TU Delft

Delft University of Technology

Document Version
Final published version

Licence
CC BY

Citation (APA)
Kapel, E., Lennartz, J., Cruz, L., Spinellis, D., & Van Deursen, A. (2026). Learning from Change: Predictive Models for Incident Prevention in a Regulated IT Environment. In *Proceedings - 2026 IEEE/ACM 48th International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2026* (pp. 328-339). Association for Computing Machinery (ACM). https://doi.org/10.1145/3786583.3786876

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

Learning from Change: Predictive Models for Incident Prevention in a Regulated IT Environment

Eileen Kapel
ING Bank
Amsterdam, Netherlands
eileen.kapel@ing.com

Jan Lennartz
ING Bank
Brussels, Belgium
Jan.Lennartz@ing.com

Luis Cruz
Delft University of Technology
Delft, Netherlands
l.cruz@tudelft.nl

Diomidis Spinellis
Delft University of Technology
Delft, Netherlands
d.spinellis@tudelft.nl

Arie van Deursen
Delft University of Technology
Delft, Netherlands
arie.vandeursen@tudelft.nl

Abstract

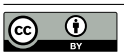
Effective IT change management is important for businesses that depend on software and services, particularly in highly regulated sectors such as finance, where operational reliability, auditability, and explainability are essential. A significant portion of IT incidents are caused by changes, making it important to identify high-risk changes before deployment. This study presents a predictive incident risk scoring approach at a large international bank. The approach supports engineers during the assessment and planning phases of change deployments by predicting the potential of inducing incidents. To satisfy regulatory constraints, we built the model with auditability and explainability in mind, applying SHAP values to provide feature-level insights and ensure decisions are traceable and transparent. Using a one-year real-world dataset, we compare the existing rule-based process with three machine learning models: HGBC, LightGBM, and XGBoost. LightGBM achieved the best performance, particularly when enriched with aggregated team metrics that capture organisational context. Our results show that data-driven, interpretable models can outperform rule-based approaches while meeting compliance needs, enabling proactive risk mitigation and more reliable IT operations.

CCS Concepts

• **Software and its engineering** → **Software reliability**; **Software post-development issues**; • **Computing methodologies** → **Natural language processing**; **Supervised learning by classification**; **Classification and regression trees**; • **Social and professional topics** → Governmental regulations.

Keywords

change management, incident management, reliability, predictive models, classification



This work is licensed under a Creative Commons Attribution 4.0 International License.
ICSE-SEIP '26, Rio de Janeiro, Brazil
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2426-8/26/04
https://doi.org/10.1145/3786583.3786876

ACM Reference Format:
Eileen Kapel, Jan Lennartz, Luis Cruz, Diomidis Spinellis, and Arie van Deursen. 2026. Learning from Change: Predictive Models for Incident Prevention in a Regulated IT Environment. In *2026 IEEE/ACM 48th International Conference on Software Engineering (ICSE-SEIP '26), April 12–18, 2026, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3786583.3786876

1 Introduction

At present, major industries such as finance, healthcare, and retail are increasingly reliant on software and related services. However, the reliability and availability of these services are often compromised by *incidents*: unplanned interruptions to a service or reductions in service quality [19]. A significant portion of incidents stem from IT changes, which involve additions, modifications, or deletions to existing IT applications [19]. As reported by Google [7], roughly 70% of outages in its live systems are due to such changes.

To monitor these change-induced incidents, we often refer to the change failure rate, which measures the percentage of deployments causing a failure in production [14]. It is important to ensure service reliability and availability, because ineffective incident management can lead to customer dissatisfaction, financial losses, and reputational damage. For example, infrastructure failures may cost $100,000 per hour, while critical application failures can reach $1 million per hour [13]. To mitigate these risks and prevent severe business disruptions, financial institutions in Europe are required to follow rigorous change and incident management processes, ensuring that all changes are carefully assessed and auditable [1, 4].

Identifying potential incident-inducing changes before deployment can help prevent incidents and ensure service reliability [15]. Engineers prefer proactive measures [21], such as enhancing testing and issue detection early in the development process, a practice known as "Shift Left". This approach improves service reliability by detecting incident-causing bugs before they reach production [15]. Incident prevention entails actively identifying potential failures and forecasting severe outages by using historical data and analytics to predict and mitigate risks [36]. Predictive incident management is enabled by artificial intelligence for IT Operations (AIOps), which uses machine learning (ML) and big data mining to forecast potential system malfunctions by analysing historical patterns [36].

Ensuring trust, involving humans in the decision-making process, and providing interpretability and explainability of AIOps

solutions is essential for gaining confidence from industry practitioners [36]. Interpretable models are often favoured over black-box models, even when slightly less accurate, as they offer great transparency and support informed decision-making.

Deploying predictive incident management in a highly regulated financial environment introduces unique challenges. Change and incident management processes in financial institutions are tightly governed by regulatory standards that demand compliance, auditability, and traceability [1, 4, 10]. These constraints limit the use of black-box models like deep neural networks or large language models [38], which often lack the transparency needed for both regulatory approval and practitioner trust.

In this work, we address these challenges by developing an approach that predicts the probability that an IT change will cause incidents. We focus on boosted tree-based classifiers (HGBC, LightGBM, and XGBoost), which are proven in practice [32], well-suited for tabular IT management data and support post-hoc interpretability through established methods such as SHapley Additive exPlanations (SHAP). Our approach generates an incident prediction score for each planned change, enabling engineers to assess deployment readiness. SHAP-based explanations provide feature-level transparency that supports user trust, informed decision-making and meets financial-sector audit requirements. Furthermore, we enrich the dataset with aggregated metrics that reflect team performance, such as change success rates, incident counts, and release outcomes, to evaluate their impact on model performance.

A key design consideration is the trade-off between accuracy and explainability. Regulatory constraints make high-performing but opaque models unsuitable, as change-approval decisions must remain traceable and auditable. The resulting prediction scores produced by the models are intended as risk signals, helping teams prioritise which changes merit closer review.

We evaluate our approach using a one-year dataset from our case company ING (International Netherlands Group), a multinational banking and financial services corporation. Building on insights from our prior work [22], which analysed the characteristics of incident-inducing changes, we tailor our solution to the organisational and regulatory requirements of financial institutions. This ensures the model is not only accurate, but also usable, trustworthy, and compliant for engineers making deployment decisions – aspects that are essential in our context but have received little attention in previous work.

Our main goal is to equip engineers with a predictive and explainable model that complies with regulatory standards, allowing them to identify high-risk changes during the assessment & planning phase and take preventive action to ensure reliable deployments. In particular, we seek to improve upon methods based on business rules and human effort, as currently employed at ING, by training ML models on historical data. We also investigate how team performance metrics can contribute to assessing incident prediction.

We address the following research questions:

- **RQ1**: What is, in practice, the performance of a rule-based approach for obtaining the incident prediction score for deployment changes?
- **RQ2**: How can we use data-driven ML models for incident prediction scoring, and how does their performance compare to the rule-based approach?
- **RQ3**: What is the effect of including aggregated team metric data on the accuracy of ML-based models for incident prediction scoring?

In our study, we demonstrate how data-driven ML methods can be employed within a tightly regulated change management process, providing more effective risk assessment than the rule-based approaches currently used in practice. Among the evaluated models, LightGBM delivers the best performance, particularly when enriched with aggregated team metrics. The model's predictions remain explainable, with influential factors including team metadata, the machine involved, and other product-specific risk indicators. By combining predictive performance with transparency, this approach has the potential to strengthen IT system reliability while reducing the time and resources spent on incident management. To our knowledge, this is the first study to emphasise both feature-level explainability and the integration of aggregated team metrics for assessing change deployment risk in a financial context.

To sum up, this work contributes the following.

(1) A comparative evaluation of a rule-based approach and three ML classifiers (HGBC, LightGBM, and XGBoost) for predicting incident risk of IT changes, using a one-year real-world dataset from a large bank. LightGBM achieves the highest weighted recall and F2-measure.
(2) A method for achieving enhanced model transparency by applying SHAP values to interpret predictions. This shows that textual change descriptions and team-related metadata are among the most influential features, providing actionable insights that support compliance and informed decision-making.
(3) An analysis of model performance stability over time, demonstrating consistent weighted F2-measure over time.
(4) Evidence that incorporating aggregated team metrics yields modest but meaningful performance improvements, especially in AUC. This underscores the potential of enriching predictive models with other data sources to improve accuracy.

2 Related Literature

Predictive models face several significant challenges, such as the lack of ground truth labels, the necessity for manual effort to obtain high-quality data, highly imbalanced datasets, and complex dependencies among components and services [36]. Despite this, substantial progress has been made in developing predictive AIOps models, particularly for incident detection and prediction. This paper examines the impact of changes on incidents, categorised into two main areas: 1) pre-change risk analysis and incident prevention [5, 16]; and 2) post-change identification of failed changes [24, 42, 44–46]. Our focus is on pre-change incident identification.

Pre-change incident identification involves predicting whether a change will result in an incident before it is deployed, based on similar historical changes that induced incidents. Changes are a common cause of incidents in live systems, responsible for up to

70% of outages [7]. Large-scale software companies have many services and resources consisting of numerous components [8], which obscure a complete view of the entire system and its relationships. These dependencies make it difficult to predict when a change will induce an incident, especially since seemingly successful changes often lead to incidents [16, 45].

Previous work on this topic is limited due to the challenges in systematically collecting data on changes inducing incidents [17]. More focus and time pressure are placed on resolving incidents than on the procedural guidelines that require proper administration [21]. Previous work at ING [22] indicates the complexity of determining links between changes and incidents, emphasising the necessity of handling data imbalance issues when utilising ML on this particular data.

On the research side, Batta et al. proposed a risk management system based on supporting evidence from past bad changes [5]. They trained multiple classification models, including logistic regression, random forest, passive aggressive, support vector machine, and LSTMs (Long Short-Term Memory models), to distinguish problematic changes from successful ones, evaluated on IBM data. Similarly, Güven and Murthy analysed IBM data to identify change-incident linkages and conducted predictive analytics to reduce change-related incidents [16]. They applied various machine learning algorithms to the properties of a change, finding that classification and regression trees achieved the highest recall.

Ahmed et al. [3] demonstrated that XGBoost is highly effective for predicting IT incident severity, often achieving performance comparable to advanced deep learning models such as BERT, RoBERTa, or ERNIE 2.0. However, their study focused exclusively on major incidents and analysed incidents after they had occurred. In contrast, our work targets high-priority incidents, including both priority 1 and priority 2 events, and predicts the risk of a change before an incident occurs. Furthermore, we extend the feature space with aggregated team metrics, providing richer context. This enables a predictive and interpretable model that leverages the proven effectiveness of tree-based classifier models while aligning with regulatory requirements for pre-deployment risk assessment in financial IT operations.

Looking at industry solutions, ServiceNow provides a cloud-based platform that aids in digitising any business process, with offerings that include IT service and IT operations management [39]. One of its features provides data-driven insight into the risk during the deployment of a change. However, its purpose is to speed up the change approval process by automatically evaluating and applying approval decisions based on business requirements [29]. While ServiceNow looks at historical change data for insights into team performance [40], we analyse historical change, incident, and aggregated metric data for insights into team performance. However, due to the proprietary nature of ServiceNow's risk assessment feature and the lack of publicly available technical details, we were unable to directly compare our model with theirs.

A key distinction of our approach is its focus on explainability. It offers engineers transparency by providing insights into which features influence the scoring and informing users about what determines their deployment risk prediction. Since all approval decisions at ING require a human-in-the-loop, this additional insight into the scoring aids the approval step in the change deployment process, reinforcing both operational effectiveness and regulatory compliance.

3 Background

This section provides an overview of the case company, its change management process, and the risk-related factors influencing it.

3.1 Case Company

Our research examines ING, a global financial services company offering financial products and services to millions of customers. With over 15,000 engineers deploying thousands of monthly changes, ING has evolved from a traditional bank into a digital platform offering online and mobile banking services. The increasing role of ICT in finance necessitates its integration into daily operations [1]. Consequently, ING can be categorised as a financial software-defined business [21].

Banks play a systemic role in the global economy and are subject to regulatory oversight to ensure they comply with risk guidelines and policies. Regulatory agencies, particularly the European Banking Authority (EBA), significantly influence the management of ING's processes, given its European base [22]. The EBA aims to ensure effective and consistent prudential regulation and supervision across Europe's banking sector [33]. Regulatory milestones such as the revised Payment Services Directive introduced in 2017 [11] and the 2019 Guidelines on ICT and security risk management [34] have significantly influenced IT operations in the financial sector.

In 2023, the European Parliament adopted the Digital Operational Resilience Act (DORA) [1], establishing a regulatory framework for digital operational resilience, ensuring technological safety, proper functioning, and quick recovery from ICT breaches and incidents, which took effect in 2025. DORA aims to enable the effective and smooth provision of financial services while preserving consumer and market trust and confidence. It specifically addresses ICT risks 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 handling change requests and managing associated risks [12]. At ING, this process follows the Information Technology Infrastructure Library (ITIL) [12] guidelines, combined with agile principles [6], enabling multi-disciplinary teams to be responsible for entire processes and value chains, end-to-end [21]. The focus is on IT changes affecting services such as hardware, networks, middleware, and software. Each change is assigned to a single team, affects at least one configuration item (CI), and has a specific implementation moment. The configuration management database (CMDB) tracks all IT assets, processes, and changes to their attributes and relationships [12].

This process is closely tied to the incident management process [12], which manages the life cycle of all incidents. When a change is required to resolve an incident, it must be logged and processed through the change management process. Conversely, incident management is responsible for the detection and resolution of incidents that may arise from unsuccessful changes.

The change management process involves five stages: *1) Logging*: Registering IT change activities as change tickets in the service management tool, with detailed descriptions to aid approvals and information sharing. *2) Assessment & Planning*: Evaluating the risk of a change, focusing on the probability of failure and potential damage if the change fails. This stage involves noting dependencies with other teams and changes, and automating deployment impact analysis and risk calculation. *3) Approval*: Approval groups assess the readiness of a change for deployment. The product owner must approve any change released to production, accepting the delivered quality and risks. Unapproved changes are cancelled. *4) Coordinate Implementation*: Deploying the change within the agreed time window. *5) Evaluation & Closure*: Evaluating the change post-implementation to determine 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 was successful, successful with problems, failed, or cancelled.

Our study focuses on the *Assessment & Planning* and *Approval* stages by introducing incident prediction scores to help engineers determine whether a planned change is ready for deployment. Accurate prediction at this stage can reduce the likelihood of incident-prone changes reaching production.

3.3 Risk-related Influences to the Process

As a financial services provider, ING must ensure reliability and business continuity [22]. Failed changes can lead to service disruptions, which negatively impact the business. Therefore, robust change deployment management is needed to ensure compliance with governance, legal, contractual, and regulatory requirements.

Under article 9 of DORA [1], financial entities need to continuously monitor and control the security and functioning of ICT systems and tools [1]. They must minimise the impact of ICT risks by designing, procuring, and implementing security policies, procedures, protocols, and tools to ensure ICT system resilience, continuity, and availability. Specifically, for production, documented policies, procedures, and controls based on a risk assessment approach are required [34]. These should be integral to the financial entity's overall change management process to ensure all changes are recorded, tested, assessed, approved, implemented, and verified in a controlled manner [1]. The risk assessment should consider potential impacts on the continuity and quality of financial services [34]. Furthermore, post-change follow-ups should be conducted to verify the successful implementation without unexpected impacts or the need for remediation [1]. Our research supports the risk assessment approach by predicting if a change is predicted to cause an incident, thus aiding the decision on whether the change should be deployed in its current state.

Additionally, the EU's AI Act [10], introduced in early 2024, establishes a legal framework for the development, market placement, putting into service, and use of AI systems within the EU. The goal is to promote the adoption of human-centric and trustworthy AI. To align with this, we emphasise human-in-the-loop decision-making and explainability in our predictions to build user trust and improve actionability.

4 Data

We analyse 175k closed change tickets deployed in the company's production environment over one year (November 2022 to October 2023), linked to incident tickets from the same period filtered by priority levels 1 and 2 (out of 5). Priority 1 incidents are characterised by high urgency, requiring to be solved as soon as possible, and high impact, indicating critical effects company-wide or across multiple business domains. Major incidents are identified as special cases of priority 1 that have a major impact on critical business processes or services [12]. Priority 2 incidents have a medium impact on a single business domain and medium urgency. A change is considered to have triggered an incident if its identifier appears in the incident's *Caused by Change* field or is mentioned in its *Solution* field as the cause. This results in about 2.4% (around 4k) of changes being identified as incident-inducing.

We further enrich the data with aggregated team metrics, specifically for the team assigned to implement each change. These metrics are derived from incidents, changes, and releases associated with that team. Release data is only available for 23.1% of changes, providing an overview of all production releases executed through pipelines and their compliance status. For each release, there is information on team metadata, start and end times, release outcomes (success, partial success, or failure), and control columns such as product owner approvals, peer code reviews, and related changes. Aggregated team metrics are linked to change tickets via the *IT Product*, which represents the deliverable owned by a single engineering team.

To protect confidentiality, we report only aggregated dataset characteristics rather than detailed feature statistics.

5 Methods

To enhance change deployment reliability, we propose generating a predictive incident prediction score during the planning phase. We start by reviewing the company's current risk assessment approach, then introduce and evaluate three classification-based ML models. We identify the best-performing model and further test its accuracy by incorporating additional features.

5.1 Baseline: Rule-Based Scoring

The company's change deployment risk assessment is supported by an automated calculation within its IT Service Management (ITSM) tool. This calculation is configured by the company and is based on data available in the tool and the CMDB. The resulting score provides an indication of the probability of failure for the planned change and the potential impact. The calculation result depends on the quality of the information in the change tickets and the CMDB, making it reliant on human effort and expertise to ensure detailed and accurate tickets.

The ITSM tool uses specific criteria, or business rules, to derive a risk category, considering factors of the probability of failure (e.g., the scope of impacted IT services, deployment complexity, and incident history) and of potential damage (e.g., Confidentiality, Integrity, and Availability ratings, SOx criticality [41], and recoverability). The output is a score from 0 to 100, mapped to risk levels: low (0-33), medium (34-59), and high (60-100).

This score is generated during the *Assessment & Planning* stage and informs the *Approval* stage, where the approval group uses it to evaluate deployment readiness [21]. This supports the four-eye principle, a widely adopted practice in financial institutions and other safety-critical domains, which requires at least two independent reviewers to approve each change before deployment. Due to confidentiality constraints, we cannot provide a deeper description of the exact business rules, weightings, or configuration logic used in the current baseline.

5.2 Scoring using ML models

To improve the current rule-based assessment, we apply data-driven ML methods to historical change and incident data from ING's ITSM system. Unlike the baseline, which depends on manually entered data and expert judgment, ML models learn patterns from past deployments that could negatively impact stability and cause incidents, reducing subjectivity and improving efficiency. This approach helps engineers and change managers maintain high service quality and fosters continuous improvement in a fast-moving environment. Also, it helps them by ensuring that significant changes receive closer scrutiny, particularly in areas such as the execution plan, testing, and rollback, while allowing lower-risk changes to proceed with less oversight. This allows for streamlining approvals by IT leads. This is particularly beneficial for Deployment Change Advisory Boards (DCABs), also known as Change Advisory Boards (CABs).

In addition to predictive scoring, we offer explainability of the models by employing SHAP, a unified framework for interpreting predictions [25]. SHAP values enable us to provide users with insights into a single prediction of a planned change. They show how each feature contributes positively or negatively to a given prediction, and identify the most important features that drive the ML models' predictions. SHAP was chosen because it is particularly well-suited for boosted tree models [27], provides individualised explanations for each prediction, and is widely used in industry and research [30]. We report the top 15 most important features influencing the predictions of the best-performing ML model. This approach aligns with other ongoing research at ING, where SHAP is similarly used to provide explanations of an ML model's decision to end users [37].

By explaining predictions, we increase user understanding and acceptance [2, 36]. From a regulatory perspective, explainability also ensures that model-driven decisions remain auditable and accountable, supporting financial sector requirements for transparency, traceability, and human oversight.

5.2.1 Feature Preparation. Before modeling, we perform extensive data processing and feature engineering to ensure the quality and relevance of our data.

For processing the incident data, we begin by removing duplicates and filtering out irrelevant incidents based on the company's closure codes, such as 'Invalid event' or 'Withdrawn by Customer'. Major incidents are a sub-category of Priority 1 incidents, but we introduce them separately as Priority 0. To enrich the data, we check if a change identifier is mentioned in the incident ticket as a cause and add this identifier to the *Caused by Change* field if not already present.

For processing the change data, we focus on closed changes. In preparing the features, we start by removing stopwords from the *Short Description* and *Description* columns of the changes. We then generate Natural Language Processing (NLP) features using scikit-learn's [35] CountVectorizer, which are fed into a TruncatedSVD for latent semantic analysis. Date features are generated from the change start timestamp, including the starting hour, day of the week, quarter, month, day of the year, day of the month, week of the year, and whether it is a weekend.

We merge both datasets, incorporating all categorical, numerical, date, and NLP features. For categorical features, we include metadata of a change, such as attributes of the CI, security ratings (confidentiality, integrity, and availability), change management information (change category, change state, CAB approval group), incident and support management information (e.g., support offering or assignment group), compliance and criticality information, and deployment and architecture information (automated deployment, fallback options, and redundant architecture). The data is merged with incident data to determine which changes caused incidents by linking the *Caused by Change* incident fields with change data. Connections where the change occurred after the incident are removed to ensure only causal links are retained.

As described in Section 4, we incorporate aggregated team metrics as features, derived from incident, change, and release data. These features consist of median weekly and monthly aggregates of team-level performance metrics, computed using the *IT Product* as a proxy for team identity. To capture team performance, we compute aggregate metrics such as the number of changes, percentage of successful changes, number of changes causing incidents, number of high-priority incidents, percentage of successful releases, and number of releases. However, since the *IT Product* field currently has only 50% coverage (with ongoing efforts to improve this), we exclude tickets lacking this information in our analysis for RQ3.

5.2.2 Approach. The *Caused by Change* field labels whether a given change results in an incident. A change is labelled as causing an incident if it has at least one link to a high-priority incident (i.e., major, priority 1, priority 2), independent of how many incidents are linked. For each change, we also record the highest incident priority, which is later used for sample weighting. This labelling process results in a binary classification, indicating that a change has either induced at least one high-priority incident (1) or not (0).

Based on these labels and the prepared features, we train multiple classification ML models on a training set. We split the data temporally, using the first eight months for training (120K changes), then two months for validation (26K changes), and the last two months as the test set (30K changes). We implemented three gradient boosting algorithms in Python: HistGradientBoostingClassifier from scikit-learn [35] (referred to as HGBC), LightGBM [23], and XGBoost [9]. We focused on classification models that can be explained post-hoc to support explainability and auditability, which is a critical requirement in our context.

Simpler models were unable to handle the Not a Numbers (NANs) inherent in the data, which provide meaningful information regarding unfilled fields. To preserve this information, we opted for models capable of handling such values, unlike other models like linear regression, support vector machine or random forest classifiers.

Through iterative training, the model learns which features are useful indicators for predicting the labelled outcome. The target variable is an incident prediction score between 0 and 100, indicating the probability that the planned change will trigger an incident.

6 Evaluation

Each model serves as a classifier that produces a numerical score between 0 and 100, indicating the probability that a change will trigger a high-priority incident. To convert these into binary predictions, we identify an optimal threshold using a temporal split: eight months for training and two months for validation. The threshold that maximizes the weighted F2-measure (defined below) on the validation set is selected and applied to a held-out two-month test set.

To simulate real-world deployment, we evaluate the best-performing model using a sliding window setup. In practice, new changes and incident data are generated continuously, and risk assessments must remain aligned with the latest information. Therefore, the model is retrained daily, incorporating the most recent data, and produces predictions on a weekly cycle to support the organisation's change planning and approval processes. From each prediction window, we derived confusion matrices and computed evaluation metrics to monitor performance and stability over time.

Precision, defined as $\frac{TP}{TP+FP}$, captures how many positive predictions are correct. Given the significant class imbalance due to the rarity of high-priority incidents triggered by a change, we report weighted variants of recall and F-measure to more fairly assess model performance. Weighted recall or wR (as defined in Equation 1) averages recall per class, weighted by the number of true instances in each class:

$$\text{wR} = \sum_{c \in C} \frac{n_c}{N} \cdot \frac{TP_c}{TP_c + FN_c} \tag{1}$$

Here, C is the set of classes (high-priority incident/ no high-priority incident), n_c the number of true instances of class c, and N the total number of instances.

To reflect the importance of minimising false negatives so that we do not miss incident-inducing changes when they are being planned, we compute the weighted F2-measure or wF_2 (as defined in Equation 2). Greater emphasis is placed on recall using β=2, which is inspired by prior work [5].

$$\text{w}F_\beta = \sum_{c \in C} \frac{n_c}{n_0 + n_1} \cdot \left(\frac{(1+\beta^2) \cdot P_c \cdot R_c}{\beta^2 \cdot P_c + R_c} \right) \tag{2}$$

Where P_c and R_c are the class-specific precision and recall scores.

Our evaluation aligns with the *Assessment & Planning* stage of the change management process, where the objective is to proactively flag changes that trigger incidents before deployment. A high recall is crucial to ensure that most incident-inducing changes are identified, even at the cost of some false positives. These predictions are accompanied by model explanations to assist change managers and engineers in making informed decisions on whether to proceed and if additional actions are needed. By reviewing the prediction explanations, we ask for more due diligence from the engineer to avoid incidents similar to those in historic changes. Therefore, a high recall (even at moderate precision) supports risk-aware decision-making.

Finally, to account for the practical significance of varying incident severities, we introduce priority-based weighting in our evaluation. Changes that lead to major or priority 1 incidents receive a weight of 5, those resulting in priority 2 incidents receive a weight of 3, and non-incident changes are weighted as 1. This ensures that the evaluation process gives proportionally more importance to correctly identifying severe incidents.

7 Results

This section compares the effectiveness of the company's existing rule-based risk assessment against the ML methods. We first evaluate the baseline to establish a performance benchmark, then assess how the ML models improve predictive accuracy. We also examine the stability of the best performing model its performance over time, and examine the impact of incorporating aggregated team metric data.

7.1 RQ1 - Performance of Rule-Based Approach

At ING, changes with a risk score of 60 or higher are considered a high-risk level during the *Assessment & Planning* stage of the change deployment management process. In this paper, we define a high-risk level to indicate that the case company considers a given change as potentially resulting in an incident. Using this threshold of 60 on the two-month test set, the baseline performance for all metrics is presented in Table 1.

Table 1: Performance of Baseline, ML Models, and LightGBM with and without Additional Features

				LightGBM	
Metric	Base	HGBC	XGB	Without	With
Threshold	60	99	98	17	29
Precision	0.04	0.02	**0.05**	0.02	0.04
wR	0.56	0.86	0.89	0.91	**0.93**
wF_2	0.88	0.88	0.90	0.92	**0.93**
AUC	0.55	0.61	**0.71**	0.67	0.60

The baseline approach achieves very low precision, indicating that most changes flagged as high risk do not actually result in incidents. This is largely due to the highly imbalanced nature of the dataset, where incident-inducing changes are rare, making it difficult for static rules to distinguish incident-inducing changes from safe ones. However, the wR and wF_2 are better, since the recall identifies 56% of the relevant cases within the test set, and the wF_2 shows the weighted precision and recall, giving a good indication for our business purposes. When examining the area under the receiver operating curve (AUC), the score of 0.56 is close to 0.5, suggesting that the model has almost no discriminating ability [18] and thus can be interpreted as failing in classifying this task [31].

Finding 1. *The baseline risk assessment fails to effectively identify high-risk changes, as evidenced by low precision, a moderate wR of 56%, and an AUC of 0.56, indicating poor discriminative performance.*

7.2 RQ2 - Comparison of ML Models vs. Rule-Based Approach

7.2.1 Performance. We evaluated three ML models, HGBC, LightGBM, and XGboost, against the rule-based baseline, optimising each model's threshold on a validation set and assessing final performance on a two-month test set (see Table 1). For each performance metric, the best-performing value is highlighted in bold.

The baseline achieves a precision of 0.04, but is outperformed by XGBoost, which achieves the highest precision (0.05). Both HGBC and LightGBM record lower precision values of 0.02. However, our primary objective is to maximise correct identifications, acting more as a guard dog rather than prioritising the minimisation of false positives. This approach favours recall over precision, making the small loss in precision acceptable.

In terms of wR, all ML models substantially outperform the baseline. LightGBM achieves the highest value (0.91), followed by XGBoost (0.89) and HGBC (0.86), whereas the baseline reaches 0.56. This performance gap is also apparent in the wF_2, which weighs recall more heavily. LightGBM again leads with a wF_2 of 0.92, followed by XGBoost(0.90), and HGBC (0.88), with the baseline also scoring 0.88. Despite matching HGBC on wF_2 numerically, this value of the baseline is less meaningful given its substantially lower recall, meaning it identifies fewer relevant cases overall.

The AUC scores provide further insight into each model its overall discriminative ability. XGBoost achieves the highest AUC (0.71), followed by LightGBM (0.67) and HGBC (0.61), while the baseline records the lowest score (0.55). As an AUC score falling between 0.5 and 0.6 means it fails in classifying this task [31] and has near-random performance, the ML models demonstrate moderate improvements in classification ability over the baseline.

There is also considerable variation in decision thresholds across models. HGBC and XGBoost both use very high thresholds (99 and 98, respectively), indicating a low decision boundary likely aimed at maximising recall. In contrast, LightGBM uses a much lower threshold (17), suggesting that its scoring scale differs and may necessitate model-specific calibration.

Overall, LightGBM offers the best performance in terms of wR and wF_2, aligning well with our goal of maximising correct identifications. Despite relatively modest AUC values, all ML models outperform the rule-based baseline and are better suited for high-recall tasks.

7.2.2 Important Features. Next, we examine the best-performing model, LightGBM. Using SHAP, we analyse the top 15 most important features for prediction outcomes, as shown in Fig. 1. Description values have been concatenated due to the abundance of NLP features. Interestingly, the textual features *Full Description* and *Short Description* have the most influence, highlighting that the content of the change significantly impacts the outcome. This is followed by the *CI Name*, which represents the configuration item that is being changed. Additionally, the *Assignment Group*, which refers to the team assigned to work on the change, and the *Support Offerings*, which are the team supporting the CI, are also highly influential. Other features, such as the change approval board group, the CI configuration group, and the owner of the CI, are less important but still influence the predictions.

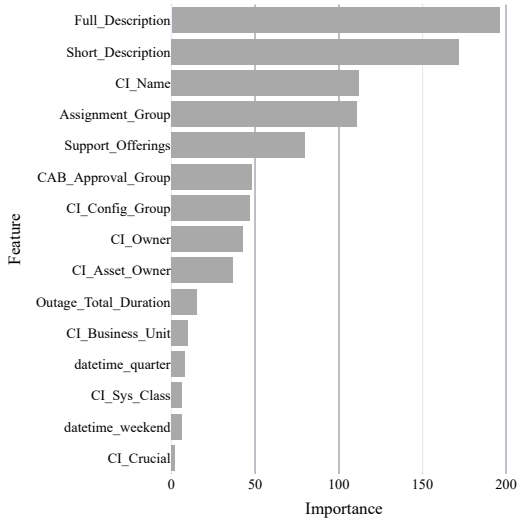


Figure 1: Feature Importance of LightGBM.

7.2.3 Stability Over Time. To further assess the stability of LightGBM, we used a sliding window evaluation to simulate its performance over time, comparing it with the rule-based baseline method (see Fig. 2). This evaluation shows that LightGBM consistently achieves higher wR values compared to the baseline over time, indicating a stronger ability to correctly identify positive cases. Both wR values show some fluctuations, with a peak around early September for LightGBM. This inconsistency is likely due to a period of minimal changes at the company, during which modifications were infrequent and strictly supervised.

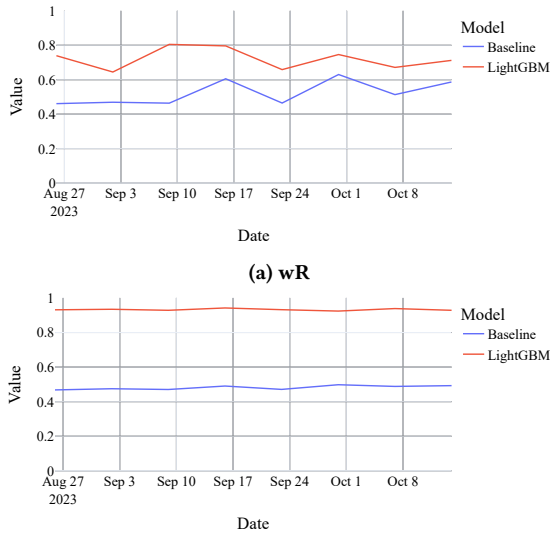
In contrast, the wF_2 for both models remains relatively stable over time. LightGBM again outperforms the baseline, highlighting its ability to maintain a good balance between precision and recall, which is important for our goal since we want to maximise the identification of correct cases. These results suggest that LightGBM not only delivers performance improvements over the baseline but also maintains its performance in a production-like environment.

Finding 2. *Among the evaluated models, LightGBM performs best in wR and wF_2, making it the most suitable for high-risk change detection. Key features are primarily textual and team-related. Despite some fluctuations in wR during periods of low change activity, LightGBM shows stable wF_2 performance over time.*

7.3 RQ3 - Inclusion of Aggregated Team Metric Data

7.3.1 Performance. The best-performing model, LightGBM, is used to evaluate whether the inclusion of additional features from metric data aggregated by *IT Product* improves its performance. The performance is compared with and without these additional features (See Table 1).

The addition of team aggregated features results in a higher classification threshold and a noticeable improvement in both precision and wR, increasing from 0.02 to 0.04 and from 0.91 to 0.93, respectively. The wF_2 also improves slightly from 0.92 to 0.93, reflecting that the model is better at balancing between precision and



(a) wR

(b) wF_2

Figure 2: Sliding Window of Baseline vs. LightGBM Performance.

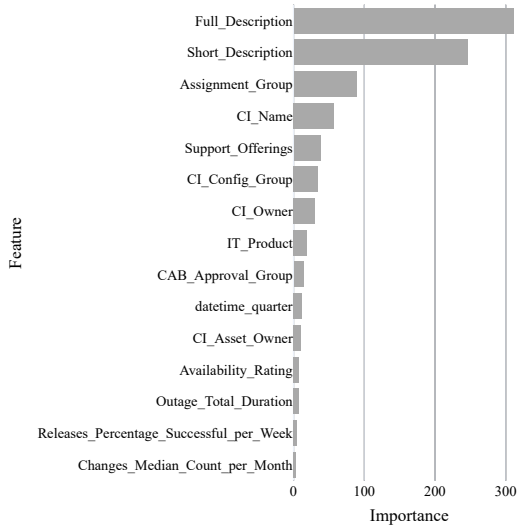
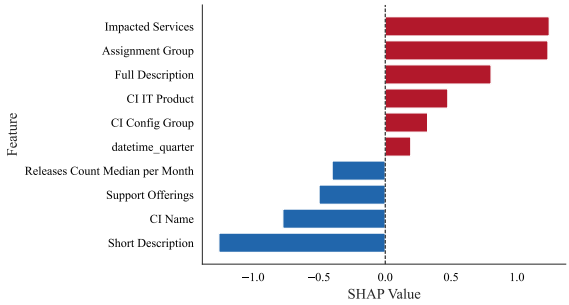
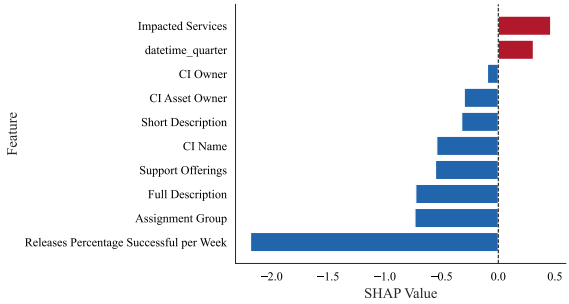


Figure 3: Feature Importance of LightGBM with Additional Features.



(a) High Scoring Change



(b) Low Scoring Change

Figure 4: Top 10 SHAP Values for Two Example Changes.

recall. However, the AUC score decreases slightly from 0.67 to 0.60, suggesting that the model's performance based on the optimised threshold improves, but its overall scoring consistency across all instances slightly worsens. Despite this drop in AUC, the gain in precision and recall indicates a more confident identification of positive instances. Integrating data from other sources, with an emphasis on ease of merging, proves to be a worthwhile approach, given the coverage issues discussed in Section 5).

7.3.2 Important Features. Upon closer inspection of the top 15 SHAP ranking of the model with the added features (see Fig. 3), we observe the inclusion of these new features. Notably, the *IT Product* ranks 8th, while *Release Percentage Successful per Week* and *Changes Median Count Per Month* are also included, appearing near the bottom of the ranking.

7.3.3 Examples. Fig. 4 shows two SHAP plots explaining individual predictions. To improve interpretability, we display the ten most impactful features. For multi-component features such as *Full Description* and *Short Description*, we use the maximum signed SHAP value to highlight the most influential element. Features with positive SHAP values (right side) contribute to a higher predicted risk, while negative values (left side) reduce it, offering users a transparent view into the model's decision-making process by quantifying the contribution of each feature to the output.

Figure 4a shows a high-scoring change, specifically a monthly infrastructure patch that resulted in a user-facing incident. In this example, *Impacted Services* is the most influential positive feature, pushing the score toward a high-risk classification. Other top contributors, which were also in the top 5 in Figure 3, include *Assignment Group*, *Full Description*, *Short Description*, *CI Name*, and *Support Offerings*, which appear on both sides of the SHAP scale, suggesting that they have different effects depending on context. Additionally, the team aggregated feature *Releases Count Median per Month* also emerges as an important factor.

Figure 4b illustrates a low-scoring change involving a routine certificate renewal for network switches, which did not result in any incident. Here, most of the top features contribute negatively, pulling the score downward. Notably, the additional feature *Releases*

Percentage Successful per Week has a strong negative SHAP value, indicating its role in de-risking the change. While *Impacted Services* is again present, its influence is limited. The same core features observed in the high-scoring case also appear here, but with negative contributions, which emphasise their dynamic impact depending on the nature of the change.

Finding 3. *Adding aggregated team metrics to the LightGBM model yields a modest performance improvement, particularly in precision, wR, and wF_2, with new features like IT Product, release success metrics, and change frequency emerging as moderately important predictors.*

8 Discussion

Both practitioners and researchers can utilise our results for using data-driven ML methods to improve the risk assessment of changes before deployment, while taking into account the corresponding threats to their validity.

8.1 Implications for Practitioners

Due to the low amount of incident-inducing changes present in the data, all models inevitably exhibit low precision. This can create operational challenges, most notably alert fatigue [7]. Predictions should therefore not be interpreted as binary signals but as a **risk-ranking mechanism** that highlights a small subset of changes that need additional attention. To better align with existing approval workflows, practitioners benefit from **multi-level predictions** (e.g., low/ medium/ high risk), enabling automatic handling of low-risk changes while directing expert attention to the highest-risk ones. To keep such a system usable, a brief **human-in-the-loop** step can be incorporated in which experts dismiss obvious false positives, apply adaptive thresholds to focus on the highest-risk changes, and provide lightweight feedback (e.g., "useful/ not useful") to iteratively refine the model. These safeguards enable meaningful decision support without overwhelming engineers or approvers.

Our results show that LightGBM outperforms the rule-based baseline, **support**ing engineers, change managers, and approvers, especially those with less experience. Whereas experienced practitioners often rely on intuition and tacit knowledge of team behaviours and historical patterns, ML models formalise these insights from historical data. This enables newer users to more accurately assess the risk of their planned changes.

Model **explainability** through SHAP values enhances interpretability and practical relevance by linking predictions to understandable features (see Fig. 4). Key features such as *Impacted Services*, *Assignment Group*, and *Releases Percentage Successful per Week* correspond to known risk factors. SHAP plots clarify how each feature influences risk, e.g., a successful release history lowers risk, while involvement of critical services raises it. This makes the model's behaviour more transparent and supports informed decision-making.

The feature importance results of RQ2 and RQ3 show similar trends. The textual descriptions dominate both, indicating that the content of the change strongly affects the predictions. These are followed by features describing the CI being changed, such as *CI Name* and *CI Config Group*, and the responsible team, like the *Assignment Group* or *Support Offerings*. **Team-related influences** may stem from factors beyond reliability, such as reporting more faults or handling riskier tasks. Future research should explore these dynamics to prevent bias against specific teams. Interestingly, CAB approval group importance diminishes once aggregated team metrics are added, suggesting that the other team-related features already capture relevant context.

Contrary to expectations, most rule-based baseline factors were absent from the top 15, except *Availability Rating*. Expected factors such as *Confidentiality*, *Integrity*, and features related to deployment complexity did not significantly influence the predictions. Surprisingly, the features *Datetime Quarter* and *Outage Total Duration*, both time-related, proved influential. This indicates that the specific quarter in which a change occurs influences the deployment risk. One possible explanation is the influence of error budgets [7], since if a company has already experienced significant user-impacting outages within a given period, it may enforce a freeze on high-risk changes to prevent further service degradation. Such policies directly affect the distribution of risky changes during specific timeframes (see Finding 2), which is then reflected in the model's predictions. Additionally, the total duration of outages previously experienced by a team significantly impacts the deployment risk of their planned changes. Added features such as *Release Percentage Successful per Week* and *Changes Median Count per Month* were found to influence the prediction, likely serving as proxies for team maturity.

Features can be grouped into actionable and non-actionable, and this distinction influences how they can be acted upon during the decision if the change is ready to be approved for deployment. **Actionable features** are those that engineers can directly influence before deploying a change, such as refining the description, improving testing plans, or adjusting rollback strategies, which enables practitioners to address concrete issues highlighted by a high score. In contrast, **non-actionable features**, such as machine-specific characteristics or team-related metadata, cannot be modified for the current deployment but still provide valuable context. If a particular CI consistently appears in high-scoring changes, this signals structural or historical risk that may warrant additional safeguards, more thorough reviews, phased rollouts, or enhanced monitoring during deployment. Similarly, if certain assignment groups are frequently associated with riskier changes, whether due to the nature of the work or team experience, it may be advisable to involve additional reviewers or support from lower-risk teams to ensure safer deployments.

Data quality significantly influences the outcomes of AIOps methods [36]. Increasing practical awareness of the importance of linking incidents to changes would yield better labels for model training. Ongoing efforts at ING have raised awareness about the importance of data quality and robust incident and change management registration to improve reliability. Scaling this across more teams and companies would benefit broader AIOps initiatives.

Lastly, due to the incomplete *IT Product* field, additional features could only be applied to part of the dataset, limiting the full view of the production environment. Better **integration of data from other sources**, with an emphasis on ease of merging, proves to be a worthwhile approach.

8.2 Implications for Researchers

A major challenge in applying ML to AIOps is the **data quality in practice**, especially the effort required to link incidents to their causing changes [36]. Future research should focus on automating both historical and real-time linking, building on prior work [5, 17, 22], to enable higher-quality labels and more accurate risk predictions. The features identified in this study could serve as a foundation for such research.

Explainability remains critical for adoption. Researchers should explore how different roles, such as engineers or change managers, prefer to receive and interpret results, and how these can be effectively communicated. Researchers should consider the end-users' perspective, focusing not only on presenting probabilities but also on how these insights will be utilised in practice.

Finally, the distinction between **actionable and non-actionable features** highlights the need for further investigation into how models can support practitioners. For example, description fields or deployment timing are actionable, while team metadata is not directly actionable but still provides context for safeguards. Future work should explore how to present both types of features effectively, guiding engineers in reducing deployment risks. Additionally, while SHAP offers valuable interpretability, it has limitations: it explains correlations identified by the model rather than causal relationships [26]. Therefore, it cannot reliably indicate which features should be manipulated to achieve a desired outcome. Understanding how to combine interpretability with causal reasoning represents an important direction for research.

8.3 Threats to Validity

The threats and limitations of the study are categorised into construct validity, external validity, and reliability [43].

1) Construct validity: We used wF_2 to emphasise recall over precision. Other values of β can be employed depending on the use case. Adjusting β for different scenarios could significantly impact performance outcomes, tailoring the evaluation to different application needs. In RQ3, higher precision and recall came with a slight AUC decrease, which is an acceptable trade-off given that AUC can be misleading in imbalanced datasets, as it emphasises overall ranking rather than the correct identification of the minority-class [47]. Low precision is expected because only a small fraction of changes cause incidents, and in this context recall is prioritised, as missing a high-priority incident carries far greater consequences than reviewing additional false positives.

2) External validity: Data-driven methods outperformed the baseline in terms of wR and AUC scores, with LightGBM also exceeding the baseline on the wF_2. These results demonstrate the value of data-driven approaches over approaches that rely predominantly on human judgment. However, they must be interpreted within the context in which the study was conducted. Similar outcomes are expected in other financial software-defined environments, although differences in processes, metadata quality, and taxonomies may influence replication outcomes. Due to confidentiality constraints, detailed feature data cannot be shared; however, our analysis provides insights expected to generalise to comparable settings.

Beyond finance, the methodology can extend to organisations with structured, audited change-management workflows. Highly regulated sectors such as healthcare and government follow similar compliance-driven processes [28], and often benefit from explainable, data-driven models that integrate into existing approval processes. Although available features and data completeness vary across domains, the overall workflow (feature extraction, supervised learning, and human-in-the-loop review) remains transferable with domain-specific adaptation of features and thresholds.

The temporal splitting and class imbalance add further considerations. Operational environments often show seasonal fluctuations (e.g., code freezes), which can lead to distribution shifts between training, validation, and test periods. Because the minority class is very small, even modest month-to-month variation in incident-inducing changes may influence precision and wF_2, making temporal evaluation more volatile than random splits. Our long historical training period, two hold-out sets, and imbalance-aware metrics help mitigate these effects, but they remain relevant when extrapolating results.

3) Reliability was evaluated via a validation set and a sliding-window evaluation over time. Precision and wF_2 remained stable, while variations in wR can be attributed to a strict change phase. Additionally, the challenge of working with an extremely imbalanced dataset, which is common in AI problems [20] and observed in other AIOps research on incident management [36], was addressed by weighting labels according to the priority of incidents. Alternative strategies, such as class downsampling or upsampling [20], or adjusting the currently used weight distributions, could be explored, but our current approach proved most effective.

9 Conclusion

This study presents a predictive incident prediction score that can assist engineers during the assessment & planning phase of IT changes in a demanding production environment. Our results demonstrate that data-driven ML methods outperformed currently employed approaches that rely on business rules and human effort. Among the tested models, LightGBM yielded the best performance, particularly when enriched by aggregated team metrics.

The model's predictions are explainable through SHAP analysis, which identifies the key features influencing outcomes. Notably, metadata related to the team handling the change, the specific machine involved, and other risk indicators inherent to the product were important in shaping the predictions.

Our findings highlight the value of adopting a data-driven ML approach to improve the reliability of IT systems. In highly regulated environments such as finance, where auditability, traceability, and human oversight are mandatory, this approach provides interpretable risk scores that support compliance. We also highlight the importance of incorporating qualitative metrics, such as team metric features, into change and incident management processes. By enabling more focused scrutiny of changes with a higher likelihood of causing incidents, this approach facilitates proactive risk mitigation and contributes to a more reliable IT environment across the organization.

Acknowledgments

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

References

[1] 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

[2] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE access* 6 (2018), 52138–52160.

[3] Salman Ahmed, Muskaan Singh, Brendan Doherty, Effirul Ramlan, Kathryn Harkin, Magda Bucholc, and Damien Coyle. 2023. An empirical analysis of state-of-art classification models in an it incident severity prediction framework. *Applied Sciences* 13, 6 (2023), 3843.

[4] Gergő Barta. 2018. The increasing role of IT auditors in financial audit: risks and intelligent answers. *Business, Management and Education* 16, 1 (2018), 81–93.

[5] 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)*. 112–123. doi:10.1109/CLOUD53861.2021.00025

[6] 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.

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

[8] Junjie Chen, Shu Zhang, Xiaoting He, Qingwei Lin, Hongyu Zhang, Dan Hao, Yu Kang, Feng Gao, Zhangwei Xu, Yingnong Dang, and Dongmei Zhang. 2021. 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* (Virtual Event, Australia) *(ASE '20)*. Association for Computing Machinery, New York, NY, USA, 373–384. doi:10.1145/3324884.3416624

[9] 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* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 785–794. doi:10.1145/2939672.2939785

[10] European Commission. 2024. The act texts. https://artificialintelligenceact.eu/the-act/

[11] EBA. 2021. Revised Guidelines on major incident reporting under PSD2. https://www.eba.europa.eu/regulation-and-policy/payment-services-and-electronic-money/guidelines-on-major-incidents-reporting-under-psd2

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

[13] Stephen Elliot. 2014. DevOps and the cost of downtime: Fortune 1000 best practice metrics quantified. *International Data Corporation (IDC)* (2014).

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

[15] Supriyo Ghosh, Manish Shetty, Chetan Bansal, and Suman Nath. 2022. How to fight production incidents? an empirical study on a large-scale cloud service. In *Proceedings of the 13th Symposium on Cloud Computing* (San Francisco, California) *(SoCC '22)*. Association for Computing Machinery, New York, NY, USA, 126–141. doi:10.1145/3542929.3563482

[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)*. 268–271. doi:10.1109/CNSM.2016.7818430

[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*. 937–942. doi:10.1109/NOMS.2016.7502929

[18] Zhe Hui Hoo, Jane Candlish, and Dawn Teare. 2017. What is an ROC curve? *Emergency Medicine Journal* 34, 6 (2017), 357–359. arXiv:https://emj.bmj.com/content/34/6/357.full.pdf doi:10.1136/emermed-2017-206735

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

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

[21] Eileen Kapel, Luis Cruz, Diomidis Spinellis, and Arie van Deursen. 2024. Enhancing Incident Management: Insights from a Case Study at ING. In *Proceedings of the 1st IEEE/ACM Workshop on Software Engineering Challenges in Financial Firms* (Lisbon, Portugal) *(FinanSE '24)*. Association for Computing Machinery, New York, NY, USA, 1–8. doi:10.1145/3643665.3648048

[22] Eileen Kapel, Luis Cruz, Diomidis Spinellis, and Arie Van Deursen. 2024. On the Difficulty of Identifying Incident-Inducing Changes. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice* (Lisbon, Portugal) *(ICSE-SEIP '24)*. Association for Computing Machinery, New York, NY, USA, 36–46. doi:10.1145/3639477.3639755

[23] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[24] Ze Li, Qian Cheng, Ken Hsieh, Yingnong Dang, Peng Huang, Pankaj Singh, Xinsheng Yang, Qingwei Lin, Youjiang Wu, Sebastien Levy, and Murali Chintalapati. 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)*. USENIX Association, Santa Clara, CA, 389–402. https://www.usenix.org/conference/nsdi20/presentation/li

[25] Scott Lundberg. 2017. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874* (2017).

[26] Scott Lundberg. 2021. Be careful when interpreting predictive models in search of causal insights. https://medium.com/data-science/be-careful-when-interpreting-predictive-models-in-search-of-causal-insights-e68626e664b6

[27] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. 2018. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888* (2018).

[28] Adya Mishra. 2019. Exploring ITIL and ITSM Change Management in Highly Regulated Industries: A Review of Best Practices and Challenges. *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences* 7 (10 2019), 1–8. doi:10.5281/zenodo.14838584

[29] Greg Morrison. 2021. Automated Change Impact Assessments. https://www.servicenow.com/community/it-service-management-blog/automated-change-impact-assessments/ba-p/2267267

[30] Edoardo Mosca, Ferenc Szigeti, Stella Tragianni, Daniel Gallagher, and Georg Groh. 2022. SHAP-based explanation methods: a review for NLP interpretability. In *Proceedings of the 29th international conference on computational linguistics*. 4593–4603.

[31] Francis Sahngun Nahm. 2022. Receiver operating characteristic curve: overview and practical use for clinicians. *Korean journal of anesthesiology* 75, 1 (2022), 25–36.

[32] Didrik Nielsen. 2016. *Tree boosting with xgboost-why does xgboost win" every" machine learning competition?* Master's thesis. NTNU.

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

[34] 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

[35] 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.

[36] Youcef Remil, Anes Bendimerad, Romain Mathonat, and Mehdi Kaytoue. 2024. AIOps Solutions for Incident Management: Technical Guidelines and A Comprehensive Literature Review. arXiv:2404.01363 [cs.OS] https://arxiv.org/abs/2404.01363

[37] Tom-Martijn Roelofs, Eduardo Barbaro, Svetlana Pekarskikh, Katarzyna Orzechowska, Marta Kwapień, Jakub Tyrlik, Dinu Smadu, Michel Van Eeten, and Yury Zhauniarovich. 2024. Finding harmony in the noise: Blending security alerts for attack detection. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*. 1385–1394.

[38] Ranjan Sapkota, Shaina Raza, and Manoj Karkee. 2025. Comprehensive analysis of transparency and accessibility of chatgpt, deepseek, and other sota large language models. *arXiv preprint arXiv:2502.18505* (2025).

[39] ServiceNow. [n. d.]. What is ServiceNow? https://www.servicenow.com/what-is-servicenow.html

[40] ServiceNow. 2024. Change Success Score Product Documentation Release Washington DC. https://docs.servicenow.com/bundle/washingtondc-it-service-management/page/product/change-management/concept/change-success-score.html

[41] 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.

[42] Xiaohan Yan, Ken Hsieh, Yasitha Liyanage, Minghua Ma, Murali Chintalapati, Qingwei Lin, Yingnong Dang, and Dongmei Zhang. 2023. Aegis: Attribution of Control Plane Change Impact across Layers and Components for Cloud Systems. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. 222–233. doi:10.1109/ICSE-SEIP58684.2023.00026

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

[44] Ennan Zhai, Ang Chen, Ruzica Piskac, Mahesh Balakrishnan, Bingchuan Tian, Bo Song, and Haoliang Zhang. 2020. Check before You Change: Preventing

Correlated Failures in Service Updates. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. USENIX Association, Santa Clara, CA, 575–589. https://www.usenix.org/conference/nsdi20/presentation/zhai

[45] 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* (Athens, Greece) *(ESEC/FSE 2021)*. Association for Computing Machinery, New York, NY, USA, 527–539. doi:10.1145/3468264.3468543

[46] Yujin Zhao, Ling Jiang, Ye Tao, Songlin Zhang, Changlong Wu, Tong Jia, Xiaosong Huang, Ying Li, and Zhonghai Wu. 2023. Identifying Root-Cause Changes for User-Reported Incidents in Online Service Systems. In *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*. 287–297. doi:10.1109/ISSRE59848.2023.00028

[47] Quan Zou, Sifa Xie, Ziyu Lin, Meihong Wu, and Ying Ju. 2016. Finding the Best Classification Threshold in Imbalanced Classification. *Big Data Research* 5 (2016), 2–8. doi:10.1016/j.bdr.2015.12.001 Big data analytics and applications.