PaPPI: Privacy-aware Process Performance Indicators (Extended Abstract)

Martin Kabierski*, Stephan A. Fahrenkrog-Petersen, Glenn Dittmann and Matthias Weidlich

Humboldt-Universität zu Berlin, Berlin, Germany

Abstract

The evaluation of recorded process executions using process performance indicators, short PPIs, serves as a main driver of process optimization and process monitoring. Yet, since many processes inadvertently record information about individuals involved in said processes, the analysis of such data is bound by data protection regulations, such as the GDPR and the CCPA. To enable the analysis of the respective data, while conforming to privacy regulations, anonymization techniques can be employed. In this work, we propose PaPPI, Privacy-aware Process Performance Indicators, a Java-based library for the definition and evaluation of process performance indicators under differential privacy. Our toolkit builds upon and extends the PPINOT library for process performance indicators, maintaining the well-established syntax and semantics of PPINOT. This way, we achieve an easy-to-use integration of privacy protection in the computation of process performance indicators.

Keywords

process mining, performance indicators, privacy-awareness, differential privacy

1. Introduction

The evaluation of recorded process executions is a main driver for the analysis of process-centric information systems. Following the common BPM life cycle, such evaluations are the backbone of any process improvement initiative and guide the re-design of processes. The analysis of recorded process executions may be based on techniques for conformance checking [1], compliance verification [2], or the evaluation of quantifiable metrics of a processes efficiency and effectiveness, which are commonly referred to as process performance indicators (PPIs) [3]. These metrics are defined by the process owner in order to communicate and monitor certain high-level goals. Their evaluation over the recorded process executions, which is typically available in the form of event logs, enables conclusions on the extent to which these goals are met.

The PPINOT metamodel [3] has been proposed as a general framework for the definition and evaluation of PPIs. According to the PPINOT model, PPIs are composed from atomic building blocks, so-called measure definitions. These measures are aggregated in a tree-like

ICPM 2022 Doctoral Consortium and Tool Demonstration Track

*Corresponding author.

martin.kabierski@hu-berlin.de (M. Kabierski); stephan.fahrenkrog-petersen@hu-berlin.de (S. A. Fahrenkrog-Petersen); glenn.dittmann@hu-berlin.de (G. Dittmann); matthias.weidlich@hu-berlin.de (M. Weidlich)

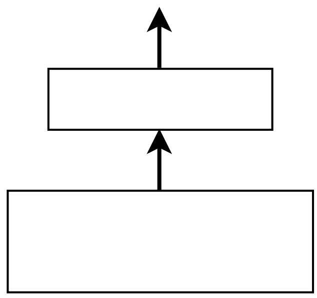

Figure 1: A PPI defined using the PPINOT metamodel.

manner to evaluate complex functions defined over the process instances and their attributes. In Figure 1, we illustrate the PPINOT model with the PPI *"Average Duration"* that is included in the public PPINOT example repository.[1] It is based on a measure that captures the time between the occurrences of two types of events with the data recorded for *one* process execution. These values are then aggregated over *all* process executions by computing the arithmetic mean.

The recorded process executions over which PPIs are defined and evaluated often include sensitive information about individuals involved in the process, such as knowledge workers in traditional business processes or patients in clinical pathways. Any handling and analysis of this data has to adhere to data protection regulations, such as the GDPR or the CCPA [4]. To this end, anonymization techniques can be employed to protect the privacy of individuals, while still supporting the evaluation of the respective data.

In recent work, we proposed a framework for privacy protection during the evaluation of PPIs defined using the PPINOT metamodel [5]. Specifically, our framework provides a privacy guarantee in terms of the well-established notion of differential privacy [6]. For this purpose, we proposed multiple privacy-preserving release mechanisms, i.e. functions, that add controlled noise to the true result of a function. In this demo, we present *PaPPI*, Privacy-aware Process Performance Indicators, a Java-based library that implements the aforementioned framework. *PaPPI* has been designed such that it wraps the the publicly available PPINOT library, so that users can rely on the established syntax and semantics for the definition of PPIs, while still benefiting from the privacy protection offered by our techniques. In particular, *PaPPI* enables the privacy-aware evaluation for any PPI, that can be defined using the PPINOT syntax. As such, *PaPPI* provides an easy-to-use way to include privacy considerations in the quantitative analysis of process executions. While *PaPPI* has not been used in practice yet, we validated its applicability for real-life scenarios in a case study on a publicly available log file [5].

We first illustrate the definition of PPIs in our toolkit (section 2), before turning to their evaluation (section 3). We then elaborate on the availability of our library (section 4), before we conclude (section 5).

[1] https://github.com/isa-group/ppinot-example

```
1  //Load Log
2  LogProvider log = new MXMLLog(new FileInputStream(new File(”simulation_logs.mxml”)),null);
3
4  //PPI Definition
5  TimeMeasure duration = new TimeMeasure();
6  duration.setFrom(new TimeInstantCondition(”EVENT 2 START MESSAGE”, GenericState.START));
7  duration.setTo(new TimeInstantCondition(”FI closed”, GenericState.END));
8  duration.setUnitOfMeasure(TimeUnit.HOURS);
9
10 PrivacyAwareAggregatedMeasure privatizedAvg = new
11 PrivacyAwareAggregatedMeasure();
12 privatizedAvg.setBaseMeasure(duration);
13 privatizedAvg.setAggregationFunction(PrivacyAwareAggregator.AVG_LAP);
14 privatizedAvg.setEpsilon(0.1);
15 privatizedAvg.setBoundaryEstimation(BoundaryEstimator.MINMAX);
16 privatizedAvg.setId(”AvgDuration”);
17
18 //PPI Evaluation
19 MeasureEvaluator evaluator = new PrivacyAwareLogMeasureEvaluator(log);
20 evaluator.eval(privatizedAvg, new SimpleTimeFilter(Period.MONTHLY,1, false));
```

Algorithm 1: Definition and evaluation of a PPI in PaPPI.

2. Defining PPIs

The definition of PPIs in PaPPI closely follows the syntax of PPINOT, i.e. the different types of measure definitions are composed in a tree-like structure to form more complex evaluation functions. In particular, the set of available measure types consists of base measures, which are evaluated over single process instances, aggregation measures, that aggregate information retrieved over multiple process instances using predefined functions (*Avg*, *Sum*, *Min* or *Max*), and derived measures, which are user-defined functions, to be evaluated over single or multiple process instances. The MeasureDefinition classes of PPINOT are extended to include, for each multi-instance measure in the defined tree, additional information about whether and how it should be privatized during evaluation. In particular, if it shall be privatized, the value of the privacy parameter ϵ, the chosen differentially private release mechanism, and a method for estimating the bounds of the input data need to be defined. In Algorithm 1, starting at line 5, the definition of the PPI of Figure 1 using PaPPI is shown. Here, we specify that the evaluation of the aggregation measure shall be privatized using $\epsilon = 0.1$ with a boundary estimation based on the minimum and maximum value of the inputs, and the *Laplace mechanism* to calculate the *Average* of the inputs.

3. Evaluating PPIs

For the subsequent evaluation of PPIs, we extended the LogMeasureEvaluator class of PPINOT, to enable the invocation of the specified privatized measures during evaluation. Using this new evaluator, the evaluation of a given PPI definition, or a set thereof, can be conducted as shown in lines 19 and 20 of Algorithm 1. Here, we specify that the PPI privatizedAvg shall be evaluated in monthly time segments for the respective log.

PPI	from	to	value
AvgDuration	2015-05-01T00:00:00.000Z	2015-05-31T23:59:59.999Z	71.8089720305794
AvgDuration	2015-06-01T00:00:00.000Z	2015-06-30T23:59:59.999Z	192.359426832879
AvgDuration	2015-07-01T00:00:00.000Z	2015-07-31T23:59:59.999Z	281.928831451766
AvgDuration	2015-08-01T00:00:00.000Z	2015-08-31T23:59:59.999Z	317.510586006456
AvgDuration	2015-09-01T00:00:00.000Z	2015-09-30T23:59:59.999Z	396.21731364286
AvgDuration	2015-10-01T00:00:00.000Z	2015-10-31T23:59:59.999Z	439.810087570682
AvgDuration	2015-11-01T00:00:00.000Z	2015-11-30T23:59:59.999Z	467.314332985255
AvgDuration	2015-12-01T00:00:00.000Z	2015-12-31T23:59:59.999Z	680.695924249304
AvgDuration	2016-01-01T00:00:00.000Z	2016-01-31T23:59:59.999Z	667.888854712306

Figure 2: Output generated by 1.

For a given PPI, the evaluator first determines, whether the provided PPI definition is admissible for privatization, i.e. if the evaluation of said PPI with the specified measures to privatize, would properly protect each of the logs accessed information. Should this not be the case, the evaluation stops, informing the user about the problematic PPI. A given PPI definition is considered non-admissible if either not all information retrieved from process instances would be privatized or if retrieved information would be privatized more than once during evaluation. The privatized results can either be printed or saved in a .csv-file, such as the one shown in Figure 2, that has been generated by 1. The file contains for each PPI and time segment the value obtained by the privatized evaluation.

Concerning the release mechanisms, we currently provide implementations of the mechanisms proposed in [5], i.e., the *Laplace mechanism* and the *Interval Mechanism* for aggregation measures, as well as the *Sample-and-Aggregate mechanism* for multi-instance derived measures. Due to the implementation utilizing a factory pattern for invoking the evaluation of the measure definitions of the PPI based on the aforementioned specifications, it is easy to extend the implementation with additional release mechanisms, by providing the factory with a mapping from a chosen identifier of the mechanism in the PPI definition to its implementation.

4. Availability

The library is publicly available on GitHub[2] under the MIT license. There, we also provide further guidance for installing the library and adding new release mechanisms. Furthermore, we provide a screencast of the definition and evaluation of PPIs using the library.[3]

We plan to extend the library with additional features, such as an automated selection of release mechanisms and the support of data-driven privacy-aware PPI definitions from structured file formats.

[2]https://github.com/MartinKabierski/privacy-aware-ppinot
[3]https://youtu.be/i_WnR-ReVnE

5. Conclusion

In this demo, we proposed *PaPPI*, a Java-based library, that serves as a wrapper around the PPINOT library, adding privacy protection in the form of differential privacy to the definition and evaluation of PPIs. We re-used and extended the concepts of the PPINOT meta model, so that privacy-protection can be easily integrated for users familiar with its definition and evaluation syntax.

Acknowledgments

This work has received funding from the Deutsche Forschungsgemeinschaft (DFG), grant number 421921612.

References

[1] J. Carmona, B. van Dongen, A. Solti, M. Weidlich, Conformance checking, Switzerland: Springer.[Google Scholar] (2018).

[2] M. e. Kharbili, A. K. A. d. Medeiros, S. Stein, W. M. van der Aalst, Business process compliance checking: Current state and future challenges, Modellierung betrieblicher Informationssysteme (MobIS 2008) (2008).

[3] A. del Río-Ortega, M. Resinas, C. Cabanillas, A. Ruiz-Cortés, On the definition and design-time analysis of process performance indicators, Information Systems 38 (2013) 470–490.

[4] G. Elkoumy, S. A. Fahrenkrog-Petersen, M. F. Sani, A. Koschmider, F. Mannhardt, S. N. Von Voigt, M. Rafiei, L. V. Waldthausen, Privacy and confidentiality in process mining: threats and research challenges, ACM Transactions on Management Information System (TMIS) 13 (2021) 1–17.

[5] M. Kabierski, S. A. Fahrenkrog-Petersen, M. Weidlich, Privacy-aware process performance indicators: Framework and release mechanisms, in: International Conference on Advanced Information Systems Engineering, Springer, 2021, pp. 19–36.

[6] C. Dwork, A. Roth, et al., The algorithmic foundations of differential privacy, Foundations and Trends® in Theoretical Computer Science 9 (2014) 211–407.