On the Difficulty of Identifying Incident-Inducing Changes

Kapel, Eileen; Cruz, Luis; Spinellis, Diomidis; Van Deursen, Arie

DOI
10.1145/3639477.3639755

Publication date
2024

Document Version
Final published version

Published in
Proceedings - 2024 ACM/IEEE 44th International Conference on Software Engineering

2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)

On the Difficulty of Identifying Incident-Inducing Changes

Eileen Kapel
Eileen.Kapel@ing.com
ING Bank
Amsterdam, Noord-Holland, The Netherlands

Luís Cruz
l.cruz@tudelft.nl
Delft University of Technology
Delft, Zuid-Holland, The Netherlands

Diomidis Spinellis
d.spinellis@tudelft.nl
Delft University of Technology
Delft, Zuid-Holland, The Netherlands

Arie van Deursen
arie.vandeursen@tudelft.nl
Delft University of Technology
Delft, Zuid-Holland, The Netherlands

ABSTRACT

Effective change management is crucial for businesses heavily reliant on software and services to minimise incidents induced by changes. Unfortunately, in practice it is often difficult to effectively use artificial intelligence for IT Operations (AIOps) to enhance service management, primarily due to inadequate data quality. Establishing reliable links between changes and the induced incidents is crucial for identifying patterns, improving change deployment, identifying high-risk changes, and enhancing incident response. In this research, we investigate the enhancement of traceability between changes and incidents through AIOps methods. Our approach involves a close examination of incident-inducing changes, the replication of methods linking incidents to the changes that caused them, introducing an adapted method, and demonstrating its results using historical data and practical evaluations. Our findings reveal that incident-inducing changes exhibit different characteristics dependent on context. Furthermore, a significant disparity exists between assessments based on historical data and real-world observation, with an increased occurrence of false positives when identifying links between unlabeled changes and incidents. This study highlights the complex nature of identifying links between changes and incidents, emphasising the contextual influence on AIOps method effectiveness. While we are actively working on improving the quality of current data through AIOps approaches, it remains apparent that further measures are necessary to address issues like data imbalances and promote a postmortem culture that brings attention to the value of properly administrating tickets. A better overview of change failure rates contributes to improved risk compliance and reliable change management.

CCS CONCEPTS

• **Software and its engineering → Risk management**; **Software post-development issues**; **Software reliability**.

ICSE 2024, April 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0501-4/24/04…$15.00
https://doi.org/10.1145/3639477.3639755

KEYWORDS

change management, incident management, traceability

ACM Reference Format:
Eileen Kapel, Luís Cruz, Diomidis Spinellis, and Arie van Deursen. 2024. On the Difficulty of Identifying Incident-Inducing Changes. In *46th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP '24), April 14–20, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3639477.3639755

1 INTRODUCTION

Nowadays, major businesses and industries, such as banking, healthcare and retail, are increasingly reliant on software and related services, often referred to as software-defined businesses [1]. This approach emphasises an agile way of working, encouraging fast-paced development and deployment of new features. While this offers many benefits, it also leads to a higher volume and faster pace of change deployment. In this context, changes are defined as modifications to existing applications, comprising additions, alterations and deletions [18].

Changes can occasionally trigger incidents, especially if they undergo inadequate testing or incorrect implementation. For instance, a software upgrade might lead to compatibility issues, resulting in an incident. The change management process aims to minimise change-related risks, thereby reducing the occurrence of incidents caused by changes. An incident is defined as an unplanned interruption to a service or a reduction in service quality at a specific time [18]. Incidents can lead to customer dissatisfaction, financial losses, and reputational damage; therefore, they must be prevented. IT service changes are one of the leading contributors to outages, accounting for about 70% of live system outages [5]. This is often referred to as the change failure rate, which measures the percentage of deployments causing a failure in production [15]. Other factors that contribute to the difficulty of identifying incident-inducing changes include the complex nature of large IT environments [6], the potential for seemingly successful changes to still induce incidents [16, 38], the service where an incident begins may be different from the once that caused it [36], the necessity for engineers to sift through large amounts of heterogeneous data to identify the root cause [36], and the fact that incidents are often the result of changes occurring hours or days before the incident [17].

To enable the application of Artificial Intelligence for IT Operations (AIOps), which utilises big data, machine learning and advanced analytics to enhance IT operations [21], proper data quality is crucial to systemically learn from the past and ensure effective

service management [5]. In practice, the quality of IT service management data may vary due to manual reporting and analysis [29]. Currently research strongly advocates for improving data quality for AIOps applications [2, 11, 29]. One area contributing to service management data quality is software traceability, the process of linking diverse artefacts to track their life cycle [10]. The cost, effort, and discipline required to create and maintain trace links can be extremely high [8]. Traditionally, the linkage between changes and incidents is done manually by developers, often resulting in incomplete records due to factors like time pressure for incident resolution or lack of motivation, highlighting the need for automation.

Reliable links between incidents and the changes that induced them are essential for several reasons. Firstly, identifying patterns and trend of particular changes that consistently trigger incidents can prevent future incidents [17]. Secondly, better understanding the impact of specific changes can improve the change management process, leading to more effective testing and implementation processes [5]. Thirdly, identifying high-risk changes that are more likely to trigger incidents can helps prioritise testing and risk mitigation efforts, reducing the likelihood and impact of incidents. Lastly, rapid identification of the change responsible for an incident improves incident response and mitigation [5]. A major challenge is that only a small number of incidents explicitly reference their inducing changes, making the set of problematic changes incomplete [3]. This results from the higher focus and time pressure put on quickly resolving incidents [17, 20]. Historical data linking these changes to resulting incidents is needed to assess the true risk of a change, factoring in the likelihood of it causing an incident based on similar change performance in the past [17].

Overall, studying the link between incidents and changes is crucial for effective incident management, helping organisations prevent, mitigate, and respond to incidents. This study aims to closely examine changes that trigger incidents, replicate previous methods linking changes to the induced incidents, introduce an adapted approach, and showcase its outcomes. While the methods from Güven et al. [17] and Batta et al. [3] have previously been applied at IBM, a large IT company, we are assessing their applicability in a financial services context.

Our main goal is to research how we can improve the traceability between changes and incidents using AIOps methods. To address this, we have formulated the following research questions:

- **RQ1**: What are the characteristics of changes that induce incidents?
- **RQ2**: Which AIOps methods have been successfully employed, and under what conditions?
- **RQ3**: To what extent can we adapt current AIOps methods to improve the traceability between changes and incidents?

This research is done on the change and incident management data of the software-defined business ING (International Netherlands Group), a multinational banking and financial services corporation. ING provides a wide range of financial services and is known for its digital banking services.

In our investigation on the three research questions, our findings highlight the challenges in determining links between changes and incidents. These difficulties are due to different dimensions that play a significant role in the AIOps methods used for accurately establishing links. Consequently, it becomes evident that the characteristics of incident-inducing changes can vary across different contexts. Additionally, the evaluation of the methods, based on historical data and real-world assessments with engineers from the company, revealed a significant disparity, with an increased occurrence of false positives when identifying links between unlabeled changes and incidents. This highlights the complex nature of identifying links between changes and incidents, emphasising the contextual influence on AIOps method effectiveness. This study contributes to improving the quality of current data through AIOps approaches. However, further measures are necessary to address issues such as data imbalances and promote a postmortem culture that brings attention to the importance of properly administrating tickets. These measures will result in a better overview of change failure rates, which, in turn, aids in risk compliance and reliable change management.

The remainder of this paper is organised as follows: Section 2 presents the related literature. This is followed by background on the change management process and a description of ING in Section 3. The research design is outlined in Section 4 with results in Section 5. In Section 6, we discuss the findings and give a conclusion in Section 7.

2 RELATED LITERATURE

In the past few years, a significant amount of research has emerged on applying AIOps approaches on change and incident service management processes.

Currently, there are two main approaches to enhance change reliability: one focuses on pre-change risk management [3, 16], while the other identifies problematic changes post-change through monitoring [22, 35, 38]. The lack of high-quality data is often cited as a significant challenge in applying AIOps approaches [2, 11, 29]. Various methods are used to address this challenge, such as utilising unsupervised or semi-supervised models on synthetic data based on real-world data.

Various post-change methods address the challenge of low-quality labeled data. SCWarn, for instance, identifies problematic changes using multimodal anomaly detection on heterogeneous multi-source data, employing an unsupervised model to compensate for insufficient labeled data [38]. To assess its performance, it was evaluated on two open-source benchmarks and synthetic data based on real-world data from two systems. Meanwhile, Kontrast employs self-supervised contrastive learning to identify problematic software changes, utilising synthetic data based on real-world data for evaluation [35]. Gandalf is a semi-supervised model developed for safe deployment in cloud infrastructure, validated using Microsoft Azure data [22].

Our study aims to prevent incidents pre-change by identifying if a change caused an incident through similar historical incident trace linkage. Previous work on this topic is limited due to the challenges in systematically collecting data on changes inducing incidents [17], primarily because more focus and time pressure are placed on resolving incidents than on the procedural guidelines that require proper administration [20].

Batta et al. proposed a risk management system based on past problematic changes [3], facing challenges because the data was

not usable "as is" due to the scarcity of incidents caused by changes being explicitly labelled as a link. They developed a semi-supervised learning approach to determine implicit linkages between change and incident records using explicit links, with data from IBM. Güven et al. employed a similar semi-supervised approach on IBM data to obtain implicit change-incident linkages from explicit ones [17]. A follow-up paper by Güven and Murthy explored these linkages for predictive analytics aimed at reducing change-related incidents [16].

Our study replicates models developed by Güven et al. [17] and Batta et al. [3] in a financial software-defined business. We aim to assess their performance and propose an adapted context-specific method based on our findings.

3 BACKGROUND

In this section, we provide background on the case company, their change management process, and the risk-related influences on this process.

3.1 Case Company

Our research is focused on ING, which is a large internationally operating Dutch company that provides a range of financial products and services to millions of customers. With over 15,000 developers deploying thousands of changes each month, the company has shifted from a traditional bank to a digital platform, offering internet and mobile banking services to customers. The growing importance of Information and Communications Technology (ICT) in the financial sector has made it essential for financial entities to incorporate ICT into their daily operations [27]. As financial services have become predominantly digital and heavily reliant on software, we classify ING as a financial software-defined business.

Due to their vital role in the global economy, banks like ING are subject to regulatory oversight, necessitating compliance with risk guidelines and policies. Regulatory agencies, particularly the European Banking Authority (EBA), have a significant influence on the management of ING's processes, given its European base. The EBA, an independent European Authority, aims to ensure effective and consistent prudential regulation and supervision across the banking sector in Europe [25]. In particular, the revised Payment Services Directive introduced in 2017 had a major impact on the IT processes of banks. The 2019 Guidelines on ICT and security risk management, further reinforces the importance of managing ICT risk in financial institutions.

From the start of 2023, the European Parliament has formally adopted the Digital Operational Resilience Act (DORA) [27]. This act provides a regulatory framework for digital operational resilience, ensuring technological safety, good functioning, and quick recovery from ICT breaches and incidents that needs to be implemented by the beginning of 2025. This framework enables effective and smooth provision of financial services while preserving consumer and market trust and confidence. DORA specifically targets ICT risk through rules on ICT risk-management capabilities, incident reporting, operational resilience testing, and monitoring of ICT third-party risks.

3.2 Change Management Process

Change management is the process responsible for change requests and risk management of changes [13]. ING employs the change management process as outlined in the Information Technology Infrastructure Library (ITIL) [13]. This is supplemented with agile principals [4], so multi-disciplinary teams are responsible for entire processes and value chains, end-to-end. In the company, they focus on IT changes that affect IT services, such as changes to hardware, network, middleware, software, and so on. A change affects at least one configuration item (CI) and has a specific implementation moment. The configuration management database (CMDB) keeps track of the location of all IT assets and processes, along with changes to their attributes and relationships [13].

The change management process is closely related to the incident management process [13], which manages the life cycle of all incidents. If a change is necessary to implement a workaround or resolve an incident, it must be logged as a change record and processed through the change management process. In turn, incident management is responsible for detecting and resolving incidents that may arise from unsuccessful changes.

The change management process is comprised of five stages: *1) Logging*: Registering IT change activities as a change record in the service management tooling. A detailed change description is necessary to support approvals and information sharing. *2) Assessment & Planning*: Assessing the risk of a change, focusing on the probability of failure and potential damage if the change fails. This stage includes registering dependencies with other teams, changes, and potential impacts on authorised changes. The deployment impact analysis and risk calculation are automated within the tooling. *3) Approval*: Evaluating the readiness of a change for deployment by approval groups. Squads are responsible for developing and maintaining their IT, ensuring compliance with company standards and compliance regulations. The product owner must approve any change released to production, including accepting the delivered quality and risks. Changes without approval are cancelled. *4) Coordinate Implementation*: Carrying out the actual change during the agreed window. IT changes can only be deployed to production within this window. *5) Evaluation & Closure*: Evaluating a change after implementation to ascertain if it functions as expected. If not, the rollback or remediation plan should be executed. A closure code is registered to indicate if the change record was successful, successful with problems, failed, or cancelled.

Our study focuses on the *Evaluation & Closure* stage by establishing more reliable links to any incidents potentially caused by the change. This enhances the completeness of change record administration at closure. Additionally, it aims to support the *Approval* stage by offering a more comprehensive view of similar historical changes that pose a risk.

3.3 Risk-related Influences to the Process

Since ING provides financial services, reliability and business continuity are essential. Failed changes can have a negative impact on the business, leading to service disruptions. Thus, robust change deployment management is needed to ensure compliance with governance, legal, contractual and regulatory requirements. The *Assessment & Planning* stage, receives high focus, as the risk level

is calculated during the assessment. The risk level considers several factors such as planned service disruption, start and end dates, CI outage, test plan, change plan, and remediation plan.

As per Article 9 of DORA [27], financial entities need to continuously monitor and control the security and functioning of ICT systems and tools [27]. They must also minimise the impact of ICT risks on these by designing, procuring, and implementing security policies, procedures, protocols, and tools that ensure ICT system resilience, continuity and availability. Specifically, for production, documented policies, procedures and controls based on a risk assessment approach are required [26]. These elements should be an integral part of the financial entity's overall change management process to ensure that all changes are recorded, tested, assessed, approved, implemented, and verified in a controlled manner [27]. The risk assessment should consider potential impacts on the continuity and quality of financial services [26]. Furthermore, post change implementation follow-up should be conducted to verify the successful implementation of changes without unexpected impacts or the need for remediation [27].

In line with DORA, ING has implemented several process entity controls for compliance. Of particular relevance to this study is the requirement that "metrics must be in place to measure change failure rate and the number of incidents (probably) caused by IT changes". Our research assists this entity control by ensuring that the links between changes and the induced incidents become more reliable.

4 RESEARCH DESIGN

In this section, we talk about the data and methods used to conduct our study.

4.1 Data

In this study, we analyse 91K closed change records deployed in the company its production environment for half a year (July 2022 to December 2022). These records are related to production incident records from the same period and were filtered based on priorities 1 and 2. Of these changes, 95% (86K) closed successfully, 4% (4K) were cancelled, and 1% (1K) induced incidents during implementation. Note that a change can also induce incidents after being closed successfully.

This data is merged with CMDB information to enrich each change record with CI Type and business application information. CI Type is a categorical value used to classify CIs, e.g. application, hardware, software, etc. A business application is an entity that is supported by a set of software components that support one or more business or IT processes, which is known by a specific name.

4.2 Methods

To investigate how to enhance traceability between changes and incidents using AIOps methods, we replicated two existing studies that focus on determining linkages between change and incident records. These studies were conducted by Güven et al. [17] (referred to as Replicated Method 1 or RM1), and Batta et al. [3] (referred to as Replicated Method 2 or RM2). Both studies employ a semi-supervised learning approach to determine implicit linkages between explicitly linked change and incident records.

An explicit linkage between a change and an incident means that an engineer has clearly identified and marked the change responsible for the incident. In contrast, an implicit linkage indicates that the incident record lacks a direct reference to the inducing change, but a probabilistic relationship has been determined by one of the methods.

Furthermore, we introduce a third method, which is an adaptation inspired by the aforementioned studies (referred to as Method 3 or M3), which addresses their limitations in our context. All three methods, including the replications RM1 and RM2, are implemented in Python.

4.2.1 RM1. Güven et al. [17] propose a semi-supervised approach for establishing causality between changes and incidents by determining contributing dimensions that indicate a change caused an incident.

Feature Preparation. First, we remove incident records that inaccurately report incidents by analysing the resolution messages to reduce bias from false incidents. Subsequently, data processing is performed. For changes, the structured features include "change ID", "start timestamp", "urgency", "CI type", and "assignment group". The fields "short description" and "description" are collectively referred to as the "change text". As for incident records, the structured features comprise "incident ID", "start timestamp", "urgency", "CI type", and "responsible group". The fields "short description", "description", and "solution" are collectively referred to as the "incident text".

To extract explicit links, regular expressions patterns are utilised to parse the incidents and extract mentioned changes in them. Furthermore, *(entity, action)* pairs are extracted from the change descriptions, representing the top-ranked pair of verbs associated with noun phrases. Additionally, the number of common words between the change and incident text is used as another indicator of the connection between changes and incidents. Moreover, text matching is employed on the change or incident text to identify all associated Configuration Items (CIs).

Based on these considerations, the dimensions used as potential indicators of causality between changes and incidents include [17]:

- Time: time elapsed between change and incident
- SameCI: whether the change and incident happened on the same CI
- SameType: whether the change and incident have the same CI Type
- SameGroup: whether the change assignment group matches the incident resolver group
- SameImpact: whether the change urgency matches the incident urgency
- SameEntityAction: whether the change (entity, action) pair exists in incident text
- NumberCommonWords: count of common words between change and incident text

Approach. Initially, a ground truth is established by identifying some changes that resulted in incidents and some that did not.

For changes that led to incidents, explicit mentions of changes in incident records are analysed. To enhance reliability, a temporal filter is applied to exclude pairs where the change occurred after the

incident, rendering it impossible for it to have caused the incident. Additionally, unstructured text cues that mention incidents being "caused by a change" or "due to a change" are examined for further validation.

For changes that did not lead to incidents, the following rules are applied: ensuring no incidents occur within the 14 days following the change, marking changes as successful, and identifying explicitly mentioned post-incident changes that are not mentioned elsewhere as causing an incident. The original paper used a 30-day window to check for incidents, but in compliance with the Process Control Standard of ING, we use a 14-day window, as changes not closed within 14 days after the planned end date are considered non-compliant.

The candidate dimensions are then evaluated for their contribution to the change-incident linkage. Test cases are generated by considering all incidents from the ground truth data set. For each incident, all changes within the four weeks preceding the incident for the specific business application are taken into account. A test case comprises an incident and the change that is a potential candidate for having caused this incident. The changes are then ranked based on the number of matching dimensions with the incident. To identify significant dimensions, the single-variable Kolmogorov-Smirnov statistics [23] is used to assess if there is a statistically significant difference between the distributions of changes that led to incidents and those that did not.

4.2.2 RM2. Batta et al. [3] propose a semi-supervised learning based approach to leverage the explicit linkages between change and incident records to uncover additional implicit links. This method relies on the importance of common tokens found in the text of changes and incidents.

Feature Preparation. Explicit linkages of incidents to changes can be found either in the "caused by change" or "solution" field of incidents. These fields are scanned to extract explicit mentions of change numbers, which are analysed to determine an appropriate time window to scan for implicit linkages.

Next, a set of candidate linkages is generated to identify implicit change-incident linkages by combining each change record with the incidents records within the implementation window of the same application. For each candidate linkage, we identify common tokens between the change and incident text after applying custom pre-processing. This pre-processing involves removing special characters (except "/", "-", ":", " " and ":"), eliminating date and time values along with timezones, converting all text to lowercase, removing stopwords, and applying lemmatisation. For change records, this pre-processing is applied to the "short description", "description", "change plan" and "backout plan", which are then concatenated to create the "change text". The same is carried out for incidents on the "short description", "description", "configuration item", "solution" and "caused by change", resulting in the creation of the "incident text". The candidate linkages that have at least one common token after this pre-processing are retained.

Approach. To compute the linkage strength for each candidate change-incident pair, we consider the set of common tokens between the change and incident records. The inverse document frequency (IDF) [30] is used to assign a weight to each token in the change and incident corpus separately. For a given token t, its IDF in the change corpus and incident corpus is denoted by C_t and I_t, respectively. The set of common tokens is represented as T, and the linkage strength S, can be computed using Equation 1. This computation is applied to all candidate and identified explicit linkages.

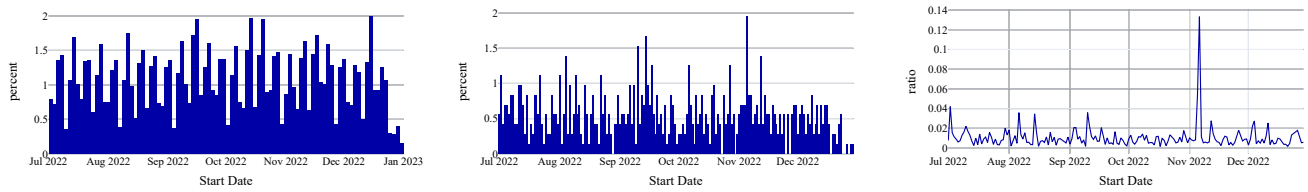
$$s = \sum_{T}^{t=1} C_t^W * I_t^W, t \in T \tag{1}$$

Afterwards, the linking strength of the explicit linkages is compared to that of the candidate linkages using an independent sample t-test to determine if there is a significant difference in means between the two groups. Subsequently, real linkages are filtered from the candidate linkages by selecting a cutoff value for linkage strength. The approach involves iterating over all the values of linkage strength in the explicit and candidate linkages to calculate the total cost and select the linkage strength with the minimum cost. To maximise the number of explicit linkages above the optimal cutoff, a cost is imposed based on the number of explicit linkages with a linkage strength below the current cutoff. Similarly, a cost is imposed based on the number of candidate linkages above the current cutoff. A higher cost is imposed for explicit linkages by a factor of η, which is the ratio number of number of candidate linkages to explicit linkages.

4.2.3 M3. Our final method, referred to as M3, draws inspiration from the previously replicated methods. It is also a semi-supervised learning based approach that leverages explicit linkages between change and incident records to uncover reliable causality between changes and incidents with implicit links. The method utilises contributing dimensions and the importance of common tokens found in the text of changes and incidents, which are supplied to a classifier.

Feature Preparation. Explicit linkages between incidents and changes are obtained by extracting information from both the "caused by change" field in incidents and by extracting change numbers from the "solution" field. Pairs where the change start date occurs after the incident creation date are filtered out to eliminate changes that resolve incidents. These linkages are thoroughly analysed to determine an appropriate window to scan for implicit linkages.

As in RM2, a set of candidate linkages is generated to identify implicit change-incident linkages by combining each change record with the incidents records within the implementation window of the same application. For each candidate linkage, we identify the common tokens between change text and incident text after applying the custom pre-processing. The text pre-processing is similar to that of RM2, with the addition of removing single digits and words consisting of less than two characters. By eliminating single integers and small tokens, we effectively eliminate low-value words and reduce processing time. This removal is necessary because, often, digits and single special characters are left over after the pre-processing of RM2. As in RM2, records are concatenated to create the "change text" and "incident text". The candidate linkages that contain at least one common token between the records are retained.



(a) (b) (c)

Figure 1: Frequency of the start date of non-incident inducing changes (a), incident inducing changes (b) and their ratio (c).

Approach. For each candidate change-incident pair, the linkage strength is computed by evaluating the IDF of the set of common tokens between the change and incident records, following the same approach as in RM2 (see Equation 1). Additionally, for each pair, the significant dimensions from RM1 are incorporated as features.

Enhancing the linkage between changes that induce incidents can be characterised as a positive and unlabeled (PU) classification problem [14]. In this context, a portion of the data is positively labeled for changes that trigger incidents, while the rest is unlabeled and may include both positive and negative instances. Following the approach of Elkan and Noto [14], we introduce variables: x representing an example and $y \in 0, 1$ as a binary label. Specifically, we set $s = 1$ when example x is labeled, and $s = 0$ when x is unlabeled. Only positive examples are labeled, so $y = 1$ is certain when $s = 1$. However, when $s = 0$, the possibility exists for either $y = 1$ or $y = 0$ to be true.

Within our data set, which comprises of both positive and unlabeled data, the probability that a certain sample is positive $[P(y = 1|x)]$ equals the probability that the sample is labeled $[P(s = 1|x)]$ divided by the probability that a positive sample is labeled in our data set $[P(s = 1|y = 1)]$.

For implementation, we employed the XGBoost classifier from scikit-learn [28], drawing inspiration from the code developed by Drouin [12]. XGBoost was selected because it is a widely used machine learning method that incorporates a theoretically justified weighted quantile sketch procedure [7]. This procedure enables the handling of instance weights in approximate tree learning. In essence, it allows the model to account for the disparity between labeled and unlabeled instances in our data.

4.3 Evaluation

To assess the methods, we generated test cases of potential change-incident pairs and conducted evaluations based on the available explicit links. The evaluations depend either on the replicated paper or conventional evaluation methods.

For RM1, we examined all incidents from the ground truth data set. For each incident, we considered the changes in a particular business application that occurred within 14 days before the incident and ranked them based on the number of matching dimensions.

For RM2, the data was split into an 80% training and 20% testing set to evaluate its performance in detecting explicit change-incident linkages. This split is time-based, ensuring that information from the training set does not leak into the test set. We selected all change-incident linkages with a linkage strength exceeding the determined cutoff as test cases, allowing us to assess the percentage that are explicitly linked in terms of precision (as defined in Equation 2) and recall (see Equation 3).

In the case of M3, we employed the classifier on a data set where 75% of the explicit links labels were removed, facilitating an assessment of its performance in terms of precision and recall.

Additionally, we conducted real-world validation of RM2 and M3 with engineers. RM2 used all candidate linkages above the cutoff as test cases, while M3 included candidate linkages classified as links but lacking explicit links in the data set. RM1 was excluded from this validation as it did not provide a means to create candidate linkages with changes that were not already linked.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

In the validation process, we focused on a specific department at ING with about 150 engineers. Test cases were evaluated by either the engineer responsible for the incident or change, or by a knowledgeable individual from the same team or department. The assessment involved rating each test case on a scale of 1 to 10 regarding the likelihood that the change caused the incident. This was followed by assessing their confidence level using the same scale and providing an explanation for their rating. These steps collectively provided insights into the model its accuracy based on the sample.

5 RESULTS

This findings of the study are showcased in this section, organised according to the research questions. They encompass the characteristics of incident-inducing changes, the successful application of AIOps methods in pinpointing these changes, and an adaptation to the existing methods with an evaluation of these methods in practice.

5.1 RQ1 - What are the characteristics of changes that induce incidents?

Change-incident linking is currently carried out during the *Evaluation & Closure* stage of the change deployment management process at ING. Engineers manually establish this link in the service management tooling once they determine that a particular change led to an incident.

In the second half of 2022, out of the 91K changes, 1K (1%) were linked to incidents they caused. Figure 1 displays the distribution of non-incident and incident inducing changes their frequency, along with their corresponding ratios. Non-incident inducing changes followed a weekly pattern, peaking in the middle of September, October and December (see Figure 1a), likely due to Microsoft's Patch Tuesday [33]. Incident-inducing changes were more irregular but still had a discernible weekly pattern (see Figure 1b), with a November spike attributed to a disaster recovery exercise upon closer examination of the incidents their description. A disaster recovery exercise is a simulation of a major incident to test the resilience of operations [9]. The overall ratio of both change types was about 1% (see Figure 1c). Notably, there was a massive 13% spike on November 6th, also linked to a disaster recovery exercise. This suggests more changes follow patch releases. Also, patches and disaster recovery exercises lead to a higher occurrence of changes inducing incidents compared to normal circumstances.

The ratio of both change types per day of the week is depicted in Figure 2. We see that the highest incidence of changes causing incidents is on Saturdays (1.7%), with the lowest on Thursdays (0.6%). This pattern is likely due to system improvements done outside of regular business hours [5], which typically fall on weekends.

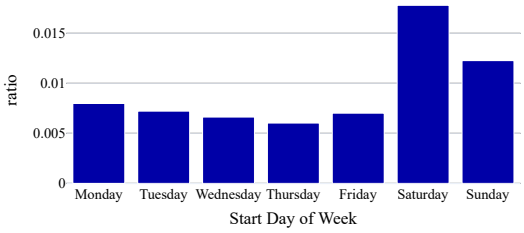


Figure 2: Ratio of non-incident inducing changes and incident inducing changes per Day of the Week.

Next, we explore the priority of incidents, both linked or not linked to changes that caused them (see Figure 3). Incidents with a priority of 1 are considered critical, with the highest level of impact and urgency, and those with a priority of 2 are deemed high, with a medium and high level of impact and urgency. Incidents caused by changes have a higher proportion of critical incidents, with 11.7% being priority 1, compared to 6.1% for incidents not caused by changes.

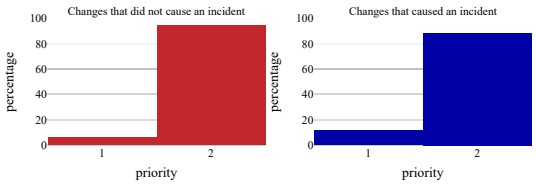


Figure 3: Distribution of the priority of incidents.

In our analysis of incident-inducing changes, we explored their risk category. The service management tooling uses specific criteria to calculate a risk category, considering factors of the probability of failure (scope of impacted IT services, deployment complexity, and incident history) and of potential damage (Confidentiality, Integrity, and Availability rating, SOx criticality [34], and recoverability). There are three categories: low, medium and high risk. In Figure 4, the risk category analysis shows that changes with medium risk levels remain consistent, whether or not they induce incidents. However, incident-inducing changes tend to have slightly higher risk levels and fewer instances with lower risk levels, with 10.2% falling into higher-risk levels compared to 2.4% for changes that didn't cause incidents.

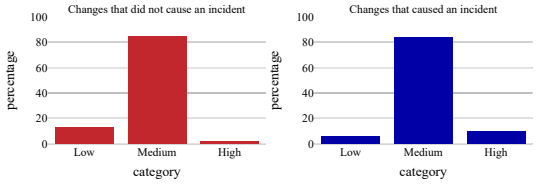


Figure 4: Distribution of the calculated risk category of changes.

The result from the risk category analysis implies that incident-inducing changes are linked to higher risk levels based on the current calculation method.

5.2 RQ2 - Which AIOps methods have been successfully employed, and under what conditions?

In Section 4.2, we introduced the semi-supervised approaches of Güven et al. [17] (RM1), and Batta et al. [3] (RM2), originally implemented at IBM. In this study, we seek to evaluate the generalisability of these models to our specific context, a financial services company.

5.2.1 RM1. Güven et al. [17] particularly focused on the time factor in establishing causality between changes and incidents. They suggested that it's reasonable to assume that the change leading to an incident occurred shortly before the incident. However, their analysis of 100 test cases found that only 3% of incidents happened within an hour of the change, and in just 11% of cases, the change and incident occurred on the same day. Additionally, when incidents occurred on the same day, the change closest to the incident was almost always the one causing it.

In contrast, in our replication, for all 153 pairs, the incident occurred within an hour and on the same day as the change implementation 100% of the time. However, among these pairs, the change closest in time to the incident was the actual cause only 9.8% of the time. This indicates that at ING, the change inducing an incident indeed tends to be implemented shortly before, but the closest change is not always responsible for the incident.

To test the hypothesis that the closest change to the incident is responsible, we collected test cases of change-incident pairs. We

ranked preceding changes by their proximity to the incident, assigning confidence scores. We then ranked these combinations and assessed the accuracy against the ground truth, as shown in Table 1. The original approach found the correct change causing an incident in the top 5 results 52% of the time, with the top-ranking change being the cause in 30% of cases. In our replication, these percentages were 35.3% and 9.8%, respectively. This suggests that relying solely on time is insufficient to establish a connection between the incident-inducing change with RM1.

Table 1: Accuracy of using time and all dimensions for change-incident pairs of the original and replication.

Original			Replication		
Dimension	top 5	top 1	Dimension	top 5	top 1
Time	52%	30%	Time	35.3%	9.8%
All	67%	51%	All	45.1%	11.8%
Significant	75%	58%	Significant	56.9%	22.2%

Additionally, we considered all defined dimensions (see Section 4.2.1). In each test case, we ranked the preceding changes based on the number of matching dimensions with the incident and compared them to the ground truth. The accuracy for both the original and our replication is also shown in Table 1. In the original approach, the correct change causing an incident ranked in the top 5 results 67% of the time, with the top-ranking change being correct in 51% of cases. In our replication, these percentages were 45.1% and 11.8%, respectively.

Including all dimensions improved results over relying solely on time. To identify significant dimensions, we used the single-variable Kolmogorov-Smirnov statistics to assess differences between change distributions leading to incidents and those that did not. In the original method, only *OwnerGroup* and *NumberCommonWords* were significant. In our replication, *SameType*, *SameImpact*, *OwnerGroup* and *NumberCommonWords* were significant with a p-value below 0.05. When using these significant dimensions and retesting, the accuracy for correctly identifying the incident-inducing change is shown in Table 1. In the original method, the correct change ranking in the top5 results was 75%, with the top-ranking change being correct 58% of the time. In our replication, these figures were 56.9% and 22.2%, respectively.

Focusing solely on significant dimensions greatly explains which features influence the ranking of the incident-inducing change. However, in the case of ING, where changes happen rapidly, identifying the closest change to an incident, as in the original, does not accurately explain their relationship. Our ING data suggests that additional dimensions, like type and impact, contribute to understanding the link between changes and incidents. In the original, a few dimensions could explain the relationship for around 6 out of 10 changes related to incidents. In the context of ING, additional information is needed to accurately establish the link, as only about 1 out of 5 changes could be linked using several dimensions.

5.2.2 RM2. Batta et al. [3] start by preparing the features, which includes identifying the explicit linkages and generating candidate change-incident pairs.

In the original, they searched for change numbers in the "caused by change" and "solution" fields of incidents, finding 58 change records. They observed that 93% of these incidents occurred within 15 days of the change implementation, leading them to use a 15-day window for scanning implicit linkages. In our replication, we found 4K change records that induced incidents, but about 700 were deemed invalid as they were linked as solutions, not causes. After removing the invalid links, 94% of these incidents occurred within three days of the change implementation, prompting us to use a three-day window for scanning implicit linkages. This highlights that at ING, linkages are typically closer in time compared to those at IBM, similar to findings of RM1.

After generating candidate change-incident pairs with at least one common token shared between records, the original study found 692 candidate linkages, potentially containing some unmarked real change-incident linkages. In our replication, we identified 81K candidate linkages, which may also contain unmarked real change-incident linkages.

In the original, the independent sample t-test on the linkage strength of explicit and candidate linkages resulted in a p-value < 0.001. Explicit linkages had a mean linkage strength of 108, while candidate linkages had a mean of 34. This led to the rejection of the null hypothesis at a significance level of $\alpha = 0.05$, indicating their significant difference. In our replication, the independent sample t-test also gave a p-value < 0.001, but with mean values of 142 for explicit linkages and 29 for candidate linkages. Notably, the mean value for candidate linkages remained around 30 in both the original and replication, while explicit linkages had a much higher mean linkage strength, exceeding 100 in both cases.

The cutoff value for linkage strength, determined through the optimisation function in the original research, was 65. This resulted in 61 implicit linkages (out of 692 candidates) being added to the set of problematic changes. In our replication, the cutoff value was 59, adding 13K implicit linkages (out of 81K candidates) to the set of problematic changes.

The data was split into an 80% training and 20% testing set based on the time of the change to evaluate the performance of the method in detecting explicit change-incident linkages. The method achieved a precision of 11.62% and a recall of 57.59%. Since recall exceeded precision, the method retrieved more relevant results but also generated some irrelevant ones. This indicates that relying solely on the number of common words may not sufficiently explain the relationship between incident-inducing changes.

5.3 RQ3 - To what extent can we adapt the current AIOps methods to improve the traceability between changes and incidents?

In this section, we present the results of the adapted method M3, and the results of the RM2 and M3 method in practice.

5.3.1 Results of M3. The M3 method builds upon the semi-supervised learning approaches replicated in RQ2, which also utilises explicit change-incident linkages to uncover implicit connections. To enhance this method, we incorporate significant dimensions from RM1, generate the linkage strength from RM2 and supply them to a positive and unlabeled (PU) classifier.

This method also starts with feature preparation by identifying explicit linkages and generating candidate change-incident pairs. We also remove single integers and small tokens from the text to filter out words with low linkage strength value and to speed up subsequent processing steps.

Based on the explicit mention of changes found in the "caused by change" and "solution" fields of incidents, we identified 11K change records. We ensure to remove explicit linkages where changes occur after incidents, improving data reliability by removing incorrect labels. After also filtering changes not linked to a business application, we had 4K records linked to incidents. Notably, 97% of these incidents occurred within three days, leading to a three-day window, similar to RM2.

After generating candidate pairs with at least one common token, we found 81K candidate linkages. These candidates could potentially contain real change-incident linkages not explicitly marked in the data.

The independent sample t-test on linkage strength showed a significant difference between explicit and candidate linkages ($p < 0.001$). Explicit linkages had a mean strength of 91, while candidate linkages had a mean strength of 28, confirming the difference.

The cutoff value for linkage strength, determined by the optimisation function, is 57. This technique added 12K implicit linkages (out of 81K candidates) to the set of problematic changes.

For all linkages, we included the dimensions *SameType*, *SameImpact*, *OwnerGroup* and *NumberCommonWords* which were identified as significant in Section 5.2.1 for RM1. We removed 75% of explicit link labels to assess the PU classifier its performance. Results include a precision of 52.15% and a recall of 47.98%, indicating that about half of the identified links are indeed explicit. Compared to RM2 with a precision of 11.62% and recall of 57.59%, this adapted method inspired by Güven et al. [17], and Batta et al. [3] yields better precision. Therefore, by incorporating additional features and leveraging PU classification, traceability between changes and incidents can be substantially improved.

5.3.2 Performance in Practice. As indicated in Section 4.3, only RM2 and M3 underwent validation by engineers on test cases from one particular department. RM2 had 37 test cases evaluated by six engineers, while M3 had 12 test cases evaluated by three engineers. Notably, eight test cases were selected by both methods.

Validation results, shown in Table 2, asked engineers to rate the likelihood of a change causing an incident on a scale of 1-10. In most cases, both methods indicated no link, and two cases in each method receiving a rating of 5, signifying uncertainty. RM2 accurately identified nine links, achieving a 25% accuracy. Conversely, M3 did not identify any accurate links among these test cases.

Engineers expressed their confidence levels in their ratings, ranging from 6 to 10 on a scale from 1 to 10, as depicted in Figure 5. In addition to providing ratings, the engineers explained the reasons behind their choices. Often, factors such as timing, configuration items, short descriptions, and descriptions of the change and incident were taken into account.

Most incorrect links were due to unrelated parts of an application being erroneously connected. For example, an incident related to machine access issues were linked to a change involving Kafka producers, responsible for writing events to a cluster. Some cases were linked where the cause of the incident are external factors impacting the application rather than issues within the business application. In RM2, the nine correctly linked cases were attributed to routine events that followed a network routing or disaster recovery change.

Table 2: Link likelihood results of RM2 and M3 in practice.

	RM2		M3	
Rating	Count	Proportion	Count	Proportion
1	21	57%	10	83%
2	4	11%	0	0%
3	1	3%	0	0%
4	0	0%	0	0%
5	2	5%	2	17%
6	0	0%	0	0%
7	0	0%	0	0%
8	0	0%	0	0%
9	1	3%	0	0%
10	8	22%	0	0%

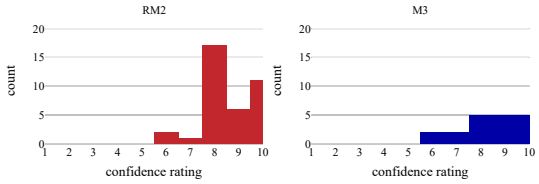


Figure 5: Confidence of Link Likelihood Results.

In practice, AIOps methods yield different results in practice at ING compared to historical data testing. While it is possible to find links between changes causing incidents that were not explicitly marked, it is also common to encounter more false positives.

6 DISCUSSION

In this section, we discuss the implications of our results and go into the threats to validity of our findings.

6.1 Implications

We see the following implications of our results for using AIOps methods to improve the traceability between changes and incidents.

6.1.1 Implications for Researchers. Firstly, as demonstrated in Section 5.2, the replicated results on ING data often exhibited significant discrepancies from those reported in the original papers. For example, both methods indicated that explicit linkages between changes and incidents are typically closer in time than those at IBM, necessitating the use of different time windows. Additionally, in RM2, an additional step was required to filter out invalid, explicitly linked incident-inducing changes. This supports the fact that simply copying a method from a paper or company and applying it to another context does not guarantee the same results. It often becomes imperative to consider the specific characteristics of each context

for successful implementation. This reaffirms the well-known best practices in software engineering of the importance of tailoring methods to the organisational context [32].

Moreover, as evident in Section 5.3.2, the evaluation of both methods with engineers in practice yields unsatisfactory results, particularly with M3 performing notably poorly. There are several potential factors contributing to these outcomes. One key issue is the substantial data imbalance between labeled and unlabeled instances. To illustrate, less than 1% of changes are accurately labeled, while candidate linkages are generated by combining each change record with all incidents records within the determined window, resulting in a significantly larger data set. Imbalanced class distribution is a well-known challenge in AI problems and can lead to an increased number of false positives in predictions [19]. XGBoost was chosen because it is reported to be capable of handling instance weights [7]. However, for future research, exploring alternative sampling techniques or cost-modifying approaches could be valuable in assessing their potential to enhance the performance of the current AIOps methods. Additionally, exploring alternative machine learning methods that might be better suited for this scenario is a worthwhile avenue to consider.

6.1.2 Implications for Practitioners. Section 5.3.2 reveals that during the evaluation in practice, most incorrect links were a result of erroneously connecting unrelated parts of a business application. Currently, candidate linkages are established based on the scope of business applications, which may not align as effectively with the context of ING as it does with IBM. Therefore, exploring relationships among various parts of an application could enhance the traceability of changes and incidents. In future research, different scopes, such as customer journeys or trace relations, could be considered for the execution of the method to assess their impact on performance.

Furthermore, since the labelling of incident-inducing changes is currently a manual task for engineers, the quality of the data may be incomplete. This can be attributed to various factors, such as time pressure for incident resolution or a lack of motivation [8]. Sandkuhl emphasises that successful implementation of AI approaches requires suitable and high-quality training data, meaning incomplete records have to be removed [31]. In this context, we are actively working on improving these incomplete records with AIOps approaches. However, the current state of data quality may not be sufficient.

Enhancing the quality of records involves recognising the value of learning from past outages, which contributes to preventing future incidents. This is achieved by fostering a postmortem culture [5]. A postmortem is a comprehensive record of an incident, encompassing its impact, the actions taken to mitigate it, the root cause(s), and the follow-up actions to prevent a recurrence [5]. While it provides a structured process for learning from previous incidents, it does entail a cost in terms of time and effort. An essential aspect of a successful postmortem culture is being blameless, focusing on identifying the contributing causes without indicting any individual or team for bad or inappropriate behaviour. Such a culture ensures that people feel comfortable bringing issues to light without fearing punishment. For a collaborative postmortem culture to succeed, it necessitates active participation and endorsement from all levels of an organisation [5]. This helps mitigate the challenge of engineers not recognising the value of properly administrating tickets [20]. Therefore, it is recommended that organisations seeking to effectively employ AIOps methods reinforce a robust postmortem culture across all levels of their hierarchy.

Additionally, the incomplete manual linking can be attributed to a greater emphasis on incident resolution rather than on procedural guidelines within the incident management process [20]. Similar pressures have been identified in other fields, such as aeronautics, where the focus on maintaining business continuity can affect other aspects like the change control process [24]. Accurately tracking incident-specific details can be managed in two ways: through manual checks or automated trace collection. As previously mentioned, manual checks come with significant costs in terms of time or effort. Therefore, implementing an automated tracker that collects comprehensive and relevant information, including monitoring metrics and communication, can enhance reliability over time [5]. This approach is recommended to ensure the quality of future generated data, which also contributes to making the change failure rate more reliable. This, in turn, assists the risk compliance of the company's process entity control based on DORA [27].

6.2 Threats to Validity

The threats and limitations of the study are categorised into construct validity, external validity and reliability [37].

1) Construct validity is ensured by evaluating the AIOps methods on both historical data and engineer evaluations as multiple sources of evidence. Also, a confidence value is given to each engineer evaluation to give weight to how accurately they value their evaluation.

2) External validity is established by replicating the methods from Güven et al. [17] and Batta et al. [3] performed on IBM data, and presenting an adapted method for ING. As the results show, replicating methods performed on other company data and applying them to another does not guarantee the same results. Context of the data and process needs to be taken into account, like the timing and relationships of parts in a business application, in order to better link changes to their induced incidents. Other financial software-defined businesses may be closer in context to obtain more generalisable results.

3) Reliability is demonstrated by replicating method RM1 and RM2 as described in their respective papers, and describing the needed adjustments to fit our context in Section 4. The results of the engineer validation heavily relies on the chosen sample of business department. Engineers directly related to either the change or incident were targeted as much as possible, to ensure that expert knowledge was used when evaluating the likelihood of a link.

7 CONCLUSION

In conclusion, this study analyses incident-inducing changes and the application of AIOps methods to detect and understand their relationships in practice.

Our findings indicate the complexity of determining links between changes and incidents, due to differing requirements for

dimensions used in methods to accurately establish them. This suggests that the characteristics of incident-inducing changes can vary across different contexts. During practical validation with engineers, a big difference is apparent between historical-based results and real-world evaluations, indicating the significant influence of context on the effectiveness of AIOps methods.

This study highlights the complexities involved in determining incident-inducing changes, emphasising the necessity for context-specific methods, handling of data imbalance issues, improved data quality, and fostering a postmortem culture. The study provides valuable insights and directions for future research to enhance the traceability of changes and incidents. This aids in providing a more comprehensive overview of change failure rates, thereby facilitating improved risk compliance and reliable change management.

ACKNOWLEDGMENTS

This work was partially supported by ING through the AI for Fintech Research Lab with Delft University of Technology.

REFERENCES

[1] Rainer Alt, Jan Marco Leimeister, Thomas Priemuth, Stephan Sachse, Nils Urbach, and Nico Wunderlich. 2020. Software-Defined Business. *Business & Information Systems Engineering* 62, 6 (2020), 609–621.

[2] Lucas Baier, Fabian Jöhren, and Stefan Seebacher. 2019. Challenges in the Deployment and Operation of Machine Learning in Practice. In *ECIS 2019 proceedings. 27th European Conference on Information Systems (ECIS), Stockholm & Uppsala, Sweden, June 8-14, 2019. Research Papers*. 163.

[3] Raghav Batta, Larisa Shwartz, Michael Nidd, Amar Prakash Azad, and Harshit Kumar. 2021. A system for proactive risk assessment of application changes in cloud operations. In *2021 IEEE 14th International Conference on Cloud Computing (CLOUD)*. IEEE, 112–123.

[4] Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, et al. 2001. The agile manifesto.

[5] Betsy Beyer, Chris Jones, Jennifer Petoff, and Niall Richard Murphy. 2016. *Site reliability engineering: How Google runs production systems*. " O'Reilly Media, Inc.".

[6] Junjie Chen, Shu Zhang, Xiaoting He, Qingwei Lin, Hongyu Zhang, Dan Hao, Yu Kang, Feng Gao, Zhangwei Xu, Yingnong Dang, et al. 2020. How incidental are the incidents? characterizing and prioritizing incidents for large-scale online service systems. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*. 373–384.

[7] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[8] Jane Cleland-Huang, Orlena CZ Gotel, Jane Huffman Hayes, Patrick Mäder, and Andrea Zisman. 2014. Software traceability: trends and future directions. In *Future of software engineering proceedings*. 55–69.

[9] IBM Cloud. 2023. IBM Cloud Docs. https://cloud.ibm.com/docs/overview?topic=overview-dr-testing

[10] CoEST. [n. d.]. Center of excellence for software & systems traceability. http://sarec.nd.edu/coest/index.html

[11] Yingnong Dang, Qingwei Lin, and Peng Huang. 2019. AIOps: real-world challenges and research innovations. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. IEEE, 4–5.

[12] Alexandre Drouin. 2013. PUAdapter: A tool that adapts any estimator that can output a probability to positive-unlabled learning. It is based on: Elkan, Charles, and Keith Noto. "Learning classifiers from only positive and unlabeled data." Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008. https://github.com/aldro61/pu-learning.

[13] IBM Cloud Education. 2019. IT Infrastructure Library (ITIL). https://www.ibm.com/cloud/learn/it-infrastructure-library

[14] Charles Elkan and Keith Noto. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 213–220.

[15] Nicole Forsgren, Jez Humble, and Gene Kim. 2018. *Accelerate*. IT Revolution, Portland, OR.

[16] Sinem Güven and Karin Murthy. 2016. Understanding the role of change in incident prevention. In *2016 12th International Conference on Network and Service Management (CNSM)*. IEEE, 268–271.

[17] Sinem Güven, Karin Murthy, Larisa Shwartz, and Amit Paradkar. 2016. Towards establishing causality between Change and Incident. In *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 937–942.

[18] ISO/IEC/IEEE. 2017. ISO/IEC/IEEE International Standard - Systems and software engineering–Vocabulary. *ISO/IEC/IEEE 24765:2017(E)* (2017), 1–541. https://doi.org/10.1109/IEEESTD.2017.8016712

[19] Nathalie Japkowicz and Shaju Stephen. 2002. The class imbalance problem: A systematic study. *Intelligent data analysis* 6, 5 (2002), 429–449.

[20] Eileen Kapel, Luís Cruz, Diomidis Spinellis, and Arie van Deursen. 2023. Incident Management in a Software-Defined Business: A Case Study. *Available at SSRN 4333515* (2023).

[21] Andrew Lerner. 2017. AIOps Platforms. https://blogs.gartner.com/andrew-lerner/2017/08/09/aiops-platforms/

[22] Ze Li, Qian Cheng, Ken Hsieh, Yingnong Dang, Peng Huang, Pankaj Singh, Xinsheng Yang, Qingwei Lin, Youjiang Wu, Sebastien Levy, et al. 2020. Gandalf: An Intelligent,{End-To-End} Analytics Service for Safe Deployment in {Large-Scale} Cloud Infrastructure. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. 389–402.

[23] Frank J Massey Jr. 1951. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association* 46, 253 (1951), 68–78.

[24] Presidential Commission on the Space Shuttle Challenger Accident. 1986. Report of the Presidential Commission on the Space Shuttle Challenger Accident. http://history.nasa.gov/rogersrep/genindex.htm

[25] European Parliament and Council of the European Union. 2018. https://www.eba.europa.eu/about-us

[26] European Parliament and Council of the European Union. 2019-11-29. Directive (EU) EBA/GL/2019/04 of the European Parliament and of the Council of 29 November 2019 on EBA Guidelines on ICT and security risk management, repealing Directive EBA/GL/2017/17. *OJ* (2019-11-29). https://www.eba.europa.eu/regulation-and-policy/internal-governance/guidelines-on-ict-and-security-risk-management

[27] European Parliament and Council of the European Union. 2022-12-27. Regulation (EU) 2022/2554 of the European Parliament and of the Council of 14 December 2022 on digital operational resilience for the financial sector and amending Regulations (EC) No 1060/2009, (EU) No 648/2012, (EU) No 600/2014, (EU) No 909/2014 and (EU) 2016/1011. *OJ* L 333 (2022-12-27), 1–79. https://eur-lex.europa.eu/eli/reg/2022/2554/oj

[28] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.

[29] Laxmi Rijal, Ricardo Colomo-Palacios, and Mary Sánchez-Gordón. 2022. AIOps: A Multivocal Literature Review. *Artificial Intelligence for Cloud and Edge Computing* (2022), 31–50.

[30] Claude Sammut and Geoffrey I. Webb (Eds.). 2010. *TF–IDF*. Springer US, Boston, MA, 986–987. https://doi.org/10.1007/978-0-387-30164-8_832

[31] Kurt Sandkuhl. 2019. Putting AI into context-method support for the introduction of artificial intelligence into organizations. In *2019 IEEE 21st Conference on Business Informatics (CBI)*, Vol. 1. IEEE, 157–164.

[32] Alex Serban, Koen van der Blom, Holger Hoos, and Joost Visser. 2020. Adoption and effects of software engineering best practices in machine learning. In *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 1–12.

[33] Meghan Stewart and Angela Fleischmann. 2023. Update release cycle for windows clients - windows deployment. https://learn.microsoft.com/en-us/windows/deployment/update/release-cycle

[34] United States Code. 2002. Sarbanes-Oxley Act of 2002, PL 107-204, 116 Stat 745. Codified in Sections 11, 15, 18, 28, and 29 USC.

[35] Xuanrun Wang, Kanglin Yin, Qianyu Ouyang, Xidao Wen, Shenglin Zhang, Wenchi Zhang, Li Cao, Jiuxue Han, Xing Jin, and Dan Pei. 2022. Identifying Erroneous Software Changes through Self-Supervised Contrastive Learning on Time Series Data. In *2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 366–377.

[36] Yaohui Wang, Guozheng Li, Zijian Wang, Yu Kang, Yangfan Zhou, Hongyu Zhang, Feng Gao, Jeffrey Sun, Li Yang, Pochian Lee, et al. 2021. Fast Outage Analysis of Large-scale Production Clouds with Service Correlation Mining. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 885–896.

[37] Robert K Yin. 2009. *Case study research: Design and methods*. Vol. 5. sage.

[38] Nengwen Zhao, Junjie Chen, Zhaoyang Yu, Honglin Wang, Jiesong Li, Bin Qiu, Hongyu Xu, Wenchi Zhang, Kaixin Sui, and Dan Pei. 2021. Identifying bad software changes via multimodal anomaly detection for online service systems. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 527–539.

Received 6 October 2023; accepted 10 January 2024