

Assessing the Robustness in Predictive Process Monitoring through Adversarial Attacks

Alexander Stevens^{id}, Johannes De Smedt^{id}, Jari Peeperkorn^{id}, Jochen De Weerd^{id}
Research centre for Information Systems Engineering (LIRIS)
KU Leuven, Leuven, Belgium

Abstract—As machine and deep learning models are increasingly leveraged in predictive process monitoring, the focus has shifted towards making these models explainable. The successful adoption of a model is dependent on whether decision-makers can trust the predictions and explanations made. However, recent studies have shown that deep learning models are vulnerable to adversarial attacks -small perturbations to the inputs- which trick deep learning algorithms into making incorrect predictions. An additional crucial property is that the explanations are robust against these adversarial attacks when the model decision was not affected. Therefore, this paper introduces a robustness assessment framework by investigating the impact of adversarial attacks on the robustness of predictive accuracy and explanations used in the field of predictive process monitoring. First, adversarial examples of cases in the independent test set are generated to examine the robustness of the predictive model against intentionally manipulated data. Next, the predictive models are compared with similar models trained on data imputed with adversarial attacks. We monitor the impact on predictive performance in terms of AUC at different stages of the case execution. Finally, the robustness of the explanations is calculated as the distance between the original explanations and the explanations extracted from the model trained on attacked data. We test multiple machine and deep learning techniques, namely the transparent logistic regression, random forests with Shapley values, and LSTM neural networks with attention. Results show that especially neural networks suffer from adversarial attacks, and the former two are mostly robust in terms of both predictive accuracy and explanations.

Index Terms—Robustness, Adversarial Attacks, Explainability, Outcome-Oriented Predictive Process Monitoring

I. INTRODUCTION

Predictive Process Monitoring (PPM) is a relatively new research field that is concerned with providing insights about the business processes of organizations. In this paper, the focus is on Outcome-Oriented Predictive Process Monitoring (OOPPM), which is concerned with predicting the future state of ongoing cases of processes [1], [2]. Due to the large availability of data about business processes, many sophisticated architectures such as ensemble methods [2] or deep learning models [3], [4] have been introduced in order to improve the predictive accuracy. Nonetheless, the inability to comprehend the decision-making of these models prohibits a successful adoption thereof. This has given rise to eXplainable AI (XAI) into the field of OOPPM [5]–[7] which relies on either inherently transparent models, or opaque models with post-hoc explanation techniques such as SHapley Additive exPlanations (SHAP) values [7]. A successful adoption of these models is conditional upon the trust that the decision-makers have in the

predictions and the explanations, and whether they conform to the process domain knowledge [8], [9]. This motivates the need for predictions and explanations made by accurate and trustworthy models. Despite XAI techniques’ efforts to improve the explainability, it does not address the need for trustworthiness. In the wider XAI literature, this is often obtained through measuring robustness.

Adversarial Machine Learning (AML) is a field of research that studies the robustness of algorithms against adversarial attacks, i.e. intentionally manipulated data instances. Many studies [10], [11] have already established that deep learning models are sensitive to adversarial attacks and often lead to misclassifications of *perceptively indistinguishable* instances. For example, in image recognition, adversarial attacks are composed of fine-grained pixel attacks where red-green-blue (RGB) values are changed subtly, forcing algorithms to change their predictive outcome while human interpreters would not do so. Other works [12] extend the issue of robustness, arguing that the explainability methods should be insensitive to a hardly perceptible permutation when the prediction is unaffected, meaning that similar instances with similar predictions should not lead to drastically different explanations.

The research rationale of this study is that evaluating the robustness of predictive models and their associated explainability methods against adversarial attacks contributes to explainable, accurate and trustworthy solutions. Accordingly, the key contribution of this paper is the introduction of a robustness assessment framework for OOPPM methods and their associated explainability techniques based on adversarial attacks. Moreover, the framework allows answering the following research questions:

- RQ1. How can you assess the robustness of predictive models against adversarial attacks in the field of OOPPM?
- RQ2. How robust are the explanation methods against adversarial attacks in the field of OOPPM when the predictions made by the predictive model are unaffected?

We engineered the adversarial attacks in the field OOPPM as such that they solely *indirectly* attack the control-flow attributes, by attacking the event payload data. A change in the control flow, given an underlying process model, is typically too apparent given process mining description often rely on control flow models first, similar to changing an image’s main composition where objects are put in different places from the original. Many OOPPM prediction outcomes are directly

related to the control flow as well. Hence, the focus is on fine-grained attacks similar to changing RGB values in only scarce locations, as payload data is also more hidden in the background, e.g., by many categorical attributes which have a high number of unique possible values.

The introduced framework assesses the vulnerability of the predictive model against intentionally manipulated data by attacking the independent test set (previously unseen during training). Additionally, the robustness of the predictive model is evaluated by comparing the performance of models trained on the original data and similar models trained on the adversarial attack imputed data. Both allow to measure the robustness of the predictive models in two different ways. Second, the robustness of the explanations is assessed with the Euclidean distance between the original explanations and the explanations extracted from the model trained on attacked data.

In the following Section II, the related state-of-the-art work about robustness and/or trustworthiness of predictive settings is discussed. Next, Section III gives the preliminaries for the field of OOPPM. The research methodology is given in Section IV and describes the design of adversarial attacks and the different application methods in the context of OOPPM. The real-life event logs and benchmark models, the implementation details and experimental results are given in Section V. Finally, Section VI describes the usefulness and applicability of the proposal, the limitations of the study and future work.

II. RELATED WORK

Deep neural networks are expressive algorithms that are able to learn complex tasks, but the successful adoption is often limited in high-stake decision-making processes due to their lack of robustness. In the field of AML, a distinction is often made between *adversarial examples* (i.e. adversarially attacked test instances) and *adversarial training* (i.e. using adversarial instances to train the predictive model). The former is targeted at fooling the algorithm into making incorrect predictions, while the latter measures the robustness of the predictive method against adversarially imputed training data.

Adversarial training in time-series prediction has already been investigated in [13], used to change the decision-making of dynamic Bayesian forecasting models. Similarly, permutations are performed in [14] and [12] to obtain instances that have the same label predicted, yet very different explanations. A taxonomy of adversarial examples and a number of defence mechanisms and countermeasures has been proposed in [15].

Recent works have already issued the lack of reliability of deep learning models in the context of predictive process monitoring [7], [16], [17]. In [16], the authors demonstrate the inability of Long Short-Term Memory (LSTM) models to generalize process model behaviour without careful measurement and evaluation. In previous work [7], the faithfulness of post-hoc explanations was shown to be compromised. The authors of [17] introduced an approach to train *robust and generalizable* predictive models that can handle spurious data correlations. Other works focus on incremental adaptations

of the predictive models used in predictive process monitoring [18], stating that the predictions for different stages of the case execution for are not stable *over time* and should be regularly updated. In [19], the authors fed existing prefixes to a Generative Adversarial network (GAN), which generates an *adversarial trace*. Nonetheless, it is not clear how such a generated trace looks like. The results indicated an improved robustness of accuracy scores over the prefix length. To our knowledge, no other works have investigated the impact of adversarial attacks in the field of predictive process monitoring.

III. BACKGROUND

An event log L is a collection of events grouped per case in the form of process execution traces. An event e from the event universe ξ is a tuple $e = (c, a, t, d, s)$ with $c \in C$ the case ID, $a \in A$ the activity (i.e. *control-flow* attribute) and $t \in \mathbb{R}$ the timestamp. An event additionally contains event-related attributes or payload attributes $d = (d_1, d_2, \dots, d_{m_d})$ that change during the course of the case, i.e. *dynamic* attributes. Some attributes do not evolve during the execution of a single case and are called case or *static* attributes $s = (s_1, s_2, \dots, s_{m_s})$. Hence, a trace is a sequence of events $\sigma_c = [e_1, e_2, \dots, e_i, \dots, e_n]$, with c the case ID and i the index in the trace, and is sorted based on the timestamps of the events. We denote an event e_i in a case j of the event log L as $e_{i,j} = (c_j, a_{i,j}, t_{i,j}, s_j, d_{i,j})$. The outcome y of a trace in the case of OOPPM is usually a binary attribute [20] and depends on the needs and objectives of the process owner [2].

A prefix log \mathbb{L} is extracted from the event log L and contains all the prefixes from the complete traces σ , which can be used to incrementally learn from different stages of the traces. An example of a trace prefix of case c of length l is defined as $\sigma_{c,l} = [e_1, e_2, \dots, e_l]$, with $l \leq |\sigma_c|$. Next, a sequence encoding mechanism is necessary when working with a varying amount of attributes, since each trace can have a different length. In the aggregation encoding [21] mechanism, both the activity and categorical static attributes are one-hot encoded, which means that each categorical static attribute results in a number of transformed attributes. The numeric static attributes remain unchanged. Second, the dynamic numeric attributes are replaced by their summary statistics *min*, *max*, *mean*, *sum* and *std*. Finally, frequency vectors for the dynamic categorical attributes are extracted for each of the unique attribute values. The corresponding values are the frequency of occurrence of that attribute value in the (prefix) trace. By contrast, the use of the encoding mechanism above in step-based models such as recurrent neural networks becomes superfluous given their sequential setup. To exploit this efficiently, the categorical (dynamic and static) attributes are transformed to a vector of continuous vectors through one-hot encoding.

In the following, the attributes (x_1, x_2, \dots, x_p) are used to denote the transformed attributes, following the steps above. The transformed prefix log is used to create a *predictive model* F , with the prediction for prefix (trace) i denoted as $\hat{y}_i = F(x_{i,1}, \dots, x_{i,p})$. The explanations for the predictions made by a transparent model F are obtained with the use of the

inherently estimated coefficients $w_{F,1}, \dots, w_{F,p}$, and indicate the importance of the different attributes x_i on the dependent variable. In the case of a black box model, the use of a (post-hoc) explainability method X is necessary to approximate the (unknown) attribute weights of the predictive model with the attribute weights $w_{X,1}, \dots, w_{X,p}$ of the explainability model, and the assumption is that $w_{F,j} = w_{X,j}$ for $j \in \{1, \dots, p\}$.

IV. ADVERSARIAL ROBUSTNESS IN PREDICTIVE PROCESS MONITORING

This section describes the methodology that is used to answer the research questions. First, we elaborate on the proposed adversarial attacks to generate indistinguishable prefix traces. Then, we describe the methods of application *adversarial examples* (IV-A) and *adversarial training* (IV-B). These methods are used to answer the first research question: *how can you assess the robustness of the predictive models against adversarial attacks in the field of OOPPM?* Next, IV-C describes how *adversarial explanations* are generated. This allows to answer the second research question: *how robust are the explanation methods against adversarial attacks in the field of OOPPM when the predictions made by the predictive model are unaffected?* An overview of the different attacks, methods, and evaluation is given in the robustness assessment framework in Figure 1.

The attacks were designed such that they attack the dynamic attributes $d = (d_{i,1}, d_{i,2}, \dots, d_{i,j}, \dots, d_{m_i,d})$ of a trace by performing permutations as follows. Assume a dynamic attribute $d_{i,j} \in D_j$, with D_j the set of values that the dynamic attribute ultimately can have. We denoted $D_{j,train} \subseteq D_j$ the set of values observed during the training of the model. A permutation of an attribute $d_{i,j}$ was taken at random from the set $D_{j,train}$. Given only values from the training set are drawn, we avoided any data leakage into the future.

The **Last Event Attack (A1)** consists of permuting the dynamic attributes of the last state of an incoming, independent trace. Assume a prefix trace $\sigma_{c,l} = [e_1, e_2, \dots, e_i, \dots, e_l]$. The attack A1 permutes the dynamic attributes of the last event and generates a prefix trace $\sigma_{c,l}^{A1} = [e_1, e_2, \dots, e_i, \dots, e_l^{A1}]$, with $e_i^{A1} = (c_j, a_j, t_j, s_j, d_j^{A1})$. This is visualized in Figure 1, the original dynamic attribute values are replaced with the permuted values (indicated in orange).

The **All Events Attack (A2)** consists of permuting all the dynamic attributes that related to each event in a trace. Similar to A1 (see Figure 1), a prefix trace was transformed into $\sigma_{c,l}^{A2} = [e_1^{A2}, e_2^{A2}, \dots, e_i^{A2}, \dots, e_l^{A2}]$, with $e_i^{A2} = (c_j, a_j, t_j, s_j, d_j^{A2})$.

One of the assumptions of adversarial attacks that should be carried over to OOPPM is that an indistinguishable instance, in this context a trace, should have an equal outcome to the original trace. This is analogous to the imperceptible changes made to images in image recognition and object detection,

where only a few pixels are changed, which are often not even visible to the human eye, e.g., by altering their red-blue-green (RGB) values in the background. In the case of OOPPM, the outcome of a trace is often dependent on the order of activities within the trace, defined by an LTL rule (see Section V-A). An example of such an LTL rule defines that a certain activity must always be followed by another activity. Therefore, the order of events within each case must remain untouched, as changing the order is very likely to cause the outcome of the trace to change given its direct impact on the LTL rule. This means that permutations of activity labels are not considered. Additionally, attacks on the static attributes are not considered, as modifying on case level also does not guarantee that the instances are indistinguishable, as case level attributes often have a one-to-one impact on label outcome [2]. To summarize, we focus on the imperceptible attack on the fine-granular payload which, similar to exact RGB values in images, are often not immediate apparent from process models. In addition, they are often too numerous compared to a single activity label to be immediately linked with a particular label. This does not exclude the possibility of other attacks, possibly utilizing case and/or control-flow attributes (e.g., based on expert knowledge), but the proposed attacks are, *ceteris paribus*, deemed the most logical (and intuitive) for the field of OOPPM. In the following, a distinction is made between constructing an adversarial example, adversarial training and adversarial examples in the context of OOPPM.

A. Adversarial Examples

Adversarial examples allow evaluating the vulnerability of the predictive model against intentionally manipulated ongoing traces. More specifically, adversarial examples generated with attack (A1) allow evaluating the influence of the last event dynamic attributes of the prefixes on making a correct prediction. Attack A2 indicates how important the dynamic attributes (of all the events in the prefixes) are in order to make correct predictions. We evaluate the performance by comparing the predictions for the original traces and for the adversarial examples (created with either attack A1 or attack A2). This is visualized in Figure 1.

B. Adversarial Training

Adversarial training is used to measure the robustness of the predictive method against adversarially imputed training data, by using adversarial traces instead of the original instances to train the predictive model. Assume the toy example in Figure 1, with the complete trace defined as $\sigma_{c,3} = [e_1, e_2, e_3]$. Two prefixes can be extracted, $\sigma_{c,2} = [e_1, e_2]$ and $\sigma_{c,1} = [e_1]$ respectively. The transformed prefix $\log \xi^*$ contains these three (prefix) traces. After the first adversarial attack (A1), the transformed prefix $\log \xi^*$ now contains three (prefixes) traces, i.e., $\sigma_{c,3A1} = [e_1, e_2, e_3^{A1}]$, $\sigma_{c,2} = [e_1, e_2^{A1}]$ and $\sigma_{c,3} = [e_1^{A1}]$ and the original traces are dropped. The model should still be able to learn the original behaviour, as the original events such as e_1 and e_2 can still be found in the other prefixes. The second attack (A2) permutes all the dynamic

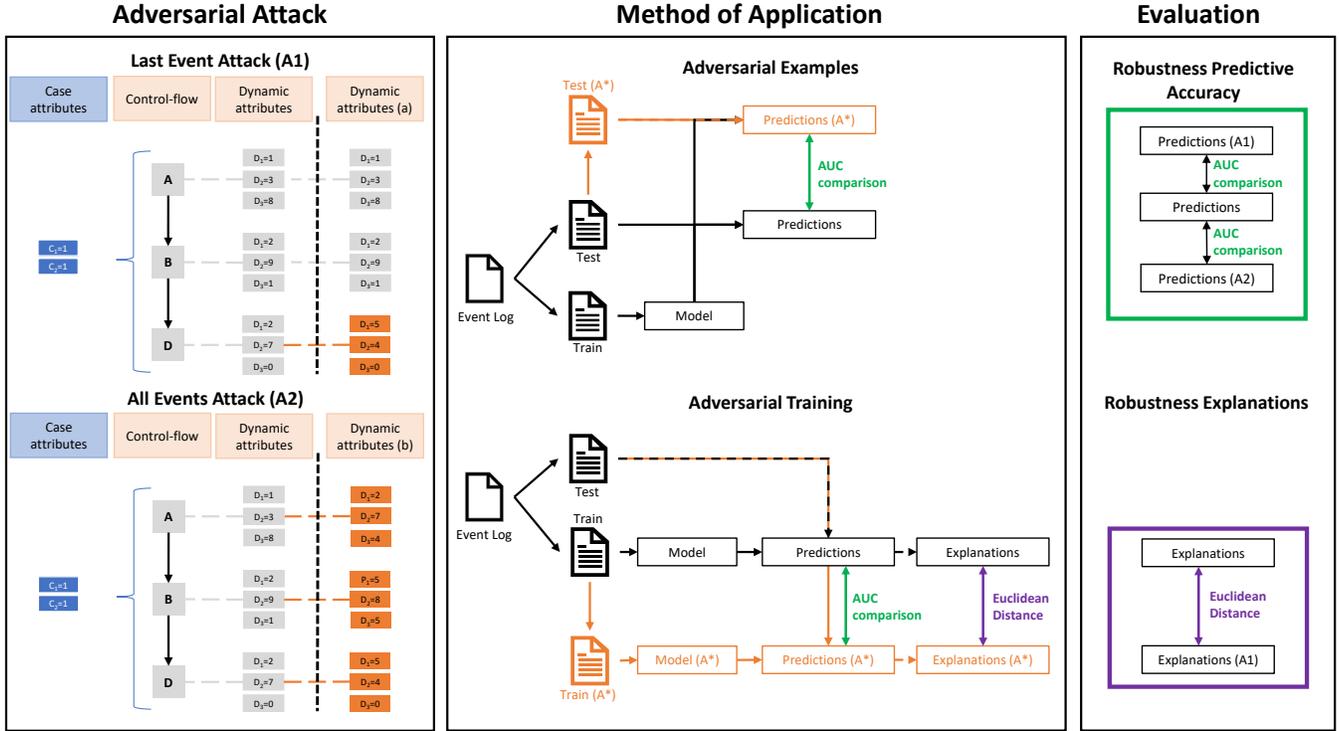


Fig. 1. The robustness assessment framework that describes the different adversarial attacks, the method of application and how the evaluation is performed. The notation A* indicates that both attack A1 and A2 can be used.

attributes of the prefix traces ξ^* used for training the model. The transformed prefix log ξ^* contains three (prefixes) traces, i.e. $\sigma_{c,3A1} = [e_1^{A1}, e_2^{A1}, e_3^{A1}]$, $\sigma_{c,2} = [e_1^{A1}, e_2^{A1}]$ and $\sigma_{c,3} = [e_1^{A1}]$. This consequently induces that the model is not able to learn the *actual* behaviour.

As shown in Figure 1, a model is trained separately on the original training data and the attacked training data. Both attacks thus gives insight into how the attacked data distorts the learning behaviour of the model by monitoring the predictive performance. While attack A1 investigates whether the model *overfits* on the last event dynamic attributes, attack A2 is useful to obtain insights about the overall importance of the dynamic attributes in the learning behaviour of the model. We evaluate the performance (AUC) by comparing the predictions made by the original model and for the adversarial attacked model.

C. Adversarial Explanations

In order to calculate the earlier mentioned robustness of explanations, we use the same setup as in IV-B. Note that we only compare the distance between explanations for test traces where the original model and the model trained on attacked data predict the same label, which is a requirement for post-hoc interpretability techniques. This is similar to the setting in related works [12], [14]. We only consider the explanation distance for adversarial training attacked with attack (A1), as the second attack (A2) permutes all the dynamic attributes, which prohibits the model to learn the correct behaviour. Additionally, this paper does not compare the explanations

between the original instances and the adversarial examples, as the distance between the explanations is dependent on the magnitude of the permutation used to create an adversarial example [10]. A possible solution could be to quantify the permutation size and incorporate this into a distance metric, but is considered out of scope for this study.

In the case of global importance coefficients such as the logistic regression coefficients, the explanation *vector* is defined as $w_{a,j} * x_{i,j}$, with w_j the coefficient weights of the model and $x_{i,j}$ the values of the instance i for the attributes x_j . The length of the explanations vectors are defined by the length of the one-hot encoding categorical attributes (i.e. order of magnitude of numerical columns is negligible). In the case of (post-hoc) explanations such as SHAP or attention values, the vector $w_{a,j} * x_{i,j}$ is calculated locally for each instance i . Intuitively, SHAP can be seen as the average marginal contribution of an attribute considering all possible combinations. Attention values, on the other hand, are calculated during runtime and are seen as the importance weights of inputs towards the output. The absolute score of these local explanations do not have a meaning, and therefore can only be relatively compared. Upon comparison between different models, it is required to normalize the explanation vectors (i.e. all the vector points of the explanation vectors are between 0 and 1). Finally, the Euclidean distance is used as a distance metric, and calculates the square root of the sum of the squared differences between the two explanation vectors.

V. EXPERIMENTAL EVALUATION

In this section, the event logs used for empirical evaluation and their specifications are described. This is followed by the benchmark models used for the analysis and implementation details. Finally, we discuss the results of the experimental evaluation.

A. Event logs

The event logs are drawn from a set of widely-used datasets for OOPPM. The first real-life event log BPIC2015 records the building permit application process. A single LTL rule is applied on the event log and split for each of the municipalities. The LTL rule defines that a certain activity *send confirmation receipt* must always be followed by *retrieve missing data*. In this paper, we use the third, fourth, and fifth municipality for the high amount of traces (BPIC2015(3) and BPIC2015(5)) and (BPIC2015(4)) for the high amount of trace variants relative to the number of traces.

The event log sepsis cases contains the discharge information of patients with symptoms of sepsis in a Dutch hospital, starting from the admission in the emergency room until the discharge of the patient. Here, the labelling is performed based on the different discharge possibilities of the patient instead of LTL rules [2], namely (1) whether the patients (eventually) admitted to intensive care (1), or (2) whether the patient is discharged from the hospital on the basis of something other than Release A (i.e. the most common release type). The last event log Production comes from a manufacturing process that contains information about the workers and machines involved in the production of an item. The outcome label is based on whether there are rejected work orders or not.

Due to the lack of domain knowledge, we assume that all these attributes can be considered as perceptively indistinguishable before and after permutation. Therefore, all the dynamic attributes are used for the attacks. The event log specifications are given in Table I.

TABLE I
EVENT LOG SPECIFICATIONS

	Traces	Events	Length	Variants	Activities	Stat./Cat. Dyn. Attr.
BPIC2015(3)	1328	57488	40	1280	380	18/12
BPIC2015(4)	577	24234	40	576	319	15/12
BPIC2015(5)	1051	54562	40	1048	376	18/12
SEPSIS(1)	782	10924	13	656	15	24/13
SEPSIS(2)	782	12463	22	709	15	24/13
PROD.	220	2489	23	203	26	3/15

B. Benchmark Models

Three predictive models will be used to benchmark the findings, covering three different approaches: statistical, machine learning, and deep learning. The Logistic Regression (LR) model is a predictive model that is often used for classification, due to its inherent transparency. Next, the Random Forest (RF) model is an ensemble model that is often used in literature due to its fast convergence and predictive accuracy. However, being an ensemble method consisting of a (often) large amounts of

decision trees, impedes its comprehensibility, and therefore a post-hoc explainability technique is required. For this, we use SHAP. Next, the attention-based bidirectional LSTM model from [5] is used. For each of the dynamic attributes, an LSTM model is trained separately, which allows obtaining *attribute attention*. Next, a separate event attention vector is constructed using each of these attribute attention vectors, to eventually obtain an event level influence to the final prediction. A final sigmoid layer allows obtaining binary predictions. Finally, the explanations are extracted from the attribute attention layers of the LSTM.

C. Implementation details

In order to prevent data leakage, a temporal split [2] on an 80/20 ratio is applied for each event log. Next, trace prefixes are extracted from the completed cases to be able to learn, preferably incrementally, from the development of the traces. Similarly to [2], trace cutting is performed before the event where 90% of the minority class has finished or before the event where the class label would be irreversibly known. The aggregation encoding is used for the statistical and machine learning models LR and RF, and the one-hot encoding accordingly as described in III. In [5], no static attributes are used in the model. Therefore, we take into account attribute attention of the static attributes by using a dense layer with *tanh* activation that is subsequently processed through a Time Distributed layer. To reduce the parameter size and avoid overfitting, the hyperparameter settings for the LSTM size for the alpha and beta layer are set to either 8 or 16. Next, a small learning rate of 0.0001 is chosen to avoid overfitting. The hyper optimization is performed with the use of hyperopt. For the LR and RF, this concerns the inverse of regularization strength number and the considered features when looking for the best split, respectively. For the LSTM, the batch size and dropout. To allow for reproducibility of the results, the code is made available on GitHub ¹.

D. Experimental Results

In the next subsections, the experimental results for the adversarial examples, adversarial training and explanations are given. This is followed by an interpretation of the overall results.

1) *Adversarial Examples*: From Table II, it is clear to see that the AUC results are almost unaffected by adversarial examples created by attack A1, meaning that the models are not vulnerable to traces where the dynamic attributes of the last event are intentionally manipulated. Furthermore, it stands out that the LSTMs seem to be the least robust against the intentionally manipulated instances created by attack A2, as in five of the six event logs, the overall AUC has dropped significantly. Remarkably, (almost) no loss in performance is observed for sepsis cases (1). This might be explained by the insights from [2], where they indicate that the short prefixes are easy to predict, but the longer prefixes are very

¹<https://github.com/AlexanderPaulStevens/RobustnessInOOPPM>

Adversarial Examples (AUC)

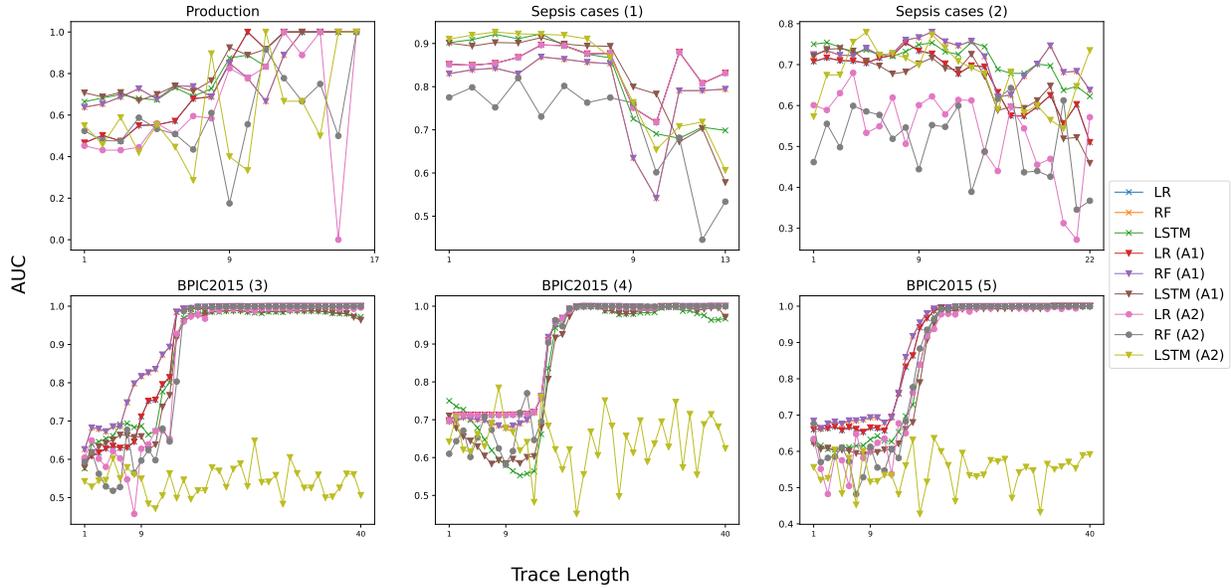


Fig. 2. Performance (AUC) for the adversarial examples

challenging. Moreover, LSTMs are able to extract sequences of activities that occur in many variants [3], which might explain why it outperforms the less advanced models. Finally, the three different models seem to be all vulnerable against the adversarial examples (A2) performed on the Production log. This is possibly caused due to the high ratio of dynamic versus static attributes [2], explaining why the adversarial attacks were successful. In Figure 2, the AUC over the different prefix lengths is given for the adversarial examples, which allows pinpointing the change in predictive performance over developing cases. Especially for the BPIC2015 logs, the neural networks are the most vulnerable against adversarial attack A2, while from a certain prediction point onward most models have an almost perfect AUC.

2) *Adversarial Training*: Next, Figure 3 and Table II show the impact on the learning behaviour of the predictive model when the model is trained on adversarially attacked data. It is clear to see that the LSTMs have drastically suffered through attack A2 for the BPIC2015 logs (AUC drop of more than 20%), while the aggregation encoded models remain relatively stable. There exists literature studying the impact of the resource involved in the execution of a case [20], [22] to the outcome of the case, meaning that permuting this dynamic attribute can have a detrimental effect on the learned behaviour (and performance) of the model. In [22], the authors state that the scheduling of the resource (i.e. which resource is assigned to the case) has an influence on the predictive accuracy of the model. However, for the event log BPIC2015, [20] states that the resource involved does not have an impactful influence, as the LTL rule that determines that outcome is rather naive. This is confirmed by the results for LR and RF, who are both rather robust against attack A1

and A2. This tells us that the LSTM model overfits on these intentionally manipulated attributes and tries to learn their dynamic behaviour. Another interesting remark is that the LR model reports an AUC of 50% for both the sepsis cases (1) and production log. An AUC of 50% means that the model is not able to outperform random guessing. Next, Table II indicates that the LR model has suffered through attack A2 for the event log BPIC2015(4), while the LR is robust against attack A2 for the other BPIC2015 logs. A possible explanation is that the model is not able to generalize over the lower amount of traces in BPIC2015 (4) compared to the other BPIC2015 logs. This might additionally explain the inability of any model to remain robust against attack A2 for the Production log, together with the fact the resource is an important attribute to determine the number of rejected work orders.

3) *Explanations*: Explanation methods should be insensitive to a hardly perceptible permutation when the prediction is unaffected, meaning that similar instances with similar predictions should not lead to drastically different explanations. In the context of this paper, this means that the Euclidean distance between the explanations extracted from the original model and the model trained on the attacked data should ideally be zero, and is given in Figure 3. As mentioned above, we take into only the distance (scaled with natural logarithm \log) between the original model and attack 1. From the results, it is clear that the logistic regression model obtains the most robust explanations for all the event logs. A possible explanation exists in the fact that the coefficients of the logistic regression are fixed coefficients for all the different instances (as these are global importance coefficients), while the Shapley values and attention values are generated locally. Nonetheless, the explanation distances of the Shapley values are seemingly

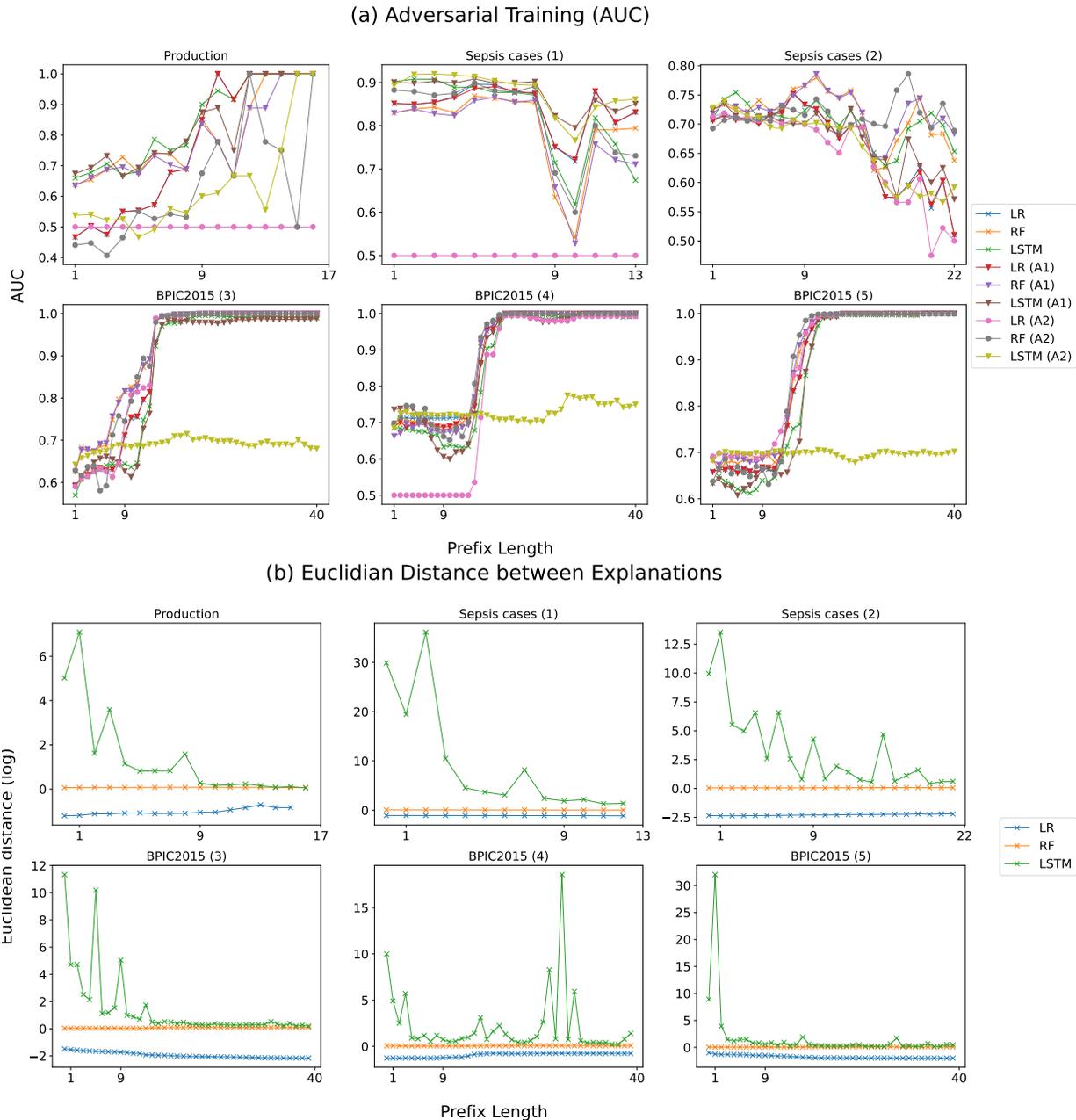


Fig. 3. The AUC over the length of the prefixes (a) and the Euclidean distance between the explanations created by the last event attack (b). Note that the natural logarithm of the Euclidean distance values are used to improve visibility of the graphs (hence the negative values).

robust over the length of the prefixes, while we observe high distances between the attention values of short prefixes, and the distances seem to converge for longer prefixes (apart from BPIC2015 (4)). Intuitively, this boils down to the fact that the impact of the adversarial attack A1 is higher for short prefixes, and the attention values generate more robust explanations for longer prefixes.

E. Interpretation

From the results, it becomes apparent that LSTMs are overfitting to some extent and that the aggregation encoding is possibly already providing a countermeasure against trace-

wide attacks such as A2. Given the relatively strong robustness to A1, it seems that most models generalize to some extent, yet mostly LR and RF are capable of providing robust explanations. The LR model seems to provide the most robust explanations, but seems to drastically fail for two event logs (sepsis cases (2) and production), meaning that the RF model provides the best trade-off in terms of predictive accuracy and explainability in light of robustness. This is in agreement with the idea that the mechanism behind random forests, i.e. *bagging*, by aggregating estimates of multiple predictors increases accuracy and stability.

TABLE II
OVERALL AUC SCORES FOR THE ADVERSARIAL TRAINING
(ADVERSARIAL EXAMPLES) FOR THE ORIGINAL MODEL, LAST EVENT
ATTACK (A1) AND ALL EVENTS ATTACK (A2).

	BPIC15(3)	BPIC15(4)	BPIC15(5)	SEPSIS(1)	SEPSIS(2)	PROD.
LR	0.96(0.96)	0.94(0.94)	0.94(0.94)	0.87(0.87)	0.69(0.69)	0.63(0.63)
LR (A1)	0.96(0.96)	0.94(0.94)	0.94(0.94)	0.87(0.87)	0.69(0.69)	0.63(0.63)
LR (A2)	0.96(0.89)	0.79(0.94)	0.95(0.89)	0.50(0.87)	0.69(0.55)	0.50(0.54)
RF	0.96(0.96)	0.94(0.94)	0.95(0.95)	0.84(0.84)	0.75(0.75)	0.71(0.71)
RF (A1)	0.96(0.96)	0.94(0.94)	0.95(0.95)	0.83(0.84)	0.75(0.75)	0.70(0.71)
RF (A2)	0.96(0.92)	0.94(0.91)	0.95(0.90)	0.86(0.78)	0.73(0.56)	0.53(0.54)
LSTM	0.94(0.92)	0.90(0.90)	0.93(0.91)	0.86(0.87)	0.72(0.73)	0.75(0.75)
LSTM (A1)	0.93(0.92)	0.91(0.88)	0.93(0.92)	0.88(0.87)	0.70(0.69)	0.76(0.75)
LSTM (A2)	0.67(0.54)	0.71(0.63)	0.68(0.54)	0.88(0.88)	0.70(0.68)	0.56(0.52)

VI. CONCLUSION

The successful adoption of the current OOPPM methods is dependent on whether the stakeholders can trust the predictions and explanations made. The goal of this paper is to evaluate the robustness of OOPPM methods and their associated explainability methods against adversarial attacks with the use of the introduced robustness assessment framework. This guides the practitioner towards obtaining models that provide explainable, accurate and trustworthy solutions and consequently ensures that the obtained insights are reliable for the business user, highlighting the practical benefit associated with this study.

Furthermore, by applying the framework to multiple real-life event logs, some conclusions could be drawn. In certain situations, the predictive models used for OOPPM purposes seem to be vulnerable against *rather naively engineered* adversarial attacks, questioning their reliability for high-stake decision-making. Furthermore, the results show that especially the associated attention mechanism for LSTM neural networks are vulnerable against adversarial attacks performed on the training data. Therefore, this framework consequently advises the use of (accurate) machine learning models, where the use of the aggregation encoding mechanism is possibly already providing a countermeasure against trace-wide attacks such as A2. Although the LR model has the most robust explanations, the RF model provides the best trade-off in terms of predictive accuracy and explainability in light of robustness.

Future work exists out of extending the experimental analysis with more algorithms and event logs. Next, we will focus on improving the robustness of the model by creating defend mechanisms against adversarial attacks (for both machine and deep learning models). Furthermore, multiple types of adversarial attacks might be introduced on top of the two *naively engineering* attacks that are proposed in this work. Next, the introduction of a more sophisticated distance metric that is able to incorporate the permutation distance into Euclidean space, which would allow measuring the distance between explanations after adversarial attack A2. Finally, we should compensate for log-size related biases during the learning of the model, especially for small sized event logs.

REFERENCES

[1] F. M. Maggi, C. D. Francescomarino, M. Dumas, and C. Ghidini, "Predictive monitoring of business processes," in *International conference*

on advanced information systems engineering. Springer, 2014, pp. 457–472.

[2] 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)*, vol. 13, no. 2, pp. 1–57, 2019.

[3] W. Kratsch, J. Manderscheid, M. Röglinger, and J. Seyfried, "Machine learning in business process monitoring: a comparison of deep learning and classical approaches used for outcome prediction," *Business & Information Systems Engineering*, vol. 63, no. 3, pp. 261–276, 2021.

[4] N. Tax, I. Verenich, M. L. Rosa, and M. Dumas, "Predictive business process monitoring with lstm neural networks," in *International Conference on Advanced Information Systems Engineering*. Springer, 2017, pp. 477–492.

[5] B. Wickramanayake, Z. He, C. Ouyang, C. Moreira, Y. Xu, and R. Sindhgatta, "Building interpretable models for business process prediction using shared and specialised attention mechanisms," *Knowledge-Based Systems*, vol. 248, p. 108773, 2022.

[6] R. Galanti, B. Coma-Puig, M. de Leoni, J. Carmona, and N. Navarin, "Explainable predictive process monitoring," in *ICPM*. IEEE, 2020, pp. 1–8.

[7] A. Stevens and J. De Smedt, "Explainable predictive process monitoring: Evaluation metrics and guidelines for process outcome prediction," *arXiv preprint arXiv:2203.16073*, 2022.

[8] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers *et al.*, *Fundamentals of business process management*. Springer, 2013, vol. 1.

[9] C. Hsieh, C. Moreira, and C. Ouyang, "Dice4el: interpreting process predictions using a milestone-aware counterfactual approach," in *2021 3rd International Conference on Process Mining (ICPM)*. IEEE, 2021, pp. 88–95.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[11] T. Wu, X. Wang, S. Qiao, X. Xian, Y. Liu, and L. Zhang, "Small perturbations are enough: Adversarial attacks on time series prediction," *Information Sciences*, vol. 587, pp. 794–812, 2022.

[12] D. Alvarez-Melis and T. S. Jaakkola, "On the robustness of interpretability methods," *arXiv preprint arXiv:1806.08049*, 2018.

[13] R. Naveiro, "Adversarial attacks against bayesian forecasting dynamic models," *arXiv preprint arXiv:2110.10783*, 2021.

[14] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3681–3688.

[15] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[16] J. Peeperkorff, S. vanden Broucke, and J. De Weerd, "Can deep neural networks learn process model structure? an assessment framework and analysis," in *Process Mining Workshops*. Cham: Springer International Publishing, 2022, pp. 127–139.

[17] P. Venkateswaran, V. Muthusamy, V. Isahagian, and N. Venkatasubramanian, "Robust and generalizable predictive models for business processes," in *International Conference on Business Process Management*. Springer, 2021, pp. 105–122.

[18] W. Rizzi, C. Di Francescomarino, C. Ghidini, and F. M. Maggi, "How do i update my model? on the resilience of predictive process monitoring models to change," *Knowledge and Information Systems*, vol. 64, no. 5, pp. 1385–1416, 2022.

[19] F. Taymouri, M. L. Rosa, S. Erfani, Z. D. Bozorgi, and I. Verenich, "Predictive business process monitoring via generative adversarial nets: the case of next event prediction," in *International Conference on Business Process Management*. Springer, 2020, pp. 237–256.

[20] J. Kim, M. Comuzzi, M. Dumas, F. M. Maggi, and I. Teinemaa, "Encoding resource experience for predictive process monitoring," *Decision Support Systems*, vol. 153, p. 113669, 2022.

[21] M. De Leoni, W. M. van der Aalst, and M. Dees, "A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs," *Information Systems*, vol. 56, pp. 235–257, 2016.

[22] A. Senderovich, M. Weidlich, A. Gal, and A. Mandelbaum, "Mining resource scheduling protocols," in *International Conference on Business Process Management*. Springer, 2014, pp. 200–216.