most unstable explanations when using LIME to explain predictions from the BPIC2012 and Production datasets. Using single buckets with aggregate encoding produced the least stable explanations for the Sepsis Cases dataset. The most stable combination varied between the three datasets. The most stable explanations were produced for the Production data set when using prefix-length bucketing with aggregate encoding, but when using single buckets and aggregate encoding for the BPIC2012 dataset.

5.2 Analysis and Findings

Finding 1: Causes of Instability. The returned results are as expected. SHAP is perfectly stable, while LIME shows instability. We further unfold the results for LIME by visualising the stability of explanations for each instance. Instability is closely linked to prefix length and can be seen to increase as the size of the input feature vector increases. This is apparent, both when comparing results across different bucketing and encoding methods for the same dataset and when comparing results between datasets.

For example, we unfold and examine explanation stability for the BPIC 2012 dataset in Fig. 2 and Fig. 3. When using single buckets with aggregate encoding, where the input size remains consistent, stability is also consistent (Fig. 2(a) and (d)). However, in Fig. 2(c) and (f), the results for prefix-length bucketing with index-based encoding indicate a general downward trend in stability as the prefix length increases. When considering the feature vector lengths, rather than the prefix length (Fig. 3(b) and (d)), it becomes clear that this downward trend is



(a) Single Bucket, Aggregate Encoding, XGBoost

(b) Prefix-Length Buckets, Aggregate Encoding, XGBoost

(c) Prefix-Length Buckets, Index-Based Encoding, XGBoost

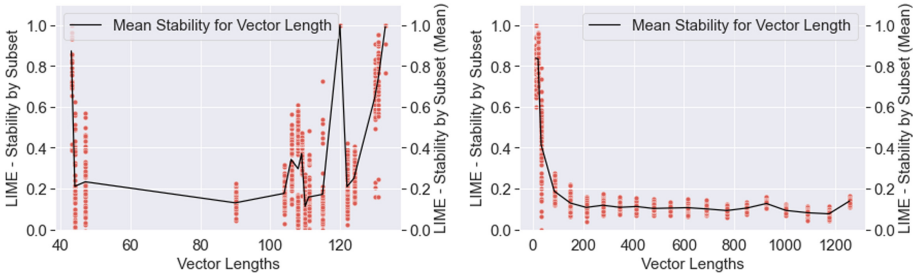
(d) Single Bucket, Aggregate Encoding, Logit

(e) Prefix-Length Buckets, Aggregate Encoding, Logit

(f) Prefix-Length Buckets, Index-Based Encoding, Logit

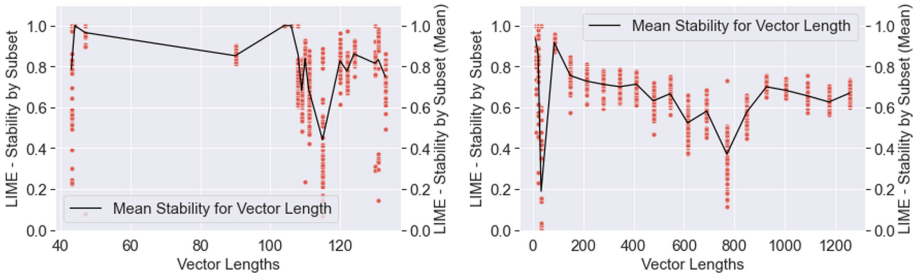
Fig. 2. The stability by subset at each prefix length for LIME using BPIC2012. Stability seems related to prefix length when using prefix-length bucketing.

related to the length of the input. As such, the metrics used can be judged to be suitable.



(a) Prefix-Length Buckets, Aggregate Encoding, XGBoost

(b) Prefix-Length Buckets, Index-Based Encoding, XGBoost



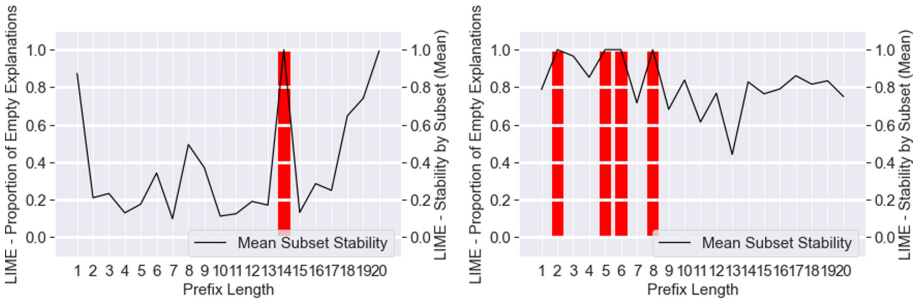
(c) Prefix-Length Buckets, Aggregate Encoding, Logit

(d) Prefix-Length Buckets, Index-Based Encoding, Logit

Fig. 3. The stability by subset at different feature vector lengths for LIME using BPIC2012. Stability generally decreases as the number of features increase.

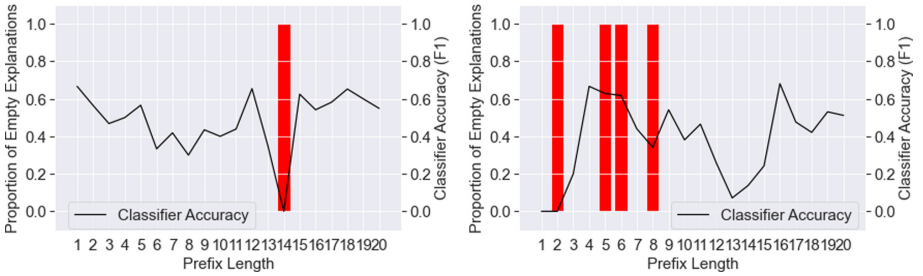
This relationship between input length and LIME stability is also true to some degree when using prefix-length bucketing with aggregate encoding (Fig. 3(a) and (c)). However, there are spikes in stability at certain prefix lengths when using this bucketing-encoding combination. This notably occurs at bucket 14 when using XGBoost (Fig. 2(b)) and at buckets 2, 5, 6, and 8 when using Logit (Fig. 2(e)), where stability does not follow the described trend. This is likely because a number of "empty" explanations with no feature attribution – where the feature weights of all features were 0 – were produced by LIME where these spikes occurred (see Fig. 4(a) and (b)).

Finding 2: Non-attributive Explanations. Non-attributive explanations, as described above, were seen in explanations for all datasets. They were primarily produced by LIME, and were extremely rare in SHAP, and occurred only when prefix-length bucketing and aggregate encoding were both used. When all explanations produced for an instance were empty, explanation stability was considered to be perfect. As such, in buckets where a large proportion of consistently empty explanations were produced, there was a noticeable spike in stability.



(a) Prefix-Length Buckets, Aggregate Encoding, XGBoost

(b) Prefix-Length Buckets, Aggregate Encoding, Logit



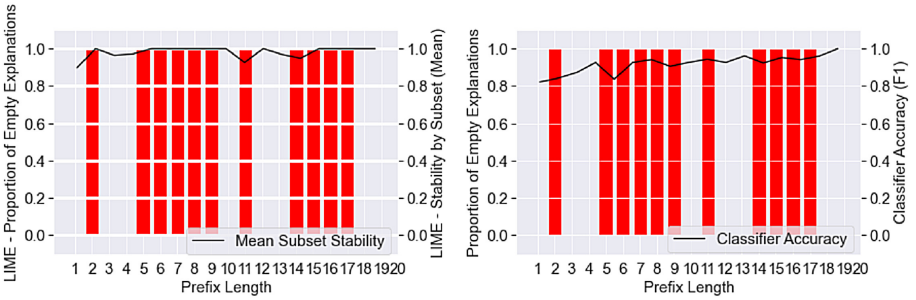
(c) Prefix-Length Buckets, Aggregate Encoding, XGBoost

(d) Prefix-Length Buckets, Aggregate Encoding, Logit

Fig. 4. The number of non-attributive, "empty" explanations generated for BPIC2012 (a and b) and its relationship to accuracy (c and d).

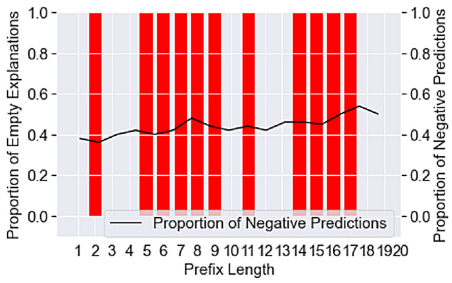
A closer investigation of this phenomena suggests that non-attributive explanations occur when model accuracy is poor. Many buckets with a high proportion of empty explanations also had a predictive model with a poor F1-score. For example, when using the BPIC2012 dataset, the XGBoost model at bucket 14 and the Logit model at bucket 2 both had F1-scores of 0, and all explanations produced for these buckets were non-attributive (see Fig. 4(c) and (d)). This also occurred when accuracy is reasonably high, but the model predicted only a single class for all or a majority of instances. This was the case for the Logit models at buckets 2 and 5 for the BPIC2012 dataset.

Therefore, non-attributive explanations for these classifiers is likely due to model underfitting. A simpler, underfit predictive model can be more easily mimicked by LIME's surrogate models than a more complex, well-fit model. Moreover, in classifiers where only a single class is predicted regardless of the input, any surrogate models produced will also disregard features. As such, when multiple explanations are created, the resulting surrogate models are identical or similar enough to ensure explanation stability.



(a) Comparison to Stability by Subset

(b) Comparison to Model Accuracy



(c) Comparison to Predictions Made

Fig. 5. A large proportion of non-attributive explanations when for the Production dataset when using Logit with using prefix-length bucketing and aggregate encoding (a), which is not related to model accuracy (b) or prediction accuracy (c).

Finding 3: Effect of Data on Non-attributive Explanations. A notable exception to this trend of non-attributive explanations, both when explaining XGBoost models and when explaining Logit models, is the Production dataset. When explaining XGBoost models, there were no non-attributive explanations generated, though some classifiers have poor quality or predict only a single class. However, when explaining Logit, explanations provided for around 60% of cases are non-attributive (Fig. 5(a)), though the accuracy of the predictive models are high (Fig. 5(b)), and no classifier predicts primarily a single class (Fig. 5(c)). This indicates that other underlying causes for non-attributive explanations also exist, though they are not immediately apparent. This also occurred in other datasets to a lesser degree. For example, the Logit model for bucket 6 of the BPIC2012 dataset had an F1-Score of 0.62 and a 0.4:0.6 ratio for the predicted class, but all explanations generated for this model were non-attributive.

It is likely that this anomalous behaviour is related to some characteristic of the Production dataset. Out of the three event logs used, Production has fewest events and cases, the shortest traces and a significant number of trace variants in comparison to the number of cases. Some form of one of these may also be present in the data used in other buckets where these exceptions occurred.

Further investigations using other datasets in needed to fully understand the causes of non-attributive explanations.

Finding 4: Effect of Feature Construction Methods on Non-attributive Explanations. We deem it to be significant that empty explanations have so far occurred only when using prefix-length bucketing with aggregate encoding. Prefix-length bucketing aims to preserve the temporal nature of business processes by sorting data based on the number of events that have occurred in the process. However, aggregate encoding is more "lossy" and preserves little of the temporal information in the event log. Firstly, although prefix-length bucketing groups cases based on events completed, this does not imply homogeneity in the traces within each bucket. If there are several variants of traces in each bucket, It is possible that this and the sparsity of data in each bucket, caused by lack of cases and use of aggregate encoding, creates poorly-fitting models.

Finding 5: Use of LIME and SHAP in PPA. It is also interesting to note that SHAP rarely provided non-attributive explanations, even when the predictive model did not appear to use any of the features in the input – that is, where the predictive model always returned the same prediction regardless of input. Given that non-attributive explanations generally appeared to indicate some problem in the underlying predictive model, this is significant. SHAP's stability and consistency may make it more suited to enable end user decision-making in PPA. However, LIME may be of more use to software engineers and data scientists in attempting to inspect and diagnose problems in the underlying process predictive models.

6 Limitations and Future Work

Past evaluations of bucketing and encoding methods and supervised machine learning models for PPA have considered their effects only on prediction accuracy [14]. However, the findings in this work emphasise the importance of the quality of explanations generated by explainable methods for machine learned process predictions. Our study also suggests that predictive model design in PPA must consider not only prediction accuracy but also compatibility with explainable methods. To this end, more extensive benchmarks are required to understand the effects of various configurations and methods used to design predictive models, as well as dataset characteristics, on explanation quality in addition to prediction accuracy.

We can be assured of the applicability of the described approach and the metrics in Sect. 3 for feature attribution explanations as they measure the stability of the output, i.e. the explanation. While the interpretation mechanism may vary across explainable methods, a feature attribution explanation will always produce a list or ranking of features, and weights associated with features. Thus, we measure the stability of these two outputs.

Future work should also consider the stability of other classes of explainable methods. The two methods evaluated in this work are both feature attribution methods, though they use different underlying mechanisms and approaches to generating explanations. Explainable methods of other classes, such as rule-based explanations, also connect features to the output. Thus, we suggest that stability by subset can also be assessed in explainable methods of classes other than feature attribution. However, this does not necessarily cover all possible aspects of the explanation in these classes. For example, the stability of the full predicates, not just features used, in rule-based explanations. As such, other classes of explainable methods, using different approaches and mechanisms of interpretation should also be considered in future works, as should a wider range of predictive models (e.g., those based on deep neural networks).

7 Conclusion

Post-hoc explainable methods are gaining popularity as a means of improving the transparency of process predictive models. However, the fitness of these methods for predictive process analytics is as yet unclear. In this work, we evaluated one aspect of explanation quality: explanation stability. We draw on research fields outside of both PPA and XAI to derive the relevant methods and metrics required for evaluation. Our result suggests that explanation stability is dependent on the characteristics of both the datasets and predictive models. We also find that, though stability may be important in supporting end-user decision-making, unexpected behaviour in explanation stability can also be useful as a diagnostic tool in determining model quality. Hence, we suggest that the choice of feature construction methods and predictive models should consider both prediction accuracy and explainable method compatibility, and as such, more extensive evaluations are required to identify suitable configurations for both.

Acknowledgements. Computational resources and services used in this work were provided by HPC and Research Support Group, Queensland University of Technology (QUT), Brisbane, Australia. The first author's research is sponsored by the Australian Government Research Training Program (RTP) Scholarship. The research is also partly supported by Centre for Data Science's First Byte Funding Program 2021 at QUT.

References

1. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning (2017). arXiv: 1702.08608v2
2. Galanti, R., Coma-Puig, B., de Leoni, M., Carmona, J., Navarin, N.: Explainable predictive process monitoring. In: 2020 2nd International Conference on Process Mining (ICPM). IEEE, October 2020
3. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(93), 1–42 (2018)

4. Guidotti, R., Ruggieri, S.: On the stability of interpretable models. In: 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019 (2019)
5. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Proceedings of the 2017 Neural Information Processing Systems Conference, Long Beach, USA, 4–9 December 2017 (2017)
6. Marquez-Chamorro, A.E., Resinas, M., Ruiz-Cortes, A.: Predictive monitoring of business processes: a survey. IEEE Trans. Serv. Comput. **11**(6), 962–977 (2017)
7. Mehdiyev, N., Fettke, P.: Prescriptive process analytics with deep learning and explainable artificial intelligence. In: ECIS 2020 Proceedings, Marrakech, Morocco, 15–17 June 2020 (2020)
8. Mohana Chelvan, P., Perumal, K.: A survey of feature selection stability measures. Int. J. Comput. Inf. Technol. **5**(1) (2016). Article No. 15
9. Nogueira, S., Sechidis, K., Brown, G.: On the stability of feature selection algorithms. J. Mach. Learn. Res. **18**(174), 6345–6398 (2018)
10. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?": explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, 13–17 August 2016 (2016)
11. Rizzi, W., Di Francescomarino, C., Maggi, F.M.: Explainability in predictive process monitoring: when understanding helps improving. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNBIP, vol. 392, pp. 141–158. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58638-6_9
12. Shankaranarayana, S.M., Runje, D.: ALIME: autoencoder based approach for local interpretability. In: Yin, H., Camacho, D., Tino, P., Tallón-Ballesteros, A.J., Menezes, R., Allmendinger, R. (eds.) IDEAL 2019. LNCS, vol. 11871, pp. 454–463. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33607-3_49
13. Sindhgatta, R., Ouyang, C., Moreira, C.: Exploring interpretability for predictive process analytics. In: Kafeza, E., Benatallah, B., Martinelli, F., Hacid, H., Bouguettaya, A., Motahari, H. (eds.) ICSOC 2020. LNCS, vol. 12571, pp. 439–447. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-65310-1_31
14. Teinemaa, I., Dumas, M., La Rosa, M., Maggi, F.M.: Outcome-oriented predictive process monitoring: review and benchmark. ACM Trans. Knowl. Discov. Data **13**(17), 1–57 (2019)
15. Visani, G., Bagli, E., Chesani, F., Poluzzi, A., Capuzzo, D.: Statistical stability indices for LIME: obtaining reliable explanations for machine learning models. J. Oper. Res. Soc., 1–11 (2021)