

Beyond Yes or No: Predictive Compliance Monitoring Approaches for Quantifying the Magnitude of Compliance Violations

Qian Chen^[0009-0004-1326-8898]¹, Stefanie Rinderle-Ma^[0000-0001-5656-6108]¹,
and Lijie Wen^[0000-0003-0358-3160]²

¹ Technical University of Munich, Garching, Germany
{qian.chen, stefanie.rinderle-ma}@tum.de

² Tsinghua University, Beijing, China
wenlj@tsinghua.edu.cn

Abstract. Most existing process compliance monitoring approaches detect compliance violations in an ex post manner. Only predicate prediction focuses on predicting them. However, predicate prediction provides a binary yes/no notion of compliance, lacking the ability to measure to which extent an ongoing process instance deviates from the desired state as specified in constraints. Here, being able to quantify the magnitude of violation would provide organizations with deeper insights into their operational performance, enabling informed decision making to reduce or mitigate the risk of non-compliance. Thus, we propose two predictive compliance monitoring approaches to close this research gap. The first approach reformulates the binary classification problem as a hybrid task that considers both classification and regression, while the second employs a multi-task learning method to explicitly predict the compliance status and the magnitude of violation for deviant cases simultaneously. In this work, we focus on temporal constraints as they are significant in almost any application domain, e.g., health care. The evaluation on synthetic and real-world event logs demonstrates that our approaches are capable of quantifying the magnitude of violations while maintaining comparable performance for compliance predictions achieved by state-of-the-art approaches.

Keywords: Predicate prediction · Outcome-oriented predictive process monitoring · Temporal constraints · Magnitude of Violation · Compliance Degree.

1 Introduction

Process compliance is a pivotal component in information systems research [1] and focuses on ensuring that processes adhere to compliance constraints that are imposed on them. Compliance constraints may stem from legislative and regulatory documents, standards and codes of practice, and business rules or contracts [10], might change frequently (e.g., financial regulations change every

12 minutes³), and non-compliance might result in the risk of reputational and financial loss and even criminal penalties [7].

Several surveys on challenges in compliance management in (process-aware) information systems exist, e.g., [1,15,16,21]. A first dimension of challenges arises from the process life cycle phase when compliance verification is supposed to take place, i.e., design time, runtime and ex post. Compliance by design [25] verifies the compliance of a process model with the set of compliance constraints. During runtime, ongoing process instances which are generated from the process model are monitored to detect or even predict compliance violations (i.e., compliance monitoring) [15]. Afterwards, the set of executed traces (an event log) are compared to a relevant process model to assess their conformity (i.e., conformance checking) [4]. Despite the importance of design and ex post compliance checking, (predictive) compliance monitoring enables business organizations to proactively detect and manage possible compliance violations during process execution [15]. However, only few existing approaches target the prediction of violations. Hence, in this work, we focus on the sub-field of compliance monitoring, i.e., *predictive compliance monitoring (PCM)* [21].

Another challenge is the lack of sufficient *ability to quantify the degree of compliance* defined as a compliance monitoring functionality (CMF10) in [15]. To predict compliance violations at runtime, most approaches [9,17,27] offer binary yes/no notions of compliance. *Outcome-oriented predictive process monitoring* [24] and *predicate prediction* [17] aim to classify each running case based on a predefined set of *categorical* outcomes, e.g., will an order be delivered on time (desired outcome) or not (undesired outcome)? However, these approaches lack the ability to quantify the extent to which an ongoing process case will deviate from the desired state. The ability to quantify the magnitude of violation is crucial, particularly for prescriptive process monitoring techniques. These techniques aim to recommend or prescribe actions to prevent undesired outcomes by minimizing a cost function of interventions. Such cost function considers factors like the probability of an undesired outcome occurring, the cost of executing interventions, and the timing of those interventions (i.e., their mitigation effectiveness) [8]. Including the magnitude of violation as a key factor in the cost function is essential, as severe deviations may require more substantial and costly interventions compared to mild ones. In addition, quantifying the magnitude of violation supports process participants in making risk-informed decisions to reduce the likelihood or severity of process faults [6]. For instance, severe violations can be prioritized for intervention to mitigate their impact, while cases with mild extent of violation may be addressed first, as they are often easier and less costly to resolve compared to severe ones.

To address these challenges, we propose two PCM approaches which extend existing predicate prediction methods with the ability to quantify the magnitude of violations. Here, the magnitude of violation corresponds to the degree of compliance [15] as utilized in literature. The hybrid approach reformulates the binary classification problem into a regression task, implicitly preserving the nature of

³ <https://thefinanser.com/2017/01/bank-regulations-change-every-12-minutes>

binary classification with a focus on violation predictions. The multi-task learning (MTL) approach handles both tasks—compliance prediction and magnitude quantification—explicitly. MTL is particularly suitable for our problem because these two tasks are inherently interdependent. Predicting the compliance status provides a broad categorization by separating normal and deviant cases, while quantifying the magnitude of violations focuses specifically on deviant cases, offering a deeper analysis. In this work, we focus on quantifying the magnitude of violation in terms of time, because meeting temporal restrictions (e.g., strict deadlines) is deemed vital in many application domains such as the logistics and aviation industry [5]. Moreover, timestamps are available in all event logs.

We evaluated our approaches based on synthetic and real-life event logs. The results show that our proposed approach, MTL in particular, is able to quantify the magnitude of violation while maintaining comparable compliance prediction performance achieved by the state-of-the-art predicate prediction approaches.

Section 2 presents a running example. Our proposed predictive compliance monitoring approaches are introduced in Sect. 3 and evaluated in Sect. 4. Related work is discussed in Sect. 5, followed by the conclusion in Sect. 6.

2 Problem Statement

We illustrate the problem based on an Order-to-Cash (O2C) process (cf. Fig. 1) as the running example. The process is triggered when a purchase order (PO) is received. An associated temporal constraint φ_t is introduced into the process model that imposes a maximum temporal distance of 24 hours between tasks `ship goods` and `confirm order`.

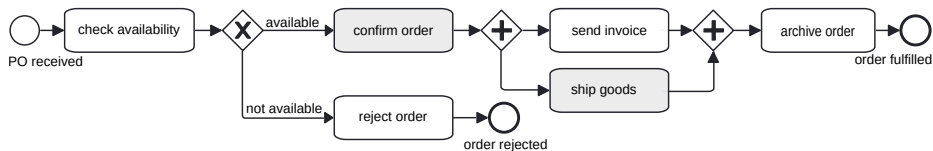


Fig. 1: Order-to-Cash Process Model

To predict compliance states of running cases w.r.t. the associated temporal constraint φ_t , existing predicate prediction approaches [9, 17, 27] classify them either into satisfaction or violation via a binary classifier. Consider the event log that has been derived from the running example in Fig. 2. Predicate prediction approaches [17] or outcome-oriented predictive process monitoring (outcome-oriented PPM) [24] first assign a boolean outcome to every case in the log according to φ_t , i.e., positive if the case violated φ_t or negative if it satisfied with φ_t . The binary classifier is then trained based on the labeled event log and applied to evolving event streams, aiming to predict which of the running case will and will not violate φ_t .

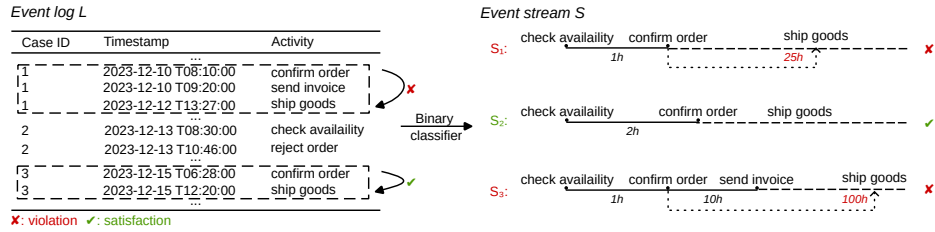


Fig. 2: Predicate Prediction

By contrast, this work does not aim only for a yes/no answer, but also at measuring the magnitude of violation that enables business organizations to gain a deeper insight into the operational performance. Consider streams s_1 and s_3 in Fig. 2 and assume a predicted duration of 25 and 100 hours between tasks **confirm order** and **ship goods** for s_1 and for s_3 , respectively. Existing predicate prediction approaches would classify both as non-compliant, but s_3 exhibits a more significant extent of deviation. Differentiating non-compliance levels of process executions can support an organization in making informed decisions to reduce or mitigate the risk of non-compliance. Potentially non-compliant process instances with a “mild degree” of violation such as s_1 with anticipated time of 25 hours, for example, can then be prioritized in light of the limited resources at disposal in order to *avoid risk*. Moreover, the magnitude of violation is considered as an important factor when predicting risks (*severity* of risks [6]) and prescribing when and how to intervene (*intervention effectiveness* [8]), with the aim to reduce the cost of non-compliance.

3 Approach

Existing predicate prediction approaches [9, 17, 24, 27] focus on predicting compliance violations in a binary yes/no notion of compliance. Hence, they lack the capability to quantify the magnitudes of violations for deviant cases. By contrast, simply framing the problem as a regression task to predict the extent of deviation which can be either positive or negative signaling satisfied or violated cases cannot emphasize violation predictions. To address this research gap we combine the prediction of the occurrence of compliance violations (classification) with predicting their magnitude (regression) for ongoing cases. To this end, we propose hybrid and multi-task learning (MTL) approaches. We first introduce the pipeline of existing predicate prediction approaches to describe the procedure for compliance predictions. The proposed approaches are presented afterwards.

3.1 Predicate Prediction Pipeline

Figure 3 shows the pipeline of the state-of-the-art methods for compliance prediction. The first step is to label each case in the event log based on the associate

constraint, and prefixes are generated and filtered based on the different requirements, e.g, only include prefixes up to a maximum length. Next, all retained prefixes can be divided into several buckets based on the length of prefixes or various clustering algorithms, and different classifiers are then trained for each such buckets. If all prefixes are considered into the same bucket, then a single classifier is trained based on all prefixes. For prefixes in each bucket, we apply different encoding techniques for attributes of the event within the trace. The most common approach is to one-hot encode categorical variables and apply normalization to numerical attributes. Finally, the predictive model is trained based on the encoded feature vectors and applied to an ongoing case to determine whether it’s compliant or not. In the following, we focus on the most critical steps—case labeling and model training—as highlighted in Fig. 3, since these steps represent the key distinctions between conventional predicate prediction approaches and our proposed methods.

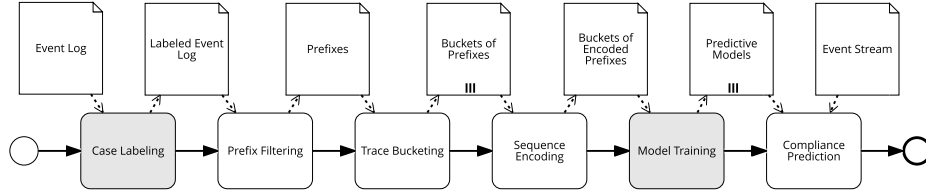


Fig. 3: Pipeline of Predicate Prediction Approaches

Case Labeling is conducted for each case in the event log based on the corresponding temporal compliance constraint. An event log $L := \{\sigma_1, \dots, \sigma_k\}$ comprises a set of completed traces σ_i where $\sigma_i := \langle e_1, e_2, \dots, e_n \rangle$ is defined as a sequence of events e_j . The prefix $hd^k(\sigma_i) := \langle e_1, e_2, \dots, e_k \rangle$ is defined as a partial trace of σ_i where $k \leq |\sigma_i|$. An event e_j , in turn, is defined as a tuple $(a, c, t, d_1, \dots, d_m)$ where a is an activity type, c is the case id, t is the timestamp of execution, and d_1, \dots, d_m are other attributes of this event. A temporal constraint φ_t specifies both the control-flow and temporal requirements each process execution must comply with. The control-flow patterns specify the occurrences or ordering of activities, such as existence and eventually follows relations. Time patterns, on the other hand, impose restrictions on the (e.g., minimum or maximum) temporal distance between the execution of activities.

The labeling function for each case w.r.t. the associated constraint φ_t is defined as follows:

$$y_{\text{binary}}(\sigma) = \begin{cases} \text{deviant} \rightarrow 1 & \text{if } \varphi_t \text{ violated in } \sigma \\ \text{normal} \rightarrow 0 & \text{otherwise} \end{cases}$$

Specifically, for a given temporal constraint φ_t , a case is labeled as deviant (encoded as 1) if it violates the constraint; otherwise, it is labeled as normal (encoded as 0). Consider the running example as introduced in Sect. 2 with φ_t :

“Goods must be shipped no later than 24 hours after order confirmation”. The labeled event log after applying φ_t to every case in the log is shown in Fig. 4. The magnitude column will be explained later in our proposed approaches.

Labeled Event log L

Case ID	Timestamp	Activity	Label	Magnitude
1	2023-12-10 T08:10:00	confirm order	deviant	29h
	2023-12-10 T09:20:00	send invoice	deviant	29h
	2023-12-12 T13:27:00	ship goods	deviant	29h
2	2023-12-13 T08:30:00	check availability	normal	0
	2023-12-13 T10:46:00	reject order	normal	0
3	2023-12-15 T06:28:00	confirm order	normal	0
	2023-12-15 T12:20:00	ship goods	normal	0

✘: violation ✔: satisfaction

Fig. 4: Labeled Event Log

Model Training takes as input the encoded feature vectors from the prefixes log and predicts their corresponding labels. In order to determine whether a running process execution is going to comply with or violate the associated temporal constraint, a simple binary classifier, e.g., decision tree (DT) or support vector machines (SVM), is often adopted as the predictive model in existing predicate prediction approaches. For a binary classification problem, we target at minimizing the binary cross-entropy loss (BCE) between the true labels y_i (either 0 or 1) and the predicted values \hat{y}_i (i.e., estimated probability within $[0, 1]$) for all (i.e., n) samples. The loss function is defined as follows:

$$\mathcal{L}_{\text{BCE}} = - \sum_{i=1}^n y_i \log(\hat{y}_i) \quad (1)$$

3.2 Hybrid Approach

In order to predict the compliance of running cases and to further assess the degree of such violations, a pure classification or regression method is not capable of achieving both goals. Thus, we present a hybrid approach which is inspired by [11] in grid distribution networks. It targets at predicting the occurrence of grid constraint violations as well as the amplitude of such violation, aiming to identify and prevent grid constraint violations for the stable and efficient operation of systems.

Case Labeling in this approach is different from the conventional predicate prediction methods as introduced in Sect. 3.1. We follow up on the idea from [11] to assign the magnitude of violation for each case in the log, extending the binary classification problem by incorporating regression to capture the degree of violations. In particular, the hybrid approach assigns a positive value to deviant cases to reflect the magnitude of violation, while normal cases are consistently

labeled as 0. For deviant cases that breach temporal constraints in the control-flow dimension, we use the case duration (i.e., the time from the start to the end of the case) as the magnitude of violation to highlight the severity of non-compliance. By doing so, it maintains consistency with the binary classification method for normal cases while offering a more nuanced evaluation of violations.

The labeling function adopted in the hybrid approach is defined as follows:

$$y_{\text{hybrid}}(\sigma) = \begin{cases} \text{magnitude of violation} & \text{if } \varphi_t \text{ violated in } \sigma \\ 0 & \text{otherwise} \end{cases}$$

Consider the running example again with φ_t imposing a maximum temporal restriction of 24 hours between `confirm order` and `ship goods`. Applying y_{hybrid} for each case in the log results in the magnitude column (cf. Fig. 4) as the target values. For example, the duration between these two events in σ_1 is roughly 53 hours based on the corresponding timestamps, then the magnitude of violation is calculated as $53 - 24 = 29$ hours.

Model Training takes the same feature vectors as input, but predicts the magnitudes of violations instead of class labels. To enable compliance predictions, we convert the continuous predictions to binary values according to the hybrid labeling function y_{hybrid} . Specifically, cases are classified as deviant if their predicted values > 0 , otherwise, they are considered normal. For the regression task, we aim to minimize the squared difference between the ground truth values y_i and predicted values \hat{y}_i by using the mean squared error (MSE) as the loss function:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2)$$

3.3 Multi-task Learning

The multi-task learning (MTL) approach is particularly suited to our problem as it allows the model to jointly learn compliance prediction (binary classification) and magnitude quantification (regression). Both tasks are closely related: the classification task identifies whether a case is compliant or deviant, while the regression task quantifies the magnitude of violation for deviant cases. By sharing knowledge across tasks, MTL exploits their interdependence, enabling the model to learn richer feature representations that improve both tasks. By doing so, this framework provides both a high-level understanding of compliance and detailed insights into the magnitude of violations, enhancing overall predictive performance. Thus, the MTL approach incorporates both the predicate prediction approach (cf. Sect. 3.1) and the hybrid technique (cf. Sect. 3.2) to predict compliance states and the magnitude of violation simultaneously.

Case Labeling for MTL approach combines labeling functions from predicate prediction (y_{binary}) and hybrid approach (y_{hybrid}) to assign labels and the magnitudes of violations for each case, resulting in a labeled event log with both label and magnitude columns as shown in Fig. 4. Thus, the integrated labeling

function is updated as follows:

$$y_{\text{mtl}}(\sigma) = \begin{cases} y_{\text{binary}}(\sigma) \\ y_{\text{hybrid}}(\sigma) \end{cases}$$

Model Training in the MTL framework addresses both classification and regression tasks simultaneously. As illustrated in Fig.5, the compliance prediction and magnitude of violation quantification tasks share the same layers, because they share common features and representations. The compliance prediction task predicts whether a case is deviant or normal, leveraging the binary cross-entropy loss function introduced in Sect. 3.1. For the magnitude of violation prediction, unlike the hybrid approach, the model incorporates the output of the compliance prediction task as an additional input, alongside the shared features. This integration is crucial because the magnitude of violation is only relevant for deviant cases, making the classification output a valuable signal for the regression task. Afterwards, it outputs either a positive value, representing the magnitude of violation, or 0, indicating satisfaction for the given case.

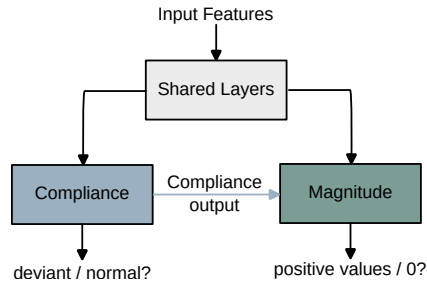


Fig. 5: Model Architecture for MTL

To train the MTL model, we minimize the total loss, which combines the losses from both compliance prediction (cf. Equation 1) and violation magnitude quantification (cf. Equation 2):

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{MSE}} \quad (3)$$

4 Evaluation

In this section, we demonstrate the efficacy of our proposed approaches by comparing them with the conventional binary predicate prediction approach (baseline). We employ the XGBoost classifier as one of the predictive models, as it has been shown to outperform other traditional machine learning models in the benchmark study of outcome-oriented PPM approaches [24]. Additionally, we incorporate a deep learning based predictive model—the Attention-based Bidirectional LSTM Neural Network (Att-Bi-LSTM)—as the second baseline due to

its superior performance compared to traditional machine learning methods [26]. Moreover, the MTL approach is well-suited for deep learning models as it requires flexible adaptation of the model architecture. This section introduces the experimental setup, including datasets and evaluation metrics, followed by data preprocessing and model adaptation. The experiment results are provided afterwards. We use the code provided by [24] for experiments with XGBoost. The implementation for Att-Bi-LSTM (source code is not provided in [26]) is available at <https://github.com/Qian915/predictive-compliance-monitoring>.

4.1 Experimental Setup

All experiments are conducted on a synthetic data set and two real-life event logs with temporal constraints involved.

Datasets. The synthetic data set `o2c` is generated from the Order-to-Cash process (cf. Fig. 1) using the Cloud Process Execution Engine⁴ (CPEE) [18]. For real-life event logs, the sepsis event log⁵ records patient pathways in a hospital for sepsis treatment, and `bpic2012w`⁶ deals with a sub-process (*work item*) within a loan application procedure.

Table 1 presents the characteristics of all datasets, and the corresponding temporal constraints are presented as follows:

`o2c_1`: “Goods must be shipped no later than 24 hours after order confirmation.”

`sepsis_1`: “Patients should be administered antibiotics within one hour after ER Sepsis Triage”

`sepsis_2`: “Lactic Acid measurements should be performed within three hours from ER Sepsis Triage.”

`bpic2012w_1`: “A scheduled validation should start within 2 days.”

`bpic2012w_2`: “The validation must be done in at most 20 minutes.”

Table 1: Characteristics of Datasets

Dataset	#traces	#event classes	Length			pos class ratio	act vio ratio
			min.	avg.	max.		
<code>o2c_1</code>	998	6	1	2.53	3	0.41	0.0
<code>sepsis_1</code>	782	15	1	7.19	61	0.63	0.2
<code>sepsis_2</code>	782	16	3	7.08	39	0.25	0.94
<code>bpic2012w_1</code>	9658	15	2	13.17	93	0.20	0.02
<code>bpic2012w_2</code>	9658	17	2	13.61	93	0.15	0.0

Metrics. For the binary compliance prediction task, we choose *area under the ROC curve (AUC)* instead of the most commonly used metric *accuracy*, because it remains unbiased even on imbalanced datasets and is threshold-independent

⁴ <https://cpee.org/>

⁵ https://data.4tu.nl/articles/dataset/Sepsis_Cases_-_Event_Log/12707639

⁶ https://data.4tu.nl/articles/BPI_Challenge_2012/12689204

[3]. AUC represents the probability that a positive instance is ranked higher than a negative one. For the regression problem of predicting the magnitude of violations, *Mean Absolute Error (MAE)* is used to evaluate the average prediction error by calculating the mean of the absolute differences between predicted and actual values. Since the binary classification approach does not provide magnitude predictions, we instead evaluate a trivial baseline approach by using the average of the actual values as predictions and calculating the mean of the absolute differences between this average and the actual values. The MAE values for our approaches and the baseline are presented as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4)$$

$$MAE_{\text{baseline}} = \frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}| \quad (5)$$

Here, y_i and \hat{y}_i are the true and predicted magnitude of violation, \bar{y} is the mean of true values, and n is the total number of test prefixes. The baseline MAE sets a reference point to assess whether our approach offers a meaningful improvement.

4.2 Data Preprocessing and Model Adaptation

All event logs are first labeled based on the corresponding labeling function as introduced in Sect. 3. For each dataset, we remove incomplete cases and include all attributes available in the event log as input features. In addition, time-related features, such as *month*, *weekday* and *hour*, are extracted based on the timestamp attribute as we also target at quantifying the magnitude of temporal violations. All traces are cut before the relevant events derived from the temporal constraint happen to allow early predictions, i.e., cut traces exactly before the corresponding class labels are known and irreversible. In this work, we do not consider the multiple occurrences of constraint-related events, thus, traces cutting can be applied even to cases with eventually follows relation involved. We keep the 80%-20% train-test-split ratio for model training and evaluation. Prefixes are extracted from the labeled event log and filtered based on a predefined maximum prefix length. For each dataset, the maximum prefix length is determined as the case length corresponding to the 90th percentile of positive cases as adopted in [24]. For our experiments, we train the predictive model based on all retained prefixes, namely prefixes are not divided into several buckets.

For data preprocessing, the main difference between XGBoost and Att-Bi-LSTM lies in how events are encoded. For XGBoost, the optimal encoding technique is aggregation: categorical attributes are represented by their frequency of occurrences, while numerical attributes are denoted with the mean and standard deviation [24]. By contrast, Att-Bi-LSTM uses one-hot encoding for categorical features and applies normalization to numerical values [26].

Model Adaptation. Traditional machine learning models do not support model adaptation. Therefore, for XGBoost, we modify only the loss function as described in Sect. 3, based on the corresponding approach. As for deep learning

models, we adapt the Att-Bi-LSTM model architecture to enable multi-task predictions (cf. Fig.6). Input features are first processed through a bidirectional LSTM layer, followed by an attention layer that identifies the most critical parts of inputs. These outputs are then passed through a global average pooling layer, a dense layer, and a dropout layer.

For the binary predicate prediction approach (baseline), all features are fed into a linear layer to predict whether the temporal constraint will be satisfied or violated in the trace. The hybrid approach also follows this structure to quantify the magnitude of violation. However, in the multi-task learning approach, compliance and magnitude predictions are performed simultaneously. In particular, for the magnitude prediction task, it takes as input the shared features concatenated with the output from the compliance prediction task representing captured patterns for separating satisfaction and violation.

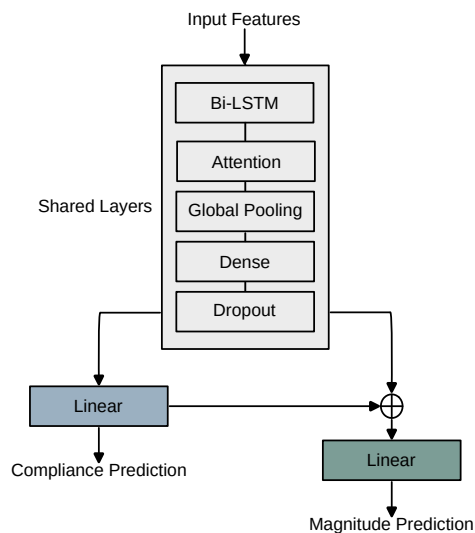


Fig. 6: Model Architecture for Att-Bi-LSTM

Furthermore, we optimize hyperparameters using the Tree-structured Parzen Estimator (TPE) algorithm [2]. Additionally, a 3-fold cross-validation is conducted for each hyperparameter configuration.

4.3 Results

We report experimental results with XGBoost (cf. Tab. 2) and Att-Bi-LSTM (cf. Tab. 3) to assess the performance of compliance prediction (AUC) and magnitude quantification (MAE in days).

XGBoost. For traditional machine learning models, such as XGBoost (cf. Tab. 2) in this study, our proposed approaches outperform the baseline in predict-

ing the magnitude of violations, achieving lower MAE values across nearly all datasets. The only exception is the hybrid approach on the sepsis datasets, where it underperforms the baseline with higher MAE values. This underperformance can be attributed to the high proportion of deviant cases which violate the temporal constraints on the control-flow dimension (cf. activity violation ratio in Tab. 1). In other words, the relevant activity type does not even occur. For such severe violations, the case duration (from the start to the end of the case) is used as the magnitude of violation, introducing a substantial number of outliers and resulting in a skewed data distribution. The hybrid approach, which focuses on regression tasks, is particularly sensitive to such imbalanced data distributions compared to the baseline (a classification approach) and the MTL approach.

Regarding compliance prediction performance, the MTL approach achieves results comparable to the baseline. However, the hybrid approach shows a significant decline in compliance prediction performance for the sepsis_2 dataset, where the activity violation ratio is extremely high at 0.94 (cf. Tab. 1). By contrast, for the o2c_1 dataset, the hybrid approach demonstrates a notable improvement, with an AUC value of 0.70. This improvement can be attributed to the synthetic dataset containing only basic attributes (case ID, activity, and timestamp) and temporal features extracted from the timestamps, which enhance the predictive performance for a regression task.

Table 2: Comparison Performance for XGBoost

Dataset	Approach	Compliance	Magnitude
		AUC	MAE
o2c_1	baseline	0.50	0.31
	hybrid	0.70	0.21
	multi-task	0.50	0.22
sepsis_1	baseline	0.57	6.06
	hybrid	0.51	8.41
	multi-task	0.57	4.59
sepsis_2	baseline	0.87	11.05
	hybrid	0.58	13.11
	multi-task	0.87	9.83
bpic2012w_1	baseline	0.56	1.42
	hybrid	0.51	1.28
	multi-task	0.55	0.91
bpic2012w_2	baseline	0.53	0.04
	hybrid	0.55	0.02
	multi-task	0.52	0.02

Att-Bi-LSTM. Table 3 summarizes the experimental results obtained with Att-Bi-LSTM. Similar to the magnitude prediction performance observed for XGBoost (cf. Tab. 2), both hybrid and MTL approaches outperform the baseline. Notably, when Att-Bi-LSTM is used as the predictive model, our proposed

approaches—particularly the MTL approach—achieve lower MAE values compared to XGBoost. This improvement stems from model adaptation in Att-Bi-LSTM (cf. Fig. 6), where the magnitude prediction task leverages patterns captured by the compliance prediction task as additional input.

The MTL approach slightly outperforms the baseline in compliance prediction. For the sepsis_2 dataset, the hybrid approach exhibits a similar decline in compliance prediction performance as observed with XGBoost (cf. Tab.2), due to the high activity violation ratio. Meanwhile, for the bpic2012w dataset, the hybrid approach underperforms other methods. This is likely due to the extremely low positive case ratios of 0.20 and 0.15 (cf. Tab.1), which result in imbalanced data distributions that negatively impact regression tasks.

Table 3: Comparison Performance for Att-Bi-LSTM

Dataset	Approach	Compliance	Magnitude
		AUC	MAE
o2c_1	baseline	0.50	0.31
	hybrid	0.50	0.23
	multi-task	0.50	0.23
sepsis_1	baseline	0.60	6.06
	hybrid	0.58	8.17
	multi-task	0.56	3.64
sepsis_2	baseline	0.86	11.05
	hybrid	0.55	9.21
	multi-task	0.87	8.67
bpic2012w_1	baseline	0.57	1.42
	hybrid	0.50	1.34
	multi-task	0.58	0.85
bpic2012w_2	baseline	0.55	0.04
	hybrid	0.50	0.03
	multi-task	0.56	0.02

Findings. In general, our proposed approaches outperform existing predicate prediction methods in quantifying the magnitude of violations, while maintaining comparable performance in compliance prediction. Specifically, when a deep learning architecture is used as the predictive model, the multi-task learning approach achieves superior results in both tasks. However, the performance of the hybrid approach is highly sensitive to data distribution. For instance, worse performance is observed on the sepsis datasets due to their skewed data distribution introduced by the extremely significant magnitudes of violations for deviant cases, which violate the temporal constraint on the control-flow perspective.

5 Related Work

In this section, we analyze related work on predicting compliance states of ongoing process instances and quantifying the degree of compliance w.r.t. a given set of compliance constraints.

Compliance Prediction. [19] classify predictive process monitoring (PPM) approaches into process-aware and non-process-aware approaches, both employing regression and classification methods to predict attributes in the future, such as next activity, next time, or the outcome of the case. As part of PPM, outcome-oriented PPM [23, 24] and predicate prediction [9, 17] focus on compliance predictions of ongoing instances regarding a given set of constraints; the binary yes or no answer w.r.t. compliance is not sufficient for compliance management and requires the ability to quantify the degree of compliance.

Compliance Degree. [14] calculate a compliance distance to indicate the degree of match between the process model and the set of control rules at design time. They define control rules into four distinct classes according to the ideal semantics: ideal, sub-ideal, non-compliant and irrelevant states. [13] classify cases into full-, partial- and non-compliance and formulate a framework to detect and evaluate the degree of violations after process executions. However, the user-defined piece-wise mapping function cannot distinguish the extent of violations in a fine-grained manner. In [20], a compliance scale model is created to measure the degree of compliance of completed process instances. Still, the mechanisms of c-semirings to classify process instances into $\langle Good, Bad \rangle$ or $\langle 0, .5, 1 \rangle$ are not accurate enough. [22] define a set of key performance indicators (KPIs) for each compliance rule and map the evaluation values of them to a compliance level from -100 to 100 taking into account values of target, threshold and worst. However, existing approaches provide various metrics for compliance degrees in different phases of process life cycle, but none of them target compliance prediction.

6 Conclusion

This work proposes two predictive compliance monitoring approaches that extend state-of-the-art binary predicate prediction methods by enabling the quantification of violation magnitudes for ongoing process executions. The hybrid approach reformulates the binary classification problem into a regression task, focusing on quantifying violations in deviant cases, which implicitly aligns with the binary classification objective. By contrast, the multi-task learning (MTL) approach explicitly incorporates both compliance prediction and magnitude quantification tasks. This design is reasonable because these two tasks are closely related, and learning them together allows for feature sharing between tasks.

The evaluation demonstrates that our approaches—particularly the multi-task learning method—outperform existing predicate prediction methods in magnitude quantification while maintaining comparable performance in compliance prediction. The performance of both tasks improves when MTL leverages deep

learning models, owing to the flexibility in adapting model architectures to better capture task-specific features. The ability to quantify the magnitude of violations is crucial in compliance monitoring, as it provides organizations with deeper insights into their operational performance and enables informed decision-makings to reduce or mitigate the risks of non-compliance.

However, our approaches have limitations. The performance of the hybrid approach is highly sensitive to data distribution, i.e., performing poorly on skewed datasets. This limitation is primarily due to the case labeling function applied to deviant cases, where the relevant event may not even occur. In future work, we aim to investigate methods for assigning the magnitudes of violation to deviant cases that breach temporal constraints in the control-flow dimension. Additionally, we plan to explore techniques to adapt the multi-task model architecture further, such as dynamically weighting each task’s loss [12], to improve overall performance.

Acknowledgments. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 277991500.

References

1. Abdullah, N.S., Sadiq, S.W., Indulska, M.: Emerging challenges in information systems research for regulatory compliance management. In: *Advanced Information Systems Engineering*. pp. 251–265 (2010)
2. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F.C.N., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*. pp. 2546–2554 (2011)
3. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* **30**(7), 1145–1159 (1997)
4. Carmona, J., van Dongen, B.F., Solti, A., Weidlich, M.: *Conformance Checking - Relating Processes and Models*. Springer (2018)
5. Cheikhrouhou, S., Kallel, S., Guermouche, N., Jmaiel, M.: The temporal perspective in business process modeling: a survey and research challenges. *Service Oriented Computing and Applications* **9**, 75–85 (2015)
6. Conforti, R., de Leoni, M., Rosa, M.L., van der Aalst, W.M.P., ter Hofstede, A.H.M.: A recommendation system for predicting risks across multiple business process instances. *Decis. Support Syst.* **69**, 1–19 (2015)
7. Elgammal, A., Sebahi, S., Turetken, O., Hacid, M.S., Papazoglou, M., van den Heuvel, W.: Business process compliance management : an integrated proactive approach. In: *Business Information Management Association*. pp. 764–781 (2014)
8. Fahrenkrog-Petersen, S.A., Tax, N., Teinemaa, I., Dumas, M., de Leoni, M., Maggi, F.M., Weidlich, M.: Fire now, fire later: alarm-based systems for prescriptive process monitoring. *Knowl. Inf. Syst.* **64**(2), 559–587 (2022)
9. Francescomarino, C.D., Dumas, M., Maggi, F.M., Teinemaa, I.: Clustering-based predictive process monitoring. *IEEE Trans. Serv. Comput.* **12**(6), 896–909 (2019)

10. Hashmi, M., Governatori, G., Lam, H., Wynn, M.T.: Are we done with business process compliance: state of the art and challenges ahead. *Knowl. Inf. Syst.* **57**(1), 79–133 (2018)
11. Junior, J., Pinto, T., Morais, H.: Hybrid classification-regression metric for the prediction of constraint violations in distribution networks. *Electric Power Systems Research* **221**, 109401 (2023)
12. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018. pp. 7482–7491. Computer Vision Foundation / IEEE Computer Society (2018)
13. Lam, H., Hashmi, M., Kumar, A.: Towards a formal framework for partial compliance of business processes. In: *AI Approaches to the Complexity of Legal Systems*. pp. 90–105 (2020)
14. Lu, R., Sadiq, S.W., Governatori, G.: Measurement of compliance distance in business processes. *Inf. Syst. Manag.* **25**(4), 344–355 (2008)
15. Ly, L.T., Maggi, F.M., Montali, M., Rinderle-Ma, S., van der Aalst, W.M.P.: Compliance monitoring in business processes: Functionalities, application, and tool-support. *Inf. Syst.* **54**, 209–234 (2015)
16. Ly, L.T., Rinderle-Ma, S., Göser, K., Dadam, P.: On enabling integrated process compliance with semantic constraints in process management systems - requirements, challenges, solutions. *Inf. Syst. Frontiers* **14**(2), 195–219 (2012)
17. Maggi, F.M., Francescomarino, C.D., Dumas, M., Ghidini, C.: Predictive monitoring of business processes. In: *Advanced Information Systems Engineering*. pp. 457–472 (2014)
18. Mangler, J., Rinderle-Ma, S.: Cloud process execution engine: Architecture and interfaces. *CoRR* **abs/2208.12214** (2022)
19. Márquez-Chamorro, A.E., Resinas, M., Ruiz-Cortés, A.: Predictive monitoring of business processes: A survey. *IEEE Trans. Serv. Comput.* **11**(6), 962–977 (2018)
20. Morrison, E.D., Ghose, A., Koliadis, G.: Dealing with imprecise compliance requirements. In: *EDOC Workshops*. pp. 6–14 (2009)
21. Rinderle-Ma, S., Winter, K., Benzin, J.: Predictive compliance monitoring in process-aware information systems: State of the art, functionalities, research directions. *Inf. Syst.* **115**, 102210 (2023)
22. Shamsaei, A.: Indicator-based policy compliance of business processes. In: *International Conference on Advanced Information Systems Engineering* (2011)
23. Teinemaa, I., Dumas, M., Maggi, F.M., Francescomarino, C.D.: Predictive business process monitoring with structured and unstructured data. In: *Business Process Management*. pp. 401–417 (2016)
24. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Trans. Knowl. Discov. Data* **13**(2), 17:1–17:57 (2019)
25. Tosatto, S.C., Governatori, G., van Beest, N., Olivieri, F.: Efficient full compliance checking of concurrent components for business process models. *FLAP* **6**(5), 963–998 (2019)
26. Wang, J., Yu, D., Liu, C., Sun, X.: Outcome-oriented predictive process monitoring with attention-based bidirectional LSTM neural networks. In: *Web Services*. pp. 360–367. IEEE (2019)
27. Zhang, J., Liu, G.: Prediction of incompliance with business goals with business-related data and context data. *IEEE Access* **8**, 187008–187020 (2020)