A Mamba-based approximate conformance checking method

Yilin Lyu*
The University of Melbourne, Grattan Street, Parkville, VIC, 3010 Australia
* Corresponding author: yillyu1@student.unimelb.edu.au

ABSTRACT

Process mining uses event flow information from log files to generate process models for better management and optimization of target processes. Approximate conformance checking quantifies the deviation of process models from process logs and is important for process compliance checking, model quality assessment and business process optimization. Existing approximate conformance checking methods are mainly classified into rule-based methods, token replay methods, and alignment-based methods. These methods usually rely on known process structures and traditional machine learning and mathematical statistics knowledge, and cannot adapt to the situation where large-scale logs and process structures are unknown. The correlation information between different traces is not effectively extracted. The effect of different length traces on coding sparsity is also not well addressed. To address the above problems, we adopt machine learning for approximate conformance checking and propose a Mamba-based method MACC. MACC efficiently improves the approximate conformance checking performance of machine learning based regressors on large-scale system logs under the scenario of unknown process structure. Specifically, we introduce the feature embedding of Mamba based classifier on the trace embedding before input into regressor. Mamba is used as the base feature extraction module of classifier, and TCN is introduced to enhance the fine-grained feature mining capability, which effectively extracts the correlation information among different traces while maintaining the model recognition rate. Finally, a split-bucket strategy based on normal distribution is designed to dynamically adjust the length threshold division to reduce the coding sparsity problem caused by longer traces in fixed-length coding. Comparative experiments are conducted on multiple machine learning methods and vanilla fitness-obtaining methods, and the proposed method achieves better results with all metrics on the BPIC2019. Also, multiple ablation experiments are conducted to discuss the effectiveness of the model structure. Comprehensive experiments demonstrated the effectiveness of the proposed method.

Keywords: approximate conformance checking, split-bucket strategy, Mamba, TCN, business process

1. INTRODUCTION

Online and automation of business processes has become an important way to improve the operational efficiency of enterprises and reduce costs [1]. However, as the complexity of business processes increases and inter-system interactions become more frequent, the conformance problem of business processes is becoming more and more prominent, which is a key factor restricting the efficient operation of enterprises [2]. Process mining techniques can extract actionable process knowledge and insights from historical execution data. Process discovery algorithms can transform captured event data into process models, while conformance checking algorithms can check whether process executions recorded in event data conform to a reference model. The main purpose of conformance checking is to verify that the process executions recorded in the logs is consistent in the form of a process model.

Existing conformance checking methods are mainly classified into rule-based methods, token-replay methods and alignment-based methods [3]. Rule-based methods capture the behavior of the model through binary relationships of events represented by model tasks. It derives rules from the model and checks the compliance of the traces according to the rules sequentially and measures the conformance with the value of fitness. The token-replay methods perform replay analysis by executing the events of the process model step-by-step according to the order of the individual events, checking at each replay step whether the current state of the model satisfies the conditions for executing the corresponding event. But they often lead to ambiguous or unpredictable results. To address these issues, alignment-based methods have quickly become the de facto standard conformance checking technique. The alignment-based methods connect each event of a trace to a sequence of model executions as a way of checking whether the trace conforms to the process rules. However, as the information system grows larger and more complex, computing alignments becomes time-consuming and memory-intensive when applied to real, large-scale event data, making it difficult to use in reality. In many large organizations, it is not necessary to compute the exact alignment between each log behavior and the process model. Instead, an approximate conformance checking is usually sufficient for adequately analyzing business processes and systems.

Artificial intelligence methods have been applied in various scenarios [4]. The development of machine learning provides a promising solution to approximate conformance checking. But they still have the following problems:

(1) Considering only the sequence relationships within traces and not the relationships between different traces.

(2) Neglecting the difference between traces of different length intervals, where shorter traces usually correspond to the beginning of a process and longer traces usually correspond to the end of a process, and their possible execution events are different.

To address these issues, we introduce machine learning methods in this task and propose MACC. MACC constructs feature embedding of traces and the feature embedding can be easily transplanted to any machine learning based regressors to improve approximate conformance checking performance. In order to efficiently mine the correlation information between different traces, a Mamba based feature extraction module is proposed in MACC. Mamba can maintain linear complexity and explore deep correlations between different traces. We further introduce TCN to enhance the ability of fine-grained feature mining. Finally, we designed a split-bucket strategy based on normal distribution, dynamically adjusting the length threshold partitioning to reduce the encoding sparsity problem caused by longer traces in fixed length encoding.

2. METHOD

The training progress of MACC can be divided into two stages as illustrated in Figure 1. In the first stage, we first modify the log to mutate it, while obtaining both correct and incorrect traces. Then we use the alignment in pm4py to compute the fitness of each trace. If the fitness a trace is less than 1, its classification label is marked as 0; otherwise, its classification label is marked as 1. Then, the logs are length-binned according to the normal distribution binning strategy. We use these traces and their classification label to train different Mamba classification models for three different buckets. In the second stage, we input the one-hot encoded traces into the classifier trained in the first stage. And the feature embeddings obtained by the classifier are added to the embedding of the original trace, the regressor is trained along with the original fitness.

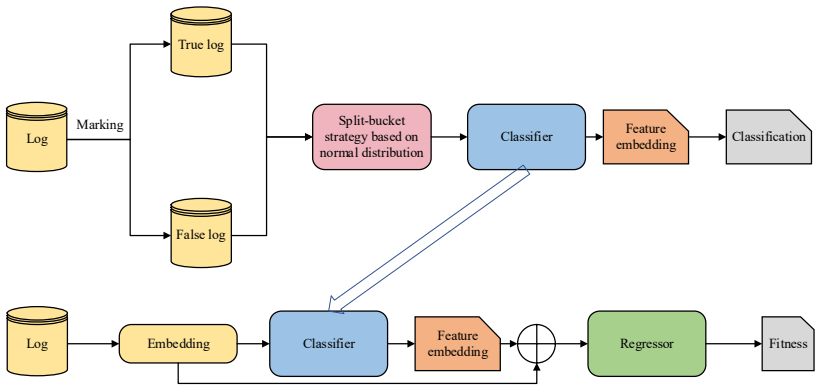

Figure 1. Workflow of the MACC.

MACC can achieve approximate conformance checking through logs alone, without providing a specific process structure. The larger the data volume of the provided logs, the better the performance of MACC, which meets the application requirements of large-scale system logs. The designed split-bucket strategy based on normal distribution can be dynamically divided by a length threshold, and traces of different lengths are trained with different Mamba models, which reduces the coding sparsity problem caused by longer traces in fixed-length coding. The designed CCM contains an improved Mamba feature extraction block [5] and introduces TCN [6] to enhance the fine-grained feature mining capability, which effectively extracts the correlation information between different traces while maintaining the model recognition rate.

2.1 Split-bucket strategy based on normal distribution

We devise a split-bucket strategy based on normal distribution (SBS). The strategy first counts the lengths of all traces in the input log and accumulates the number of traces of the same length. Then, a normal fit is performed according to the distribution of the number of traces of different lengths to determine the mean μ and standard deviation σ of the overall distribution. Subsequently, $\mu+\sigma$ and $\mu-\sigma$ are selected as the length division thresholds. Then after that, the input logs

are divided into three buckets according to the thresholds, and each bucket contains traces of different length intervals. Finally, the traces in each bucket are coded uniquely and hotly according to the maximum length respectively, and the insufficient lengths are filled with zeros.

The above bucket strategy ensures that the lengths are close to the traces to be encoded, and avoids the sparse encoding problem caused by a large number of filled zeros by encoding shorter traces according to the length of longer traces. At the same time, as the business process proceeds, the sequence relationships between traces of similar lengths become closer, and the bucket strategy can also improve the ability of model to detect the conformance of traces of different lengths.

2.2 Mamba based classification model

The structure of the Mamba based classification model is shown in Figure 2. The one-hot encoded traces are first layer normalized and copied in two copies. One copy goes through the linear layer and then row attention (RA) computation is performed. Then after that, feature extraction is performed after TCN and nonlinear activation is performed using SiLU function. Subsequently, feature extraction is performed using SS2D and layer normalization is performed. Another copy is matrix multiplied with the result of the first copy after passing through a linear layer and SiLU function in turn. Finally, the final output is obtained by summing with the original input after passing through a linear layer. This progress can be formulated as (1)-(3).

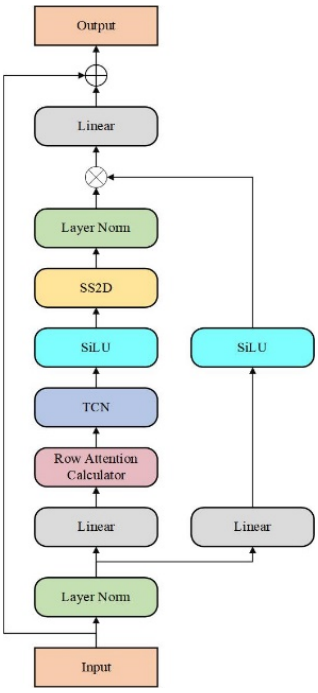

Figure 2. The structures of the proposed Mamba based classification model.

$$O_1 = LN\left(SS2D\left(SiLU\left(TCN\left(RA\left(N\left(LN\left(I\right)\right)\right)\right)\right)\right)\right) \tag{1}$$

$$O_2 = SiLU\left(N\left(LN\left(I\right)\right)\right) \tag{2}$$

$$O = I + N\left(O_1 \otimes O_2\right) \tag{3}$$

Where, I is the input, O is the output, O_1 and O_2 is the intermediate result. LN represents layer normalization, N represents linear layer, RA represents row attention calculation, $\otimes$ represents matrix multiplication.

The purpose of designing line attention computation is to extract the correlation information of different sequence sees while reducing the computational effort. TCN is introduced with the aim of extracting the temporal features encoded in each trace using causal convolution and dilated convolution therein at different steps and scales. SS2D is introduced with the aim of achieving global sense field at the cost of linear complexity.

3. EXPERIMENTS

3.1 Experimental details

The hardware environment for our experiments has a GPU of RTX 3060, a processor of Intel i5-13600KF, and a software environment of Pytorch. The optimizer chosen for the training process is Adaptive Gradient (AdaGrad), the base learning rate is set to 0.00001, and the learning rate update strategy is CosineAnnealingLR, and the maximum number of training rounds is 200.The loss function is Cross Entropy Loss, which is the most commonly used loss function in classification problems. The ratio of training logs and test logs is divided into 7:3. The experiments are conducted using four evaluation metrics for model assessment, namely Accuracy (ACC), Mean Absolute Error (MAE), Mean Square Error (MSE) and R^2 .

3.2 Datasets

The experiments were conducted on Event Graph of BPI Challenge 2019 (BPIC2019). The dataset is available on the following data open access website: https://data.4tu.nl. BPIC2019 is derived from the purchase order handling process in 60 subsidiaries of a large Dutch multinational company operating in the field of paints and coatings. Information from both datasets is shown in Table 1. To our knowledge, there are no traces in the logs that do not conform to the real process model. Therefore, we first obtain the model of the real logs. Perturbations are added into these traces to generate traces that do not conform to the process model, and use the alignment technique in pm4py to calculate the fitness of each trace with respect to the model, which will be used as the dataset for our experiments in this paper. Notably, the classification model is also trained with these data as illustrated in Section 2.

Table 1. Information of the two datasets.

Name of datasets	Number of events	Maximum trace length	Minimum trace length	Number of legal traces	Number of illegal traces
BPIC2019	16	14	5	21856	10873

3.3 Comparative experiments

MACC is conducted comparative experiments with nine baselines on BPIC2019 to validate its effectiveness, which includes raw regressor, four machine learning methods, three deep learning classifiers and alignment-based method. Two regressors, Random Forest (RF) and Multilayer Perceptron (MLP), are chosen to verify the generalization of the classifiers on different regressors. The results of the comparative experiments are shown in Table 2.

Table 2. Results of comparative experiments.

Regressor	Classifier	ACC (%)↑	MAE (10^{-4})↓	MSE (10^{-5})↓	R^2(10^{-2})↓
RF	No classification embedding	97.32	54.12	24.36	94.42
	KNeighbors [7]	97.66	53.94	24.01	94.01
	DecisionTree [8]	98.02	52.86	22.77	92.68
	AdaBoost [9]	97.89	53.45	23.68	93.75
	GaussianNB [10]	98.16	52.69	22.34	92.13
	MLP [11]	98.49	52.23	21.97	91.88
	LSTM [12]	98.77	51.98	21.53	91.67
	GRU [13]	99.01	51.65	20.99	89.26
	MACC	99.21	50.23	20.64	88.15
MLP	No classification embedding	97.42	47.68	20.45	85.78
	KNeighbors [7]	97.84	47.01	19.67	85.28
	DecisionTree [8]	98.25	45.81	18.04	84.67
	AdaBoost [9]	98.03	46.23	18.58	84.99
	GaussianNB [10]	98.35	45.35	17.98	84.52
	MLP [11]	98.61	44.28	17.52	83.27
	LSTM [12]	98.88	43.78	16.88	82.45
	GRU [13]	99.12	42.82	16.23	82.18
	MACC	99.96	41.23	15.24	81.23

MACC achieved the best values on all four evaluation metrics on the dataset. With the RF regressor, MACC has improved by 0.20% over the second best classifier on ACC and decreased more on the remaining metrics. With the MLP regressor, the MACC has improved by 0.84% over the second best classifier on the ACC and decreased more on the rest of the metrics. This indicates that the effectiveness of MACC for large-scale structure-unknown process approximate conformance checking scenarios. Meanwhile, the ACC of MACC improves by 1.89% on the RF regressor and 2.54% on the MLP regressor compared to not adding classification vectors. At the same time, the results of the deep learning classifiers are better than the machine learning classifiers in all evaluation metrics, which indicates that the deep learning classifiers are more effective in the field of large-scale log approximate conformance checking.

We also focus on the global fitness. Here, we use the same setup as [14]. And as illustrated in Fig. 3, four models in [15] are adopted in the experiment, including single most frequent trance (Model 1), all trace in parallel (Model 2), all transitions in parallel (Model 3), and flower model (Model 4). These values are compared with different methods from literature: alignment-based recall [16], behavioral recall [17] and Markovian recall with k=3 [18] and RNN-based recall [14]. The results can be found in Table 3.

Table 3. Comparative results on different models.

Models	Alignment-based recall	Behavioral recall	Markovian recall	RNN-based recall	MACC recall
Model 1	0.85	0.69 (-0.16)	0.72 (-0.13)	0.89 (+0.04)	0.83 (-0.02)
Model 2	1.00	0.90 (-0.10)	0.91 (-0.09)	0.96 (-0.04)	0.97 (-0.03)
Model 3	1.00	1.00	1.00	1.00	1.00
Model 4	0.78	0.84 (+0.06)	0.71 (-0.07)	0.81 (+0.03)	0.81 (+0.03)
MAE	0	0.08	0.07	0.04	0.03

From Table 3, all methods can achieve satisfactory accuracy on short trace model. Especially on Model 3, all methods can match the alignment-based method. As the length of traces increases, the performance of all methods decreases. But the proposed MACC achieves the best accuracy across all method. Compared with the state-of-the-art RNN-based fitness, MACC reduces the MAE by 0.01.

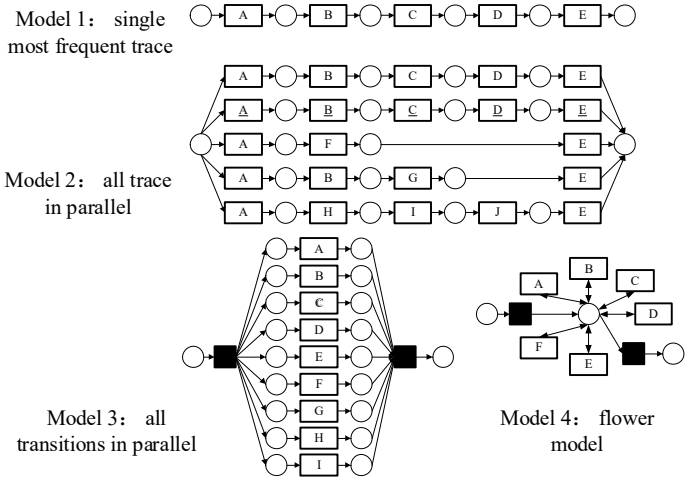

Figure 3. Models for the experimental setup.

3.4 Ablation studies

In order to verify the validity of the MACC structure, we conducted a total of five ablation experiments on the BPIC2019 dataset, which has a large amount of data. The effect of each structure on MACC is verified by modifying the number of SBS, TCN, SS2D, and RA structures. The number before each structure indicates the number of structures. The results of the ablation experiments are shown in Table 3.

Table 3. Results of ablation experiments.

Name of datasets	Methods	ACC (%)↑	MAE (10^{-4})↓	MSE (10^{-5})↓	R^2 (10^{-2})↓
BPIC2019	0×SBS	97.78	44.46	16.05	83.17
	2×SBS	98.72	42.31	15.42	81.98
	0×TCN	98.67	42.63	15.63	82.06
	0×SS2D	97.52	44.56	16.11	83.64
	0×RA	98.34	43.21	15.86	82.75
	MACC	99.96	41.23	15.24	81.23

The existing structure of the MACC has achieved the best values on all four metrics on the BPIC2019 dataset. Any modification to the existing structure has resulted in a decrease in model performance. The model performance degradation caused by removing SS2D is the most severe, with a 2.44% decrease in the ACC metric. The model performance degradation caused by removing the SBS is the next worst, with a 2.18% decrease in the ACC metric. Adding an SBS causes the smallest model performance degradation, with a 1.24% decrease in the ACC metric. It can be seen that the designed SBS and RA, the introduced SS2D and TCN all affect the performance of MACC to a greater extent. In summary, the existing MACC structure is the most reasonable.

4. CONCLUTIONS

We propose a novel Mamba based method, MACC, for all regression based business process approximate conformance checking methods. By adding extra feature embeddings from MACC, approximate conformance checking regressors can improve their performance compared to vanilla machine learning based methods. But the model is relatively large. The direction of future research will focus on model compression and acceleration.

REFERENCES

[1] Gažová, A., Papulová, Z., & Smolka, D. "Effect of business process management on level of automation and technologies connected to industry 4.0." Procedia Computer Science 200 : 1498-1507 (2022).

[2] Riaz, S., Morgan, D., & Kimberley, N. "Using complex adaptive systems (CAS) framework to assess success factors that lead to successful organizational change: a new way to understand change implementation for success." Journal of Organizational Change Management (2024).

[3] Torres, W., Van den Brand, M. G., & Serebrenik, A. "A systematic literature review of cross-domain model approximate conformance checking by model management tools." Software and Systems Modeling 20: 897-916 (2021).

[4] Zhao H, Liu Q, Sun H, et al. "Community Awareness Personalized Federated Learning for Defect Detection". IEEE Transactions on Computational Social Systems (2024).

[5] Gu, A., Goel, K., & Ré, C. "Efficiently modeling long sequences with structured state spaces." arxiv preprint arxiv:2111.00396 (2021).

[6] Bai, S., Kolter, J. Z., & Koltun, V. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." arxiv preprint arxiv:1803.01271 (2018).

[7] Blough, D. M., Leoncini, M., Resta, G., & Santi, P. "The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks." IEEE Transactions on Mobile Computing 5.9 : 1267-1282 (2006).

[8] Azam, Z., Islam, M. M., & Huda, M. N. "Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree." IEEE Access (2023).

[9] Ramakrishna, M. T., Venkatesan, V. K., Izonin, I., Havryliuk, M., & Bhat, C. R. "Homogeneous adaboost ensemble machine learning algorithms with reduced entropy on balanced data." Entropy 25.2 (2023): 245.

[10] Gill, K. S., Anand, V., Gupta, R., & Hsiung, P. A. "Detection of Malware Using Machine Learning techniques on Random Forest, Decision Tree, KNeighbour, AdaBoost, SGD, ExtraTrees and GaussianNB Classifier." 2023 4th IEEE Global Conference for Advancement in Technology (GCAT) (2023).
[11] Peng, S., Yan, Y., Shuai, Q., Bao, H., & Zhou, X. "Representing volumetric videos as dynamic mlp maps." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023).
[12] Yu, Y., Si, X., Hu, C., & Zhang, J. "A review of recurrent neural networks: LSTM cells and network architectures." Neural computation 31.7 : 1235-1270 (2019).
[13] Cahuantzi, R., Chen, X., & Güttel, S. "A comparison of LSTM and GRU networks for learning symbolic sequences." Science and Information Conference. Cham: Springer Nature Switzerland (2023).
[14] Peeperkorn, Jari , S. V. Broucke , and J. D. Weerdt . "Supervised Conformance Checking Using Recurrent Neural Network Classifiers." (2021).
[15] van Dongen B F, Carmona J, Chatain T. "A unified approach for measuring precision and generalization based on anti-alignments" International conference on business process management. Cham: Springer International Publishing: 39-56 (2016).
[16] Aalst, Wil Van Der , A. Adriansyah , and B. V. Dongen . "Replaying history on process models for conformance checking and performance analysis." Wiley Interdisciplinary Reviews Data Mining & Knowledge Discovery 2.2(2012):182-192.
[17] Goedertier, Stijn , et al. "Robust Process Discovery with Artificial Negative Events." Journal of Machine Learning Research 10.urn:hdl:123456789/217117(2009).
[18] Augusto, Adriano , et al. "Measuring Fitness and Precision of Automatically Discovered Process Models: A Principled and Scalable Approach." IEEE Transactions on Knowledge and Data Engineering PP.99(2020):1-1.