BenchIMP: A Benchmark for Quantitative Evaluation of the Incident Management Process Assessment

Alessandro Palma
Sapienza University of Rome
Rome, Italy
palma@diag.uniroma1.it

Nicola Bartoloni
Sapienza University of Rome
Rome, Italy
bartoloni.1909908@studenti.uniroma1.it

Marco Angelini
Link Campus University of Rome
Rome, Italy
m.angelini@unilink.it

ABSTRACT

In the current scenario, where cyber-incidents occur daily, an effective Incident Management Process (IMP) and its assessment have assumed paramount significance. While assessment models, which evaluate the risks of incidents, exist to aid security experts during such a process, most of them provide only qualitative evaluations and are typically validated in individual case studies, predominantly utilizing non-public data. This hinders their comparative quantitative analysis, incapacitating the evaluation of new proposed solutions and the applicability of the existing ones due to the lack of baselines. To address this challenge, we contribute a benchmarking approach and system, BenchIMP, to support the quantitative evaluation of IMP assessment models based on performance and robustness in the same settings, thus enabling meaningful comparisons. The resulting benchmark is the first one tailored for evaluating process-based security assessment models and we demonstrate its capabilities through two case studies using real IMP data and state-of-the-art assessment models. We publicly release the benchmark to help the cybersecurity community ease quantitative and more accurate evaluations of IMP assessment models.

CCS CONCEPTS

• **Information systems → Process control systems**; **Decision support systems**; • **Security and privacy**; • **Computing methodologies → Model development and analysis**;

KEYWORDS

Cybersecurity Benchmark, Incident Management, Cybersecurity Processes, Quantitative Assessment

ACM Reference Format:
Alessandro Palma, Nicola Bartoloni, and Marco Angelini. 2024. BenchIMP: A Benchmark for Quantitative Evaluation of the Incident Management Process Assessment. In *The 19th International Conference on Availability, Reliability and Security (ARES 2024), July 30–August 02, 2024, Vienna, Austria.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3664476.3664504

1 INTRODUCTION

In an increasingly interconnected and digital world, the importance of an effective Incident Management Process (IMP) cannot be overstated. Organizations are constantly exposed to a wide range of potential incidents that can disrupt their operations, compromise data security, and damage their reputation. These incidents encompass a broad spectrum, from cyberattacks and data breaches to natural disasters, accidents, and service interruptions. Setting up the IMP is a critical procedure for any organization's security and operational strategy. It encompasses a structured approach to identifying, managing, and resolving unexpected events (i.e., security incidents) that can impact data and systems' confidentiality, integrity, and availability. If not properly managed, the IMP may require significant time and resources to coordinate internally with the team and analyze resources [19, 35]. For this reason, the analytical evaluation of the IMP, namely *IMP assessment*, has become extremely important to identify and quantify the incident causes and provide mitigation actions to improve the process.

Problem statement and contribution. Given the difficulty of quantitatively assessing the IMP due to general and manual guidelines provided by security standards [10, 47], different works proposed new assessment models to support an objective IMP assessment [8, 25, 46]. On one side, the security standards do not provide a systematic evaluation of the IMP assessment models. On the other hand, given the lack of open-source data and methodologies to test these approaches [17], these works suffer the limitation that they have been tested and applied in extremely specific scenarios (e.g., small healthcare anonymized networks [8]). This impedes the ability to conduct comparative quantitative evaluations of these approaches, making their applicability challenging due to the absence of objective criteria to choose one over the others.

While in other cybersecurity fields benchmarking approaches caught on to address comparative analyses [5], to the best of the authors' knowledge, a similar solution is missing for the assessment of the IMP. To address this gap, this paper moves the first step toward the quantitative evaluation of the IMP assessment. We contribute BenchIMP, a benchmark system that supports security experts in the comparative analysis of models for quantitative evaluation of the IMP assessment, leveraging the notion of incident cost. It validates the performance of assessment models and analyzes their robustness considering different error scenarios and multiple analytical perspectives (e.g., different sources of human mistakes and error magnitudes). BenchIMP supports multi-metric analysis for a comprehensive quantitative objective evaluation in a fully automated system, not achievable with existing manual approaches. We present two case studies using a real IMP log from an IT company to show the capabilities of the proposed benchmark to evaluate a new assessment model and enable a comparative analysis with the

ARES 2024, July 30–August 02, 2024, Vienna, Austria

ACM ISBN 979-8-4007-1718-5/24/07
https://doi.org/10.1145/3664476.3664504

current state-of-the-art solutions for IMP assessment. In summary, the paper contributes:

- The definition of benchmarking scope and requirements for quantitative IMP assessment;
- The benchmark design and architecture for the evaluation of IMP assessment models;
- The development and public release of BenchIMP, the benchmarking system for the IMP assessment[1];
- Two case studies showing the benefits for security assessors, practitioners, and researchers in leveraging the enabled quantitative assessment.

2 BACKGROUND

This section provides the fundamental definitions of the IMP and how its assessment is performed. Then, we introduce the notion of incident cost and assessment model.

Incident Management Process Assessment. Different standards exist describing the IMP (e.g., ISO 27035 [27], ITIL [29], ENISA [22]), which define five crucial phases for an effective process: *planning and preparation*, where the organization establishes policies, competent teams, and logging procedures; *detection and reporting*, involving the identification and logging of potential incidents, such as suspicious activities; *incident analysis* to categorize incidents based on their impact and urgency; *incident response* providing necessary mitigation actions, and finally *incident closure*.

Nowadays, various IT service management systems [45, 48] assist organizations in executing the IMP. Thus, users open tickets to report detected problems that may lead to incidents, and then the ticketing system tracks the entire IMP lifecycle. The logged data in this process is referred to as the *incident management process log* (or simply IMP log), and it can be retrieved from these ticketing systems or manually managed by security operators.

The IMP log is the necessary input to perform the IMP assessment as it represents the actual execution of such a process [55]. In this scenario, *Incident Management Process Assessment* is the task of analyzing and evaluating the implementation of the IMP of an organization. When performing this task, the assessor typically examines the IMP log and uses checklists to evaluate various security requirements, such as process definition, process accountability, and consistency between organizational plans [28]. Finally, she estimates each requirement's compliance level based on predefined qualitative scales (e.g., compliant, partially compliant, not compliant) and aggregates the results (e.g., average) to assess the overall IMP. Let us note that this kind of assessment is qualitative and, as such, does not express quantitative aspects such as, for example, the cost arising from the bad execution of the IMP.

Incident costs and assessment models. Given the importance of quantitatively indicating the incident impacts, one key element concerning the assessment of the IMP revolves around incident costs. It is not limited to just the financial outlays but encompasses the effects of not adhering to security standards or the tangible efforts to respond to incidents. The precise interpretation of incident costs varies depending on the particular purpose of the assessment. However, the crucial point is that it serves as a measurable quantitative incident metric. An assessor is interested in quantifying such costs to measure the incident impacts and design security strategies accordingly. Thus, in the rest of the paper, we use the encompassing term *incident cost* as a multi-dimensional quantifier of any cost/impact related to incidents.

While the importance of incident costs to organizational assets is recognized [35], there is no standard way to assess the cost of the incidents because of its variability in different contexts. Thus, a recent trend is the definition of models to evaluate the incident costs using the IMP log [8, 15, 26, 42, 53], that we refer as *IMP assessment models*. Their goal is to support the assessor in computing the incident costs from the log features through data-driven solutions to reduce the subjectivity of a purely human evaluation.

3 RELATED WORK

Due to the critical nature of security processes, different solutions for supporting their execution have gained greater prominence in recent years. Among them, Varela-Vaca et al. [52] and Bernardi et al. [11] leverage process mining to provide assessment methodologies to calibrate and validate the performance scenario of (security) processes. Similarly, Ganin et al. [23] bridge the gap between risk assessment and management by combining threat, vulnerability, and impact metrics to provide a comprehensive assessment, while Battaglioni et al. [10] propose a probabilistic model to compute the likelihood of occurrence of a cyber incident. Finally, Angelini et al. [8] propose a framework to review and assess the cyber risk assessment by leveraging security ontologies and attack graphs.

All these works underline the importance of supporting the assessment of security processes. However, they are qualitative, validated on specific use cases with no publicly available data (e.g., financial and hospital infrastructure), or require a great component of human evaluation of the asset values prior to the assessment, hindering objective comparison.

Benchmarking approaches in cybersecurity. A typical solution in the literature consists of designing benchmarks to support the quantitative evaluation of cybersecurity approaches, as they enable objective comparability and validation in different contexts. For example, Dumitras and Shou [20] propose a benchmark for malware detection approaches, leveraging comprehensive data sources, including binary and URL reputation, telemetry, email spam, and malware samples. Similarly, Wang et al. [54] develop a quantitative assessment of the risks of adopting randomization techniques for hardening outdated binaries during the binary-rewriting process.

In the field of web security, Tukaram et al. [9] define an inventory of rules for authentication, encryption, availability, and a set of recommendations for microservices. Similarly, Oliveira et al. [37] introduce a benchmark for assessing and comparing the security of web service frameworks to support their comparison according to multiple attributes such as memory usage and throughput.

Different benchmarks are also proposed for IoT security, as Almakhdhub et al. [5] who introduce a benchmark suite and evaluation framework for IoT security in terms of performance, memory usage, and energy consumption. Likewise, De Ruck et al. [18]

[1]The benchmark code and results are available at https://github.com/Ale96Pa/BenchIMP

present a platform that generates customized Linux-based firmware benchmarks representative of the manufacturers' devices.

Finally, other benchmarks are developed for cyber-physical systems, such as Luna et al. [34], who propose a benchmark for Cloud Security Level Agreements with respect to user-defined requirements, and Amin et al.[7] who introduce a game-theoretic framework to benchmark the risks that arise from reliability failures.

As shown, benchmarking is used in different security domains, mostly specialized and tailored towards technical approaches (e.g., malware detection, IoT), while no solution exists that copes in a similar way with security processes. This paper presents the first contribution in this direction, informing the methodology for creating a benchmark for the IMP from these works.

3.1 Quantitative IMP Assessment

Although no benchmark exists for IMP assessment, some efforts are present in the literature to evaluate some of its components. Among them, Alam et al. [3] address the data collection problem by proposing a framework for incident data based on a deep-learning solution to label incident categories of the collected data.

Other works focus on incident analysis, as Peng et al. [39] with an IM framework that integrates heterogeneous data and leverages data mining to support multi-criteria decision-making for incident response. Similarly, Piegorsch et al. [40] present an experimental evaluation of analytic technologies for risk and vulnerability assessment, using a multi-metric approach to evaluate vulnerabilities based on societal, hazard, and environmental indicators.

Finally, other solutions concentrate on incident modeling, as Jain et al. [31] that systematically integrates modeling and simulation tools to address a comprehensive incident response, including different contexts (from healthcare to natural disasters). In addition, Shaked et al. [46] and Riviera-Ortiz et al. [42] propose two model-based approaches to designing cyber security incident response playbooks and support logging procedures, respectively.

Contrary to our proposal, these works mostly address data management problems rather than security assessment. In fact, they do not propose solutions that support quantifiable and comparative analyses; rather, they concentrate on data collection, integration, and modeling due to their heterogeneous format.

4 BENCHMARK DESIGN

In this section, we describe the proposed approach for benchmarking IMP assessment models. First, we introduce the benchmark scope, formalizing the requirements that a benchmark should address to support the quantitative evaluation of IMPs. We then describe the design choices and the benchmark components.

4.1 Benchmark Scope and Requirements

To describe the benchmark scope and elicit its requirements, we present a motivating example.

Motivating example. Let us consider an organization that wants to get certification from ISO. For this purpose, it has defined its own assessment model that penalizes more severely the incidents that are non-compliant with ISO 27035 [27] to evaluate the incident costs accordingly. The reference process is reported in Fig. 1: it

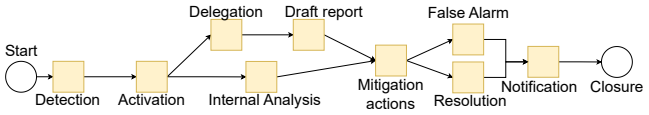


Figure 1: Incident Management Process from ISO 27035:2013.

starts with the detection of an incident followed by its activation (i.e., the opening of a ticket). Then, it is either analyzed internally or delegated to a third-party company and, based on the analysis, mitigation actions are put in place to resolve the incident or mark it as a false alarm. Finally, the incident is notified to all involved users and closed. Beyond the ISO 27035 process, the assessor has the IMP log with all the activities, personnel, and resources involved during the process in the last two years.

An assessor who evaluates the IMP of the organization must consider the IMP log and the developed assessment model, manually determine if the organization's model correctly assesses the incidents, and investigate the log (that, as in this case, can be very large). Although different log analysis tools exist [16], they do not correlate the log data to the assessment model. Thus the assessor should make a great effort to decide whether the assessment model is performing correctly. This task can be cumbersome, time-consuming, and error-prone because the assessor has no comparable methods to objectively evaluate the IMP assessment model.

Scope and requirements. Starting from the motivating example, we identify the benchmark scope as composed of three main use cases in the context of quantitative IMP assessment. They focus on the incident cost, as it is one of the main parameters driving the design of response strategies [43].

1) *Evaluation of an IMP assessment model*: the IMP assessment is based on the computation of incident costs, characterizing one or more features of the incidents. The assessment model used to assign costs to incidents may vary depending on the context, thus an assessor must provide an objective evaluation of the model, possibly compared with state-of-the-art solutions, to choose the most fitting one.

2) *Robustness of the IMP assessment model*: the IMP log may present different mistakes introduced by security operators [47] or errors coming from automated data collection (e.g., inaccurate sensors). Thus, the assessor must be supported by a quantitative indication of the robustness of the assessment models in the presence of errors in the log, where their nature and distribution resemble realistic scenarios [50].

3) *Indication of relevant incident features*: the log features play a central role in the IMP; therefore, the assessment must be supported by the analysis of their influence on the performance and robustness of the assessment models. This supports the assessor in designing mitigation actions accordingly (e.g., if s/he finds that the root cause of low robustness is the incident priority feature, then a possible process mitigation is more training in labeling incident priority).

Taking into account these general use cases, we defined a set of seven fine-grained requirements for the design of a benchmark, both specific for IMP assessment (R1-R4) and general for any benchmarking system according to existing guidelines (R5-R7) [21, 30].

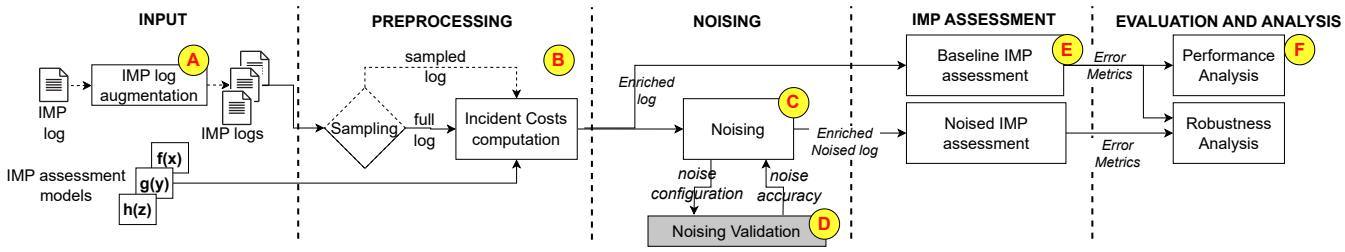


Figure 2: Benchmarking architecture. The necessary input is the IMP log, which is potentially augmented and sampled to accommodate fairness and scalability settings, respectively. We refer to *enriched log* as the log extended with the incident costs. Noise emulates potential errors in the input log to finally evaluate the performance and robustness of IMP assessment models.

R1: Quantifiable IMP assessment: The benchmark must provide a quantitative validation of the IMP assessment models that evaluate the incident costs based on IMP log features.

R2: Comparative analysis: The benchmark must provide comparative analyses with baseline assessment models.

R3: Performance of assessment models: The benchmark must be capable of providing suitable metrics that measure the performance of assessment models.

R4: Robustness of assessment models: The benchmark must be capable of providing suitable metrics that measure the robustness of assessment models to potential errors in the input log.

R5: Fairness: The benchmark must provide a fair assessment for different IMP logs.

R6: Extensibility: The benchmark should be easy to extend and use with any compatible system and input.

R7: Scalability: The workloads should be scalable to accommodate benchmarking with higher workloads to manage (i.e., more logs or assessment models) or on increasingly powerful hardware.

4.2 Benchmark Architecture

Fig. 2 presents the benchmark architecture with its components. The only necessary input is an IMP log containing data about the incident activities with their timestamps; additional incident features may be present and used to calculate the incident costs (e.g., impact, priority, involved personnel). Let us remark that while different works leverage technical logs [16, 42], the proposed benchmark system requires a *process log*, as the focus is on IMP assessment. The other input is the IMP assessment model(s) assigning costs to the incidents according to predefined criteria that may change depending on the context. While the benchmark mandates only the presence of the IMP log, it is agnostic on the presence of additional inputs. In this latter case, the benchmark integrates state-of-the-art assessment models [19, 33, 36, 43].

4.2.1 IMP Log Augmentation (Fig. 2.A). While in the literature there are different assessment models for IMP, one of the most critical challenges in the domain of security processes is the lack of log data since most of them are usually not shared publicly. To overcome this limitation, we introduce a log augmentation module that generates synthetic logs from real ones, considering reasonably realistic constraints to enable comparisons among different scenarios (addresses R5). The dotted arrows in Fig. 2 indicate that it is an optional module for cases where the user is not interested in testing more scenarios or focuses only on her own logs. For example, if an analyst wants to study the behavior of an assessment model under conservative assessment, then the actual log can be augmented by overestimating the incident impacts. It is the case when the impact is evaluated according to a categorical scale (i.e., low-medium-high) and the medium impact is overestimated to high [8].

4.2.2 Incident Cost Computation (Fig. 2.B). In the preprocessing phase, the benchmark calculates the incident costs using state-of-the-art assessment models (addresses R1, R2) and the additional ones given in input (addresses R5, R6). Given that incident logs may be very large, it is possible to configure a sampling mechanism to run the benchmark on a subset of the incidents, making the approach more scalable in terms of space and time resources (addresses R7). In Fig. 2, the sampling module is represented with dotted lines to indicate it is an optional configuration, while by default the benchmark runs on the whole input log. For example, let us consider the IMP of the motivating example and the incident execution of Table 1.

Incident ID	Timestamp	Activity	Operator	Impact
INC001	01-02-2016 18:42	detection	monitor	7
INC001	01-02-2016 19:12	activation	desk	7
INC001	01-02-2016 20:01	resolution	assessor	7
INC001	01-02-2016 20:11	closure	desk	7

Table 1: Example of an incident execution.

For the sake of example, let us consider an assessment model that calculates incident costs based on man-hours, and each operator in Table 1 works as a single person (i.e., no team). Then, the incident $INC001$ has a cost of 1.48, corresponding to the incident duration.

4.2.3 Noising (Fig. 2.C). The noising phase has the goal of introducing noise in the input log according to policies emulating realistic errors during the IMP (addresses R4). For example, a possible noise policy is decreasing the incident priority to emulate its underestimation from the assessor. According to the existing literature on log data quality [13, 32], we considered three noising policies:

(i) *missing data*: it represents the case in which the operator does not fill the log entries, or the automatic data collection system crashes or presents incomplete information. From a practical perspective, it is obtained by replacing the values of the log features

with null values. In the example of Table 1, missing data may be emulated with null values for the impact.

(ii) *imprecise data*: it represents the case in which the data collection system or the operator inserts a wrong value for a log feature, but it is still included in the feature domain, making the information inaccurate. To emulate this error, we replace the values of the log features with other values that are *within* the feature domain. The domain of a feature is the range of values that the feature has in the original log. In the example of Table 1 and assuming that incident impact is defined in the range [0,10], the imprecise value could be another value, different from 7 in the range [0,10].

(iii) *incorrect data*: it represents the case in which the operator inserts a wrong value or the data collection system is not reliable. Thus the error significantly impacts the log quality making the information incorrect. We emulate it by replacing the value of the log features with a different one out of the feature domain. In the example of Table 1 and assuming that incident impact is defined in the range [0,10], an incorrect incident impact may be a negative value or a value bigger than 10.

Based on these three noising policies, the noising is configured according to two parameters: the number of *log entries* to noise and the *noise magnitude*, which is the maximum difference between the noised data and the original one. For example, a feature f with value v that becomes dirty with 50% of imprecise noise magnitude corresponds to assign to f the value $v \pm 50\% \cdot v$, with the sign assigned randomly.

4.2.4 Noising Validation (Fig. 2.D). The noising phase may have different sources of randomness such as the selection of the log entries to which add noise. For this reason, the benchmark provides a noising validation module (Fig. 2.D) with a twofold objective. First, it verifies that the number of dirty entries corresponds to the noise configuration settings. For example, different types of noise could affect the same log entry, resulting in fewer dirty entries with respect to the configured ones. Secondly, it checks that the introduced noise meets the constraints of the configured magnitude. For example, a feature might be defined in the domain of values [0,10] (e.g., incident priority). If an instance of the feature is 10, then an imprecise noise with a magnitude greater than 10% may result in a violation of the domain. This means the imprecise data would become incorrect, thus changing the noising policy.

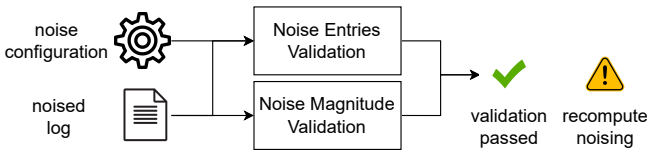


Figure 3: Noising validation component: it validates the noise entry accuracy and noise magnitude accuracy.

As shown in Fig. 3, the noising validation module takes as inputs the noise configuration settings (i.e., the number of dirty log entries and maximum magnitude and the set of features to noise) and the actual noised log. The output is an indication of the accuracy of the noising for each policy, based on which the benchmark either continues with the evaluation or automatically corrects the noising log to reach the requested level of noise.

More formally, the first step is the validation of the number of noised entries with respect to the configuration setting. We refer to *noise entry accuracy*, acc_{NE}, as a measure to validate the number of noised entries. Let $N = \{\text{missing, imprecise, incorrect}\}$ be the set of types of noise. Let E_n be the number of log entries with noise of type $n \in N$. Let E_{config} be the number of entries to be noised according to configuration parameters. Then:

$$acc_{NE} = \frac{\sum_{n \in N} E_n}{E_{config}} \tag{1}$$

Let us note that by definition of the noising module, the following constraint is always respected: $\sum_{n \in N} E_n \leq E_{config}$. Thus, based on the value of the noise entry accuracy, we define two validation thresholds (t_1, t_2) such that:

(i) if $t_1 \leq acc_{NE} < 1$, the noise is *strongly valid* because it respects the configured noise parameter;

(ii) if $t_2 \leq acc_{NE} < t_1$, the noise is *valid*, because a fewer number of entries than the configured ones are noised (e.g., due to the same entry noised with different types of noise);

(iii) if $acc_{NE} < t_2$, the noise is *invalid* because the number of noised entries is too far from the configured one.

If the value of acc_{NE} is strongly valid or valid, then the noising validation is passed (in the second case an operator could look at detailed results to understand potential causes for the misalignment between the requested noise level and the obtained one); otherwise, the noising is rejected. In this latter case, the system performs another noising by detecting the already noised entries to not overwrite them. This reflects the configured parameters, although losing part of its randomness. The rationale of this choice is to converge more rapidly by correcting the existing run instead of launching a new one.

The second validation step has the goal of checking that the random noise introduced for imprecise data does not overcome the features' domain, becoming incorrect data. In the example of Table 1, if the imprecise noise has magnitude +50%, the impact becomes 10.5 which is incorrect since it is out of the domain [0,10]. We refer to *noise magnitude accuracy*, acc_{NM}, as a measure of this validation:

$$acc_{NM} = \frac{e_{actual}}{e_{configured}}, \tag{2}$$

where e_{actual} and $e_{configured}$ are respectively the number of actual entries with imprecise data and the configured ones. By design of the noising module $e_{actual} \leq e_{configured}$. As for noise entry accuracy, acc_{NM} indicates the valid, almost valid, and invalid noise according to the same thresholds. In the case of rejection, the next iteration of noising reverses the sign of the magnitude of the entries noised with imprecise data with out-of-domain values. The rationale for this choice is the greedy convergence of the noised log to the nosing configuration. In this way, the next imprecise noised values will fall within the feature domain range. Let us note that neither the noising nor the noising validation modules should strictly depend on the log feature values because we do not make any assumption about the noise source to emulate the presence of error in the log without bias.

4.2.5 IMP Assessment (Fig. 2.E). This module has the goal of running each assessment model compared with all the state-of-the-art models integrated into the system, and considered one by one as *ground truth* (GT) to validate the model under analysis. Under the assumption that different state-of-the-art assessment models capture different scenarios of incident costs, this module supports the assessor in understanding the relations between different assessment models. The comparison is measured according to the current state-of-the-art *error metrics* for comparing observations:

(i) Mean Absolute Error (MAE) [12], measuring the average absolute difference between estimated and actual cost values;

(ii) Mean Squared Error (MSE) [12], measuring the average squared difference between the estimated and actual cost values;

(iii) Median Absolute Deviation (MAD) [44], which measures the variability of a univariate sample of data.

Considering the example of Table 1, one assessment model can calculate the costs based on man-hours, resulting in a cost equal to 1.48. Another assessment model may consider only the impact, resulting in a cost equal to 7. The above metrics measure the difference in the distribution between the costs assigned according to the different cost models (possibly normalized). The benchmark runs the assessment models both on the original input log (addresses R2, R3) and the noised one (addresses R2, R4). Let us remark that further customized assessment models as well as ground truths can be added to the benchmark at will, making it extensible — i.e., the process will be repeated for the added models considering added GTs — (addresses R6).

4.2.6 Evaluation and Analysis (Fig. 2.F). After the IMP assessment module computed the assessment models and gathered error metrics from each of them, these metrics collectively represent the *performance*. By performance of an assessment model, we mean its accuracy in the validation with the ground truths (i.e., state-of-the-art models), indicating how much its distribution is coherent with already validated (and thus more trusted) solutions — GT, manually or automatically obtained — (addresses R3). Given the potential discrepancy that the error metrics taken may introduce if considered singularly (e.g., the most accurate model could be different if considering MSE and MAD), we define a unique measure that considers all the error metrics to give a comprehensive score for the performance of the models. Formally, given an assessment model m evaluated with ground truth g and given $T = \{MAE, MSE, MAD\}$ the set of error metrics, then the *Multi-Metric Rank* (MMR) is:

$$MMR(m, g) = \sum_{t \in T} \alpha_t \cdot (1 - s_{t,m,g}), \tag{3}$$

where $s_{t,m,g}$ is the error metric t for the model m with ground truth g, and $\alpha_t \in [0, 1]$ is the weight of error t in the multi-metric rank configurable such that $\sum_{t \in T} \alpha_t = 1$. Finally, the MMR is normalized with min-max normalization [12] to define a ranking among the assessment models; a higher MMR corresponds to a better performance.

As a final evaluation, we compare the assessment models run with the noised log and the original one to quantify their *robustness* (addresses R4). By robustness of an assessment model, we mean its ability to cope with wrong inputs, keeping the level of performance consistent even in the presence of errors in the IMP log. More specifically, for assessment models to be considered robust, the error metric resulting from noised input has to be consistent with the same error metric resulting from original input [56], thus measuring the performance degradation in relation to wrong input logs. According to this definition, we evaluate the robustness of an assessment model m as:

$$RB(m) = 1 - \left(\frac{|MAE_{original} - MAE_{noise}|}{max(MAE_{original}, MAE_{noise})}\right), \tag{4}$$

where $MAE_{original}$ and MAE_{noise} are the MAE error metrics of model m with original and noised input respectively. The rationale of the MAE choice among the possible error metrics is that it avoids mutual cancellation of the positive and negative errors and represents the distance better than the MSE and MAD. Let us note that this definition of robustness does not consider log features because they affect the performance and they are captured by MAE, MSE, and MAD metrics. In contrast, robustness measures the variability of the performance.

5 THE BENCHIMP SYSTEM

In this section, we describe BenchIMP, the system that implements the proposed benchmarking approach for the quantitative evaluation of IMP assessment. We discuss the system details for each component of the benchmark architecture (see Fig. 2). All components use scientific Python as implementation technology [51].

Input. BenchIMP takes as input one or more IMP logs (in CSV or XES format) that must contain the incident IDs, activities for each incident, and their timestamp as minimum requirements. Additional log features (i.e., number of employees involved in each incident, priority, urgency, and impact) may be present for more comprehensive assessment models. The other input is one or more IMP assessment models, which are software modules that compute the cost of each incident based on the IMP log features.

Concerning the logs, BenchIMP has integrated two logs coming from real implementations of the IMP [6, 41]. They are open-source datasets and, to the best of the authors' knowledge, are the only ones freely available. The former is from the UCI machine learning repository and collects data from the audit system of the ServiceNow™ [45] platform used by an IT company[2], containing 24918 incidents. The latter is from 4TU ResearchData and collects data concerning a ticketing IMP of the Help desk of an Italian software company[3], containing 4580 incidents. Both of them are anonymized for privacy issues and, for each incident, report descriptive features related to the IMP (i.e., for each incident, its identifier, the different phases it is composed of, the timestamp, and the identifier of the people in charge of each phase), incident classification (i.e., incidents priority, category, and location), and incident diagnosis (i.e., impacted Service Level Agreement and the number of times the caller rejected the resolution).

Concerning the assessment models, BenchIMP integrates four of them to represent the current state-of-the-art of IMP assessment:

M1 [33]: The cost components are based on losses in revenue, additional expenses, and intangible costs of each incident, and then related to process metrics (e.g., penalty payments). The cost of an

[2]https://doi.org/10.24432/C57S4H

[3]https://data.4tu.nl/articles/_/12675977/1

incident is the sum of its *additional* expenses, that is $C = t_{add} * p_{add}$, where t_{add} and p_{add} are the working duration and the number of people involved in extra-work for solving the incident.

M2 [36]: The cost is estimated with the potential loss due to SLA violations as a function of the duration of each incident. The loss is the rate at which it is instantaneously accumulated at any given instant time, considering the priority of each process. Formally, $Loss^k = \sum_{\forall i | BP_i \in BP^k} \sum_j w_i \int_{t \in \tau_j^i} \beta_i^k (t + \delta_i^k) dt$, where $Loss$ is the cost due to SLA violations.

M3 [19]: The cost is associated with process resources through the cost of person-hours employed per process instance, thus the cost is $C(T) = \sum_i t_i * p_i$, where t_i and p_i are respectively the time and the number of people involved in the activity i of incident T.

M4 [43]: The cost computation is modeled as a linear regression problem to define the cost in relation to organization revenue (or the number of employees), compromised records, concurrency of incidents, and their impacts.

Log augmentation. Given the potential limitation of running the benchmark on only two datasets, the log augmentation module provides automatic support to generate synthetic realistic IMP logs. To this aim, BenchIMP leverages the *mixup* strategy for data augmentation [49, 57]. It consists of training a neural network on convex combinations of pairs of examples and their labels, so as to regularize the neural network to favor linear behavior in-between training examples. The rationale for this choice is that it presents improved generalization error compared to other state-of-the-art techniques for tabular datasets, such as the IMP log. Furthermore, mixup helps to combat the memorization of corrupt labels, sensitivity to adversarial examples, and instability in adversarial training [57]. BenchIMP trains the neural network on the features used by the input assessment models and uses them, one by one, as prediction labels. In this way, it builds one dataset per incident feature by augmenting the original logs and potentially re-training the neural networks with different samples to reach a configurable minimum number l of logs (by default $l = 10$ in BenchIMP).

Sampling. For the optional sampling of the input logs, BenchIMP leverages random sampling of the incidents with a configurable sampling rate. The rationale of the random sampling is due to its good trade-off between the decrease of computation time and the accuracy of the represented samples [1]. However, different sampling algorithms may be added and configured. Let us remark that sampling is not applied by default.

Noising configuration. BenchIMP performs different configurations of noising, progressively varying the number of dirty log entries and the magnitude of imprecise and incorrect data. We vary the percentage of noised entries from 0 to 50% of the entire log, while the magnitude from 0 to 50% for imprecise data and from 100% to 150% for incorrect data. These values result from the assumption that considering an incident log with more than 50% of errors is unrealistic: it would be assumable as randomly choosing its feature values, for which no guarantees of correctness could be provided.

5.1 The Resulting Benchmark

We run BenchIMP in a Linux server with Intel(R) Xeon(R) Gold 6248 CPU 2.50GHz and 256 GB RAM. It has been used to compute a publicly available benchmark[4] that, according to the described BenchIMP configurations, has the following settings:

- There are $l = 10$ IMP logs;
- There are $g = 4$ assessment models as baseline solutions;
- We select $f = 5$ log features that are noised, according to the ones used by the assessment models: incident duration, activities duration, incident priority, number of employees involved in the incident, and incident impact. We consider any combination of them for a total of 2^f combinations.
- There are $n = 3$ types of noise: missing, imprecise, and incorrect data, as provided by the benchmarking design (see Section 4.2). We considered any combination of them (noise with a single type, noise with any two of the types, and noise with all three types), for a total of 2^n combinations.
- We vary the percentage of noised entries of the original IMP log from 0 to 50% of the total number of log entries, with a step of 5%, for a total of $m = 10$ different configurations.
- We vary the magnitude of noise from 0 to 50% for imprecise data, and from 100% to 150% for incorrect data, both with steps of 5%, for a total of $p = 10$ different configurations that, considering all combinations between imprecise and incorrect data, results in 2^p different percentages.

Given any combination of the above parameters, the benchmark considers $l \cdot g \cdot 2^f \cdot 2^n \cdot m \cdot 2^p$ experiments, corresponding to 104,857,600 experiments, for evaluating the performance and robustness of an assessment model. The benchmark can be used by several stakeholders, such as researchers and security practitioners, to quantitatively evaluate IMP assessment models, their performance and robustness, and support decisions to implement in their operational environment (e.g., Security Operation Centers).

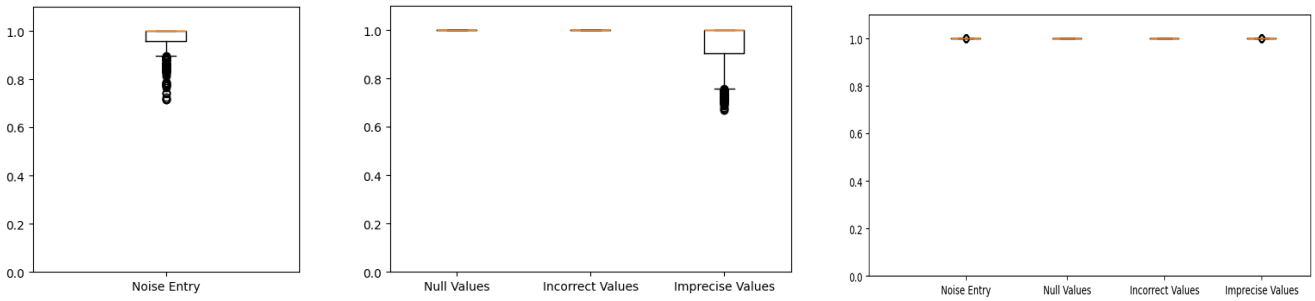
6 CASE STUDIES

This section presents two case studies showing how BenchIMP can be employed with different benefits. To present its capabilities, we leverage the motivating example of Section 4 that refers to the ISO 27035 process and a real IMP log [6]. The first case study shows how to evaluate a new assessment model, while the second one shows how BenchIMP can support the recommendation of assessment models according to an assessor's requirements.

6.1 Assessment model evaluation

The persona of the first case study is a security analyst who designed a novel IMP assessment model, and the goal is to *evaluate the new assessment model's performance and robustness to validate it with the current state-of-the-art IMP*. We considered one of the most recent assessment models proposed in the literature for the IMP [15] adjusted to measure the costs of incidents based on compliance with security standards [2]: it is a real assessment model currently employed in a compliance assessment system [38]. It leverages Extra-Trees Regression (ETR) [24], an ensemble technique that creates different decision trees, in which each node is a given incident

[4] https://github.com/Ale96Pa/BenchIMP

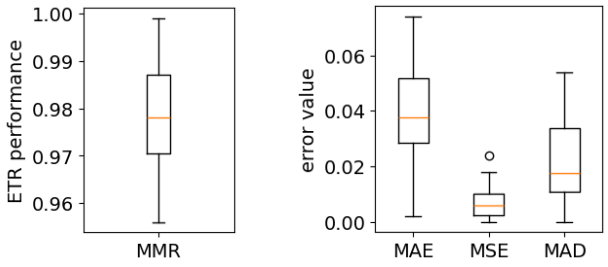


(a) Noised Entries Accuracy. (b) Noise Magnitude Accuracy. (c) Noise Accuracy after mitigation.

Figure 4: Accuracy of the (a) number and (b) magnitude of the noised log entries, and (c) the result after the mitigation.

feature and the tree minimizing the Gini impurity [14] is used to estimate the incident costs. In the following, we refer to this model as *ETR*, and an analyst seeks to evaluate it for potential integration into her organization.

6.1.1 Performance analysis. The performance evaluation is the first step for the *ETR* quantitative assessment. To this aim, the analyst leverages the multi-metric rank (MMR) as a performance measure, considering MSE, MAE, and MAD metrics together, as reported in Fig. 5a. It supports the analyst in having a unique indication of the performance of the proposed assessment model. Indeed, it shows that *ETR* has a median MMR of 0.98, which is a high value given that it is defined between 0 (bad performance) and 1 (good performance) as described in Section 4.2.6. It corresponds to the aggregated measure of the state-of-the-art error metrics against all the existing assessment models used as ground truths as described in Section 5. Their analysis is shown in Fig. 5b, where each boxplot represents the distribution of MAE, MSE, and MAD, respectively. A



(a) Multi-Metric Rank. (b) Error metrics.

Figure 5: Distribution of the (a) Multi-Metric Rank and (b) error metrics for the ETR assessment model.

relevant aspect is that the proposed *ETR* has excellent performance when considering MSE with a median error of 0.005 and little variability (small size of the box), while it has higher variability when considering MAE, although the median value of 0.04 is still low. This indicates that the *ETR* model has a good performance when compared with the state-of-the-art models, and the analyst can conclude that it is a good model for her organization because it is coherent with the current assessment models and, at the same time, customized for the organization's needs. Without the benchmark, the analyst requires more effort to have a quantitative indication of how much the proposed model differs from the existing (and already validated) ones, which measure extra expenses, resources, and SLA violations of the IMP. Beyond the good performance of *ETR*, the analyst must evaluate its robustness to error in the log.

6.1.2 Noising Validation. The next step is the robustness analysis to determine how much ETR can tolerate potential (human or machine) errors in the IMP log. Before this, the analyst must first investigate the correctness of the introduced noise. Fig. 4a reports the distribution of the noise entry accuracy according to the different benchmark experiments. Although the median is high (i.e., median accuracy of 0.99), it is worth noting that more than 25% of the experiments have an accuracy of 0.85, while 21 experiments (outliers in the boxplot) can even reach an accuracy lower than 0.65. This results from the random selection of the entries to be noised and the fact that the same entry can be noised with more than one type of error for different log features. Similarly, Fig. 4b shows the distribution of the magnitude accuracy. As expected, the missing and incorrect noise are 100% accurate because their magnitude randomness does not affect the features' domain. On the contrary, for the imprecise noise, almost 50% of the cases show an accuracy lower than 0.95, with 25% of them being lower than 0.8 and 48 experiments (outliers in the boxplot of Fig. 4b) reaching an accuracy of 0.45.

It should be noted that in all the cases, the median accuracy is 0.99. However, in some scenarios (e.g., critical infrastructures, budget-constrained scenarios), the presence of outliers with very low accuracy may not be tolerated for the robustness evaluation. For this reason, the analyst leverages the correction mechanism employed in BenchIMP (see Section 4.2.4) to adjust the number of dirty entries according to the configuration parameters. The resulting accuracy metrics after the correction are reported in Fig. 4c. After the second run of the noising procedure, we get 100% of noise entry accuracy and noise magnitude accuracy because the correction is deterministic and limits the errors introduced by the randomness. This shows how the noise management of the framework allows for fast convergence to the desired noise levels for a generic noise configuration.

6.1.3 Robustness analysis. Once the introduced noise is properly validated, the next step is the robustness evaluation of the assessment model. Fig. 6 reports the robustness distribution according

Figure 6: Robustness analysis according to missing, imprecise, and incorrect data for ETR assessment model.

to the types of errors analyzed separately, i.e., considering only a single type of data noise. This analysis makes it noticeable that the incorrect data noise causes a median robustness degradation of 20% and a mean degradation of 50%. This is much higher than missing and imprecise noises, which have a median robustness close to 1. We may expect this result since imprecise data results in out-of-domain feature values that lead to excessively low/high incident costs. This indicates that an important policy is to avoid imprecise data, and the analyst may implement this policy, for example, by assigning constraints to the log features during the IMP. However, cases with only one type of noise in isolation may be unrealistic and unrepresentative. For this reason, the next analysis considers the impact of the different types of noise in their combined presence. Fig. 7 reports the trend of the robustness distribution to the

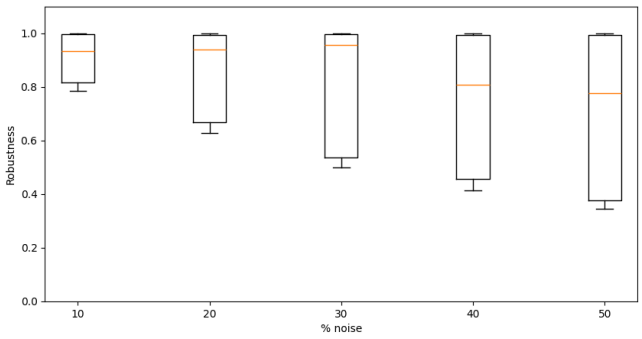


Figure 7: Robustness trend on the increasing percentage of data noise in the presence of missing, incorrect, and imprecise noise for the ETR assessment model.

percentage of data noise, including all types of noise. It shows the robustness degradation of *ETR* when increasing the amount of noise in the log, reaching a robustness score of 0.3 in the worst cases. In particular, there is an evident degradation of the median robustness when the IMP log is noised for more than 40%. This means that the proposed assessment model is good only in scenarios where the analyst can consider the IMP log at least 40% accurate; otherwise, its estimations may begin to degrade.

To further investigate the robustness and its degradation, the analyst studies the features that mostly impact the assessment model robustness to refine it eventually. Fig. 8 reports the distribution of the *ETR* model robustness according to incident duration, priority, number of involved employees, impact, and urgency, which are the log features used by the assessment model. The boxplots indicate that the number of employees feature has the highest impact, degrading the median robustness to 0.65, while the other features do not significantly impact it (median values of 0.99). This means that the number of employees is a critical feature for *ETR*, and this indicates to the analyst that a possible improvement of the assessment model may be weighting the features differently (eventually excluding some of them as being unreliable) to make it more robust.

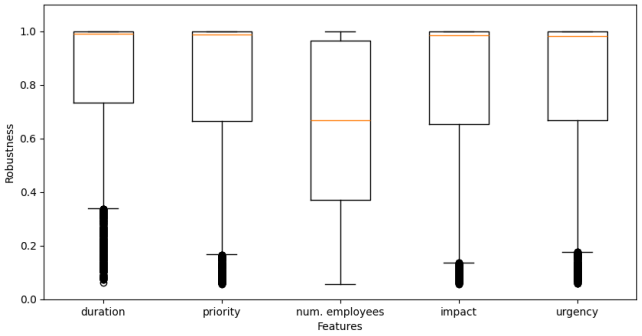


Figure 8: Robustness distribution analysis with the presence of noise for the different IMP log features of the ETR assessment model.

With this last analysis, the analyst can conclude that the *ETR* model has very good performance with a median MMR of 0.98 and high robustness to missing and imprecise errors with median values of 0.95. However, it can be further improved to better tolerate the incorrect errors in the IMP log by assigning a lower weight to the number of employees feature.

6.2 Recommendation of Assessment Models

The persona of the second case study is a security assessor who wants to adopt an existing assessment model for her IMP, and the goal is *to support the assessor in choosing the assessment model that best suits her needs by exploiting comparative analysis.* For the sake of this case study, we consider the four assessment models integrated in BenchIMP. Let us remember that model $M1$ penalizes the extra activities in the process, $M2$ assigns costs to SLA violations, $M3$ evaluates the expenses of the involved resources, and $M4$ models a linear regression solution. As a first step, the assessor compares the performance of the different assessment models according to the MMR (Fig. 9).

Looking at Fig. 9, models $M1$ and $M4$ present the highest MMR and also the more stable errors, with most of the values between 0.97 and 0.99. Indeed, a higher MMR indicates that the model presents fewer errors when compared with all the other ones. In contrast,

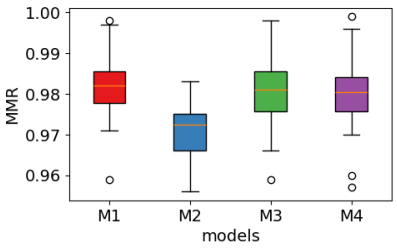


Figure 9: Comparison of the performance of the assessment models through the Multi-Metric Rank (MMR).

the $M3$ model shows a great variability of the MMR: this can be attributed to the fact that the log features used in this model (i.e., incident expenses) are heterogeneous values that may not be entirely representative of the process workflow. In addition, model $M2$ presents the lowest MMR, indicating that it is more prone to errors. Thus, from the performance analysis, the best candidates for the assessor are $M1$ and $M4$ models.

The next step of the analysis aims to study the robustness of the assessment models under analysis, reported in Fig. 10. It shows

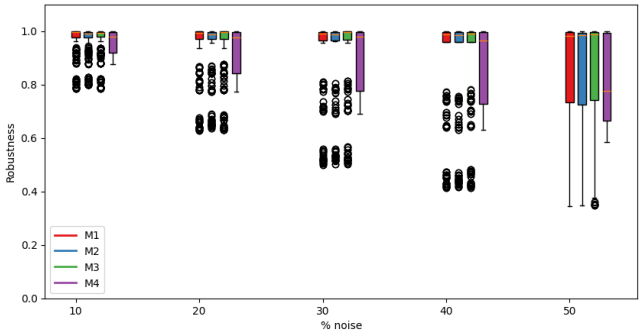


Figure 10: Comparison of the robustness of the assessment models.

the robustness degradation of all the assessment models when increasing the amount of noise in the log. An interesting aspect is that model $M4$ has the worst robustness trend, with degradation visible already with 10% of noise in the log. In contrast, model $M1$ has the best robustness, with a high median (equal to 0.9) even in the case of maximum noise. In addition, most of the degradation comes after 40%. Thus, the assessor can conclude that the most fitting assessment model for her needs is $M1$ for its best performance and robustness. BenchIMP could even be used to instrument an automatic recommendation system leveraging its data, asking the assessor to specify acceptable thresholds for performance, robustness, and log features available, in order to recommend the best assessment models respecting the specified constraints or looking for sub-optimal solutions.

In summary, these case studies showed how BenchIMP supports security practitioners and researchers in the quantitative analysis of the performance and robustness of a newly proposed assessment model for IMP. It also allows comparability among different models, supporting the identification of the most appropriate assessment model to use for an IMP log. Without the proposed benchmark, the assessor or researcher could have validated the assessment models only in a specific scenario, requiring much effort to compare different assessment models.

7 LIMITATIONS AND OPPORTUNITIES

This paper proposed a benchmarking approach, a system that implements it, and the resulting benchmark for IMP assessment. To the best of the authors' knowledge, it is the first contribution that allows for quantitatively evaluating and comparing IMP assessment models. The case studies showed the benefits of using BenchIMP: (i) quantitatively and objectively evaluate the performance of an assessment model; (ii) accurately emulate human and process errors of the data collection; (iii) quantitatively evaluate the robustness of the assessment models to errors in the log; (iv) support the decision-making on the best assessment models based on their performance and robustness; (v) enable comparability and reproducibility of the assessment model validation.

In this section, we first report on some of the limitations of the proposed approach. A current limitation is the BenchIMP extensibility. BenchIMP has been designed to be easily extended with different input logs, assessment models, and adjustable analytical parameters. However, to add these elements, the user must download the benchmark code, add the custom components, change the configuration file, and launch the benchmark. We plan for future work to provide a more user-friendly solution that supports users on the incremental extension of the benchmark and its customization.

Another limitation concerns the high computation cost of both space and time necessary to run all the experiments in the benchmark. We mitigated this problem by providing a sampling module to support incremental computation. Further solutions are currently being investigated and tested, such as the application of distributed and parallel computing.

Beyond limitations, the proposed benchmark opens up novel opportunities for development and research.

Benchmark generality. Although we modeled the benchmark, the quantitative validation, and the case studies in the context of IMP, we believe they can be easily generalized to other security processes by suitably formatting the input log and retrieving the appropriate assessment models from the literature. The input log must be an event log, i.e. containing trace IDs, activities, and their timestamps. Additional features for the traces should be present to compute their cost (e.g., the resources involved during the process). The assessment models must use features of the input log to compute the costs. For example, in the case of Intrusion Detection Systems (IDS), the log includes the set of the collected alerts for each target host. The alerts in the same host can be modeled as different activities (i.e., different steps of an attack). There exist different models in the literature to assess detection features that can be used as state-of-the-art assessment models [4, 55]. Thus, by leveraging the benchmark, the assessor can determine the performance and robustness of these models and use the most appropriate to evaluate IDSs. In such a case, the introduced noise may emulate

the possible IDS malfunctioning (e.g., crashed, imprecise, or incorrect sensors). We foresee efforts from ourselves and other authors to correctly apply the benchmarking approach to create specific benchmarks for the other security processes not discussed in this work.

Benchmark extensibility. According to the general benchmark system requirements [21, 30], the proposed benchmark is highly customizable and easy to configure and put into action thanks to its modularity. Concerning customization, the assessor can use different features of the input log to model the assessment models. In particular, they do not necessarily need to be state-of-the-art models, but even private/third-party ones can be used. This enables the usage of this benchmark in different public and private contexts and supports security researchers in validating novel assessment models for security processes. On the other hand, one can use the benchmark with default configurations (as presented for BenchIMP), or s/he can easily configure other benchmark parameters, that are the number of cores to run the benchmark (for scalability), and noising parameters, sampling rates, log features used by the models, and error metrics for the multi-metric analysis.

8 CONCLUSION

This paper presented a first step toward the usage of benchmarking approaches for the quantitative and comparable evaluation of IMP assessment models. We presented a set of requirements and a general approach for benchmarking it. This reduces the potential subjective bias introduced by manual assessment approaches. We contributed BenchIMP, the first benchmark system performing IMP assessment, allowing the comparative analysis of state-of-the-art solutions for the estimation of incident costs and supporting the development of new models for this task. Finally, we showed the usage of BenchIMP in two case studies highlighting the benefits enabled by our proposal. On the one hand, it supports security researchers and practitioners in validating newly proposed assessment models for IMP, as BenchIMP defines how to evaluate performance and robustness quantitatively. On the other hand, it helps security assessors and decision-makers select the assessment models that best support their IMP. For these reasons, BenchIMP is publicly accessible[5].

We plan as future work to develop a platform in which security practitioners can upload and share the results of their assessment models used in the benchmark. We also plan to leverage this platform to involve real stakeholders to enhance the usability of the benchmark and its further extension to foster its adoption.

ACKNOWLEDGMENTS

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU and by the project COVERT: In searCh Of eVidence of stEalth cybeR Threats under the Spoke 3 “Attacks and Defences” of project SERICS (PE00000014) (CUP: B85E22002000005).

[5]https://github.com/Ale96Pa/BenchIMP

REFERENCES

[1] Anita S Acharya, Anupam Prakash, Pikee Saxena, and Aruna Nigam. 2013. Sampling: Why and how of it. *Indian Journal of Medical Specialties* 4, 2 (2013), 330–333. https://doi.org/10.7713/ijms.2013.0032

[2] Giacomo Acitelli, Marco Angelini, Silvia Bonomi, Fabrizio M. Maggi, Andrea Marrella, and Alessandro Palma. 2022. Context-Aware Trace Alignment with Automated Planning. In *2022 4th International Conference on Process Mining (ICPM)*. 104–111. https://doi.org/10.1109/ICPM57379.2022.9980649

[3] Firoj Alam, Umair Qazi, Muhammad Imran, and Ferda Ofli. 2021. HumAID: Human-Annotated Disaster Incidents Data from Twitter with Deep Learning Benchmarks. *Proceedings of the International AAAI Conference on Web and Social Media* 15 (May 2021), 933–942. https://doi.org/10.1609/icwsm.v15i1.18116

[4] Hashim Albasheer, Maheyzah Md Siraj, Azath Mubarakali, Omer Elsier Tayfour, Sayeed Salih, Mosab Hamdan, Suleman Khan, Anazida Zainal, and Sameer Kamarudeen. 2022. Cyber-attack prediction based on network intrusion detection systems for alert correlation techniques: a survey. *Sensors* 22, 4 (2022), 1494. https://doi.org/10.3390/s22041494

[5] Naif Saleh Almakhdhub, Abraham A. Clements, Mathias Payer, and Saurabh Bagchi. 2019. BenchIoT: A Security Benchmark for the Internet of Things. In *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 234–246. https://doi.org/10.1109/DSN.2019.00035

[6] Claudio Amaral, Marcelo Fantinato, and Sarajane Peres. 2019. Incident management process enriched event log. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C57S4H.

[7] Saurabh Amin, Galina A. Schwartz, and Alefiya Hussain. 2013. In quest of benchmarking security risks to cyber-physical systems. *IEEE Network* 27, 1 (Jan. 2013), 19–24. https://doi.org/10.1109/MNET.2013.6423187 Conference Name: IEEE Network.

[8] Marco Angelini, Silvia Bonomi, Claudio Ciccotelli, and Alessandro Palma. 2020. Toward a context-aware methodology for information security governance assessment validation. In *International Workshop on Cyber-Physical Security for Critical Infrastructures Protection*. Springer, 171–187. https://doi.org/10.1007/978-3-030-69781-5_12

[9] Anusha Bambhore Tukaram, Simon Schneider, Nicolás E. Díaz Ferreyra, Georg Simhandl, Uwe Zdun, and Riccardo Scandariato. 2022. Towards a Security Benchmark for the Architectural Design of Microservice Applications. In *Proceedings of the 17th International Conference on Availability, Reliability and Security* (Vienna, Austria) *(ARES '22)*. Association for Computing Machinery, New York, NY, USA, Article 116, 7 pages. https://doi.org/10.1145/3538969.3543807

[10] Massimo Battaglioni, Giulia Rafaiani, Franco Chiaraluce, and Marco Baldi. 2022. MAGIC: A Method for Assessing Cyber Incidents Occurrence. *IEEE Access* 10 (2022), 73458–73473. https://doi.org/10.1109/ACCESS.2022.3189777

[11] Simona Bernardi, Juan L. Domínguez, Abel Gómez, Christophe Joubert, José Merseguer, Diego Perez-Palacin, José I. Requeno, and Alberto Romeu. 2018. A systematic approach for performance assessment using process mining: An industrial experience report. *Empirical Software Engineering* 23, 6 (Dec. 2018), 3394–3441. https://doi.org/10.1007/s10664-018-9606-9

[12] Peter J Bickel and Kjell A Doksum. 2015. *Mathematical statistics: basic ideas and selected topics, volumes I-II package*. Chapman and Hall/CRC.

[13] R.P. Jagadeesh Chandra Bose, Ronny S. Mans, and Wil M.P. van der Aalst. 2013. Wanna improve process mining results?. In *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. 127–134. https://doi.org/10.1109/CIDM.2013.6597227

[14] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. 2017. *Classification and regression trees*. Routledge.

[15] Sharon Christa, V Suma, and Uma Mohan. 2022. Regression and decision tree approaches in predicting the effort in resolving incidents. *International Journal of Business Information Systems* 39, 3 (2022), 379–399. https://doi.org/10.1504/IJBIS.2022.122342

[16] Rafael Copstein, Jeff Schwartzentruber, Nur Zincir-Heywood, and Malcolm Heywood. 2021. Log Abstraction for Information Security: Heuristics and Reproducibility. In *Proceedings of the 16th International Conference on Availability, Reliability and Security* (Vienna, Austria) *(ARES '21)*. Association for Computing Machinery, New York, NY, USA, Article 93, 10 pages. https://doi.org/10.1145/3465481.3470083

[17] Frank Cremer, Barry Sheehan, Michael Fortmann, Arash N Kia, Martin Mullins, Finbarr Murphy, and Stefan Materne. 2022. Cyber risk and cybersecurity: a systematic review of data availability. *The Geneva Papers on risk and insurance-Issues and practice* 47, 3 (2022), 698–736. https://doi.org/10.1057/s41288-022-00266-6

[18] Dairo de Ruck, Victor Goeman, Michiel Willocx, Jorn Lapon, and Vincent Naessens. 2023. Linux-based IoT Benchmark Generator For Firmware Security Analysis Tools. In *Proceedings of the 18th International Conference on Availability, Reliability and Security (ARES '23)*. Association for Computing Machinery, New York, NY, USA, Article 19, 10 pages. https://doi.org/10.1145/3600160.3600181

[19] Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A Reijers, et al. 2013. *Fundamentals of business process management*. Vol. 1. Springer.

[20] Tudor Dumitraș and Darren Shou. 2011. Toward a standard benchmark for computer security research: the worldwide intelligence network environment (WINE). In *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*. ACM, Salzburg Austria, 89–96. https://doi.org/10.1145/1978672.1978683

[21] Jerrit Eickhoff, Jesse Donkervliet, and Alexandru Iosup. 2023. Meterstick: Benchmarking Performance Variability in Cloud and Self-hosted Minecraft-like Games. In *Proceedings of the 2023 ACM/SPEC International Conference on Performance Engineering* (, Coimbra, Portugal,) *(ICPE '23)*. Association for Computing Machinery, New York, NY, USA, 173–185. https://doi.org/10.1145/3578244.3583724

[22] ENISA. 2010. Good Practice Guide for Incident Management. https://www.enisa.europa.eu/publications/good-practice-guide-for-incident-management

[23] Alexander A Ganin, Phuoc Quach, Mahesh Panwar, Zachary A Collier, Jeffrey M Keisler, Dayton Marchese, and Igor Linkov. 2020. Multicriteria decision framework for cybersecurity risk assessment and management. *Risk Analysis* 40, 1 (2020), 183–199. https://doi.org/10.1111/risa.12891

[24] Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine learning* 63, 1 (2006), 3–42. https://doi.org/10.1007/s10994-006-6226-1

[25] G. Gonzalez-Granadillo, S. Dubus, A. Motzek, J. Garcia-Alfaro, E. Alvarez, M. Merialdo, S. Papillon, and H. Debar. 2018. Dynamic risk management response system to handle cyber threats. *Future Generation Computer Systems* 83 (2018), 535–552. https://doi.org/10.1016/j.future.2017.05.043

[26] S. Zohra Halim, Mengxi Yu, Harold Escobar, and Noor Quddus. 2020. Towards a causal model from pipeline incident data analysis. *Process Safety and Environmental Protection* 143 (2020), 348–360. https://doi.org/10.1016/j.psep.2020.06.047

[27] ISO. 2013. *Part 1: Principles of incident management; Part 2: Guidelines to plan and prepare for incident response; Part 3: Guidelines for ICT incident response operations*. Standard. International Organization for Standardization, Geneva, CH.

[28] ISO. 2014. *Quality Management Systems*. Standard. International Organization for Standardization, Geneva, CH.

[29] ITIL. 2019. *Information Technology Infrastructure Library*. Standard. Axelos, UK.

[30] Raj Jain. 1991. *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. Vol. 1. Wiley New York.

[31] Sanjay Jain and Charles R. McLean. 2006. An Integrating Framework for Modeling and Simulation for Incident Management. *Journal of Homeland Security and Emergency Management* 3, 1 (March 2006). https://doi.org/10.2202/1547-7355.1194 Publisher: De Gruyter.

[32] Mohammed Oussama Kherbouche, Nassim Laga, and Pierre-Aymeric Masse. 2016. Towards a better assessment of event logs quality. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. 1–8. https://doi.org/10.1109/SSCI.2016.7849946

[33] Axel Kieninger, Florian Berghoff, Hansjörg Fromm, and Gerhard Satzger. 2013. Simulation-Based Quantification of Business Impacts Caused by Service Incidents. In *Exploring Services Science*, João Falcão e Cunha, Mehdi Snene, and Henriqueta Nóvoa (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 170–185. https://doi.org/10.1007/978-3-642-36356-6_13

[34] Jesus Luna Garcia, Robert Langenberg, and Neeraj Suri. 2012. Benchmarking cloud security level agreements using quantitative policy trees. In *Proceedings of the 2012 ACM Workshop on Cloud computing security workshop (CCSW '12)*. Association for Computing Machinery, New York, NY, USA, 103–112. https://doi.org/10.1145/2381913.2381932

[35] Elinor M. Madigan, Corey Petrulich, and Kelly Motuk. 2004. The cost of noncompliance: when policies fail. In *Proceedings of the 32nd Annual ACM SIGUCCS Conference on User Services* (Baltimore, MD, USA) *(SIGUCCS '04)*. Association for Computing Machinery, New York, NY, USA, 47–51. https://doi.org/10.1145/1027802.1027815

[36] A. Moura, J. Sauve, J. Jornada, and E. Radziuk. 2006. A quantitative approach to IT investment allocation to improve business results. In *Seventh IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'06)*. 9 pp.–95. https://doi.org/10.1109/POLICY.2006.7

[37] Rui André Oliveira, Miquel Martínez Raga, Nuno Laranjeiro, and Marco Vieira. 2020. An approach for benchmarking the security of web service frameworks. *Future Generation Computer Systems* 110 (Sept. 2020), 833–848. https://doi.org/10.1016/j.future.2019.10.027

[38] Alessandro Palma and Marco Angelini. 2024. Visually Supporting the Assessment of the Incident Management Process . In *EuroVis Workshop on Visual Analytics (EuroVA)*, Mennatallah El-Assady and Hans-Jörg Schulz (Eds.). The Eurographics Association. https://doi.org/10.2312/eurova.20241116

[39] Yi Peng, Yong Zhang, Yu Tang, and Shiming Li. 2011. An incident information management framework based on data integration, data mining, and multicriteria decision making. *Decision Support Systems* 51, 2 (May 2011), 316–327. https://doi.org/10.1016/j.dss.2010.11.025

[40] Walter W. Piegorsch, Susan L. Cutter, and Frank Hardisty. 2007. Benchmark Analysis for Quantifying Urban Vulnerability to Terrorist Incidents. *Risk Analysis* 27, 6 (2007), 1411–1425. https://doi.org/10.1111/j.1539-6924.2007.00977.x

[41] Mirko Polato. 2017. Dataset belonging to the help desk log of an Italian Company. Version 1. 4TU.ResearchData. dataset. DOI: https://doi.org/10.4121/uuid:0c60edf1-6f83-4e75-9367-4c63b3e9d5bb.

[42] Fanny Rivera-Ortiz and Liliana Pasquale. 2020. Automated modelling of security incidents to represent logging requirements in software systems. In *Proceedings of the 15th International Conference on Availability, Reliability and Security* (Virtual Event, Ireland) *(ARES '20)*. Association for Computing Machinery, New York, NY, USA, Article 35, 8 pages. https://doi.org/10.1145/3407023.3407081

[43] Sasha Romanosky. 2016. Examining the costs and causes of cyber incidents. *Journal of Cybersecurity* (Aug. 2016), tyw001. https://doi.org/10.1093/cybsec/tyw001

[44] Peter J. Rousseeuw and Christophe Croux. 1993. Alternatives to the Median Absolute Deviation. *J. Amer. Statist. Assoc.* 88, 424 (1993), 1273–1283. https://doi.org/10.1080/01621459.1993.10476408

[45] ServiceNow. 2022. *ServiceNow-TM*. Standard. Gildesoft, USA.

[46] Avi Shaked, Yulia Cherdantseva, and Pete Burnap. 2022. Model-Based Incident Response Playbooks. In *Proceedings of the 17th International Conference on Availability, Reliability and Security* (Vienna, Austria) *(ARES '22)*. Association for Computing Machinery, New York, NY, USA, Article 26, 7 pages. https://doi.org/10.1145/3538969.3538976

[47] Mikko Siponen and Robert Willison. 2009. Information security management standards: Problems and solutions. *Information & Management* 46, 5 (2009), 267–270. https://doi.org/10.1016/j.im.2008.12.007

[48] Solarwind. 2021. *Solarwind*. Standard. SolarWinds Corporation, USA.

[49] Lim Soon Hoe, Erichson N. Benjamin, Utrera Francisco, Xu Winnie, and Mahoney Michael W. 2021. Noisy Feature Mixup. arXiv:2110.02180 [cs.LG]

[50] Haipei Sun, Kun Wu, Ting Wang, and Wendy Hui Wang. 2022. Towards Fair and Robust Classification. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. 356–376. https://doi.org/10.1109/EuroSP53844.2022.00030

[51] Guido Van Rossum and Fred L Drake Jr. 1995. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

[52] A.J. Varela-Vaca, Rafael M. Gasca, and A. Jimenez-Ramirez. 2011. A Model-Driven engineering approach with diagnosis of non-conformance of security objectives in business process models. In *2011 fifth International Conference On Research Challenges in Information Science*. 1–6.

[53] Junhua Wang, Boya Liu, Ting Fu, Shuo Liu, and Joshua Stipancic. 2019. Modeling when and where a secondary accident occurs. *Accident Analysis & Prevention* 130 (2019), 160–166. https://doi.org/10.1016/j.aap.2018.01.024 Road Safety Data Considerations.

[54] Pei Wang, Jinquan Zhang, Shuai Wang, and Dinghao Wu. 2020. Quantitative Assessment on the Limitations of Code Randomization for Legacy Binaries. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. 1–16. https://doi.org/10.1109/EuroSP48549.2020.00009

[55] Charles Xosanavongsa, Eric Totel, and Olivier Bettan. 2019. Discovering Correlations: A Formal Definition of Causal Dependency Among Heterogeneous Events. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. 340–355. https://doi.org/10.1109/EuroSP.2019.00033

[56] Huan Xu and Shie Mannor. 2012. Robustness and generalization. *Machine learning* 86 (2012), 391–423.

[57] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2017. mixup: Beyond Empirical Risk Minimization. *CoRR* abs/1710.09412 (2017). arXiv:1710.09412 http://arxiv.org/abs/1710.09412