Ingénierie des Systèmes d'Information
Vol. 31, No. 2, February, 2026, pp. 511-519
Journal homepage: http://iieta.org/journals/isi

Predicting Incident Resolution Time in IT Service Management via Lifecycle Feature Aggregation and Machine Learning

Mulyati[1*], Deris Stiawan[2], Abdul Rahman[3], Moh'D Suliman Shakkah[4], Rahmat Budiart[5]

[1] Department of Information Systems, Universitas Multi Data Palembang & Faculty of Engineering, Universitas Sriwijaya, Palembang 30113, Indonesia
[2] Faculty of Computer Science, Universitas Sriwijaya, Indralaya 30662, Indonesia
[3] Faculty of Computer Science, Universitas Multi Data Palembang, Palembang 30113, Indonesia
[4] Faculty of Science and Information Technology, Department of Computer Science, Irbid National University, Irbid 21110, Jordan
[5] College of Computing and Information, Al-Baha University, Al Aqiq 65779, Saudi Arabia

Corresponding Author Email: muliati@mdp.ac.id

https://doi.org/10.18280/isi.310219

Received: 4 November 2025
Revised: 10 January 2026
Accepted: 17 February 2026
Available online: 28 February 2026

Keywords:
incident resolution time prediction, lifecycle feature aggregation, machine learning, predictive process monitoring, IT service management, feature selection, ensemble learning, LightGBM

ABSTRACT

Accurate prediction of incident resolution time is critical for improving service quality and operational efficiency in IT service management (ITSM). However, most existing approaches rely on time-zero features and fail to capture the dynamic evolution of incidents throughout their lifecycles. This study proposes a lifecycle-aware machine learning framework that leverages aggregated lifecycle features derived from ServiceNow incident logs. The proposed approach consists of data preprocessing, lifecycle feature aggregation, feature selection using Light Gradient Boosting Machine (LightGBM), and predictive modeling using Linear Regression (LR), Random Forest (RF), and LightGBM regressors. Experimental results demonstrate that the RF model trained on aggregated features achieves the best performance, with an R^2 of 0.8318 and a Mean Absolute Error (MAE) of 60.67 hours, significantly outperforming models trained on non-aggregated data (R^2 = 0.5412). The aggregation process effectively reduces noise, preserves temporal context, and improves interpretability by highlighting key process-related features such as sys_mod_count, assignment_group, and reassignment_count. These findings confirm that lifecycle feature aggregation provides a robust and interpretable solution for predictive process monitoring and decision support in university Computer Security Incident Response Teams (CSIRTs).

1. INTRODUCTION

In the era of higher education digitalization, universities increasingly rely on information technology (IT) services to support academic, research, and administrative activities, as highlighted in prior studies on e-learning adoption and digital transformation in higher education [1]. Campus IT infrastructures encompass learning management systems (LMS), student management systems, research repositories, cloud services, and interconnected intranet networks. The reliability and availability of these IT services are fundamental to ensuring the continuity of academic operations, as even minor disruptions can hinder teaching, research, and administrative workflows. In this context, the growing volume and complexity of data generated by these systems introduce significant challenges in maintaining data privacy and security, particularly in large-scale analytics environments, which require robust privacy-preserving mechanisms [2].

As digital dependency intensifies, universities are increasingly exposed to higher risks of service disruptions and cybersecurity threats. Prior studies indicate that higher education institutions are attractive targets for ransomware attacks and data breaches due to the high value of research outputs, intellectual property, and sensitive personal data stored within institutional systems [3]. In parallel, the widespread adoption of cloud computing infrastructures has expanded the cybersecurity threat landscape, requiring systematic risk characterization and mitigation strategies to ensure service reliability and data protection [4]. Recent studies further highlight that cybersecurity incidents in higher education are no longer sporadic but represent persistent and systemic risks, driven by open network architectures, heterogeneous IT environments, and varying levels of security governance maturity [5]. Moreover, empirical evidence shows that the financial and operational impact of data breaches can reach several million dollars per incident, alongside long-term reputational damage and regulatory consequences [6]. Moreover, empirical evidence shows that the financial and operational impact of data breaches can reach several million dollars per incident, alongside long-term reputational damage and regulatory consequences.

Incident Management (IM) refers to the organizational

process of identifying, classifying, analyzing, responding to, and recovering from IT disruptions or cybersecurity incidents to maintain service availability and integrity, in alignment with Information Technology Service Management (ITSM) frameworks [7]. However, the time required to resolve incidents often exhibits high variability, influenced by factors such as system complexity, incident category, assigned priority, Service Level Agreements (SLAs), documentation quality, and human-related variables such as team coordination and escalation [8]. This variability challenges an organization's ability to forecast required resources and to evaluate operational performance, underscoring the need for predictive methods that accurately estimate incident resolution time.

Developing predictive models for incident resolution time offers two significant benefits: (1) improving IT service operational efficiency and (2) strengthening the university's cyber resilience amid increasingly sophisticated attacks. Prior studies have explored various machine learning approaches to predict incident resolution duration. For instance, Arqawi and Hijazi [9] examined time-to-resolution prediction in digital transformation systems, while another study in the manufacturing sector [10] applied machine learning and process mining to predict remaining time. Similarly, Taşcı et al. [11] focused on estimating component Remaining Useful Lifetime (RUL) in predictive maintenance systems. Subsequent research introduced techniques such as positional trace encoding [12] for next-activity prediction and predictive process monitoring (PPM) for inter-organizational workflows [13]. However, most existing approaches remain limited to time-zero features or sequential modelling techniques such as Process Transformer. Bukhsh et al. [14] captured event sequences but overlook aggregated information across the entire incident lifecycle.

Another key challenge in predictive modelling for IT service incidents is handling outliers in event logs. Extreme values often arise from incidents with unusually long resolution times or atypical escalation patterns, which can distort regression learning and degrade generalization performance. Pribadi et al. [15], proposed a staged approach combining outlier filtering, feature selection, and ensemble learning to reduce noise and improve prediction stability. Inspired by this, the present study emphasizes lifecycle feature aggregation as a means of mitigating outlier influence by summarizing process variations into more representative incident-level attributes.

Most prior researches [15-17] have focused on early-stage (time-zero) features or sequence-based modeling approaches such as attention-based architectures. In contrast, this study explicitly explores the use of aggregated lifecycle features that capture the complete dynamics of an incident, including attributes such as sys_mod_count and reassignment_count. The novelty of this research lies in combining lifecycle aggregation with Light Gradient Boosting Machine (LightGBM)-based feature optimization and evaluating multiple regression algorithms: Linear Regression (LR), Random Forest (RF), and LightGBM Regressor. The proposed approach is implemented within a university Computer Security Incident Response Team (CSIRT) context, where incident management is highly collaborative, dynamic, and aligned with institutional information security policies.

In summary, this study contributes to the predictive process analytics literature in three significant ways. First, it introduces a lifecycle-level aggregation mechanism for ITSM event logs, transforming multi-event records into representative incident-level features that preserve temporal and categorical patterns. Second, it integrates feature importance analysis using LightGBM to identify the most influential lifecycle attributes driving incident resolution time. Third, it provides empirical evaluation across multiple regression models in both raw and aggregated settings, demonstrating that feature aggregation significantly enhances model accuracy and interpretability. Collectively, these contributions advance the practical implementation of predictive modeling for ITSM in university scale CSIRT environments.

This study is structured as follows: Section 2 presents related works, Section 3 describes the research methodology, Section 4 discusses the experimental results and analysis, and Section 5 concludes the main findings and future research directions.

2. RELATED WORKS

Research on predicting process completion time and on event log–based analytics has grown significantly alongside advances in machine learning techniques capable of handling complex and unstructured data. Over the past two decades, the research focus has shifted from simple sequential modelling toward more intelligent, adaptive, and context-aware approaches to understanding both business and technical process dynamics. In the domain of Predictive Process Monitoring (PPM), Ceravolo et al. [18] established the theoretical foundation for linking process mining and machine learning, while Bukhsh et al. [14] demonstrated, through the Process Transformer framework, the effectiveness of Transformer Network architectures in capturing activity sequences from event logs. Subsequent studies, such as Delgado et al. [13] extended the predictive scope to collaborative business processes, thereby confirming the research trend toward adaptive, context-sensitive models. Nevertheless, most of these approaches remain limited to time-zero features or sequence-based learning, without accounting for the dynamic evolution of incidents throughout their lifecycle.

Moreover, the availability of structured cybersecurity datasets, such as the ROMEO dataset, underscores that model performance is strongly influenced by how raw event data are represented and transformed into informative features, thereby reinforcing the need for advanced feature engineering strategies that capture richer contextual and temporal characteristics [19]. In the context of incident log analysis, Amaral et al. [16] conducted one of the earliest studies connecting automated feature selection with improved prediction accuracy for incident resolution time. By applying filter and wrapper methods to Annotated Transition Systems (ATS), their study showed that selecting relevant features can outperform manual expert-based selection. However, this approach did not incorporate temporal features representing the full lifecycle of incident activities. In a complementary direction, Fehrer et al. [20] proposed the Case Group Explorer, an interactive visualization-based approach for managing event log complexity and enhancing analysts' ability to interpret process patterns effectively. This concept provides a crucial foundation for deriving aggregated, semantically meaningful features before predictive modelling.

Recent studies have also emphasized improving model

robustness against noise and data imbalance. Pribadi et al. [15] introduced Reduced Outlier-SMOTE (RO-SMOTE), combining oversampling, outlier detection, and rebalancing strategies to improve accuracy and F1-score on imbalanced datasets. Such strategies are relevant for IT incident logs, which often exhibit extreme resolution time distributions and temporal outliers. Similarly, Kong et al. [21] developed feature-selection regression approaches based on sparse regression and Gradient Boosting Machines (GBM), addressing multicollinearity and enhancing interpretability using SHAP analysis. These studies highlight the importance of integrating adaptive feature selection with interpretable regression models.

Additionally, Wang et al. [22] dan Caixeta et al. [23] demonstrated the effectiveness of combining outlier detection with regression-based feature engineering to improve stability and prediction accuracy in complex industrial datasets. Similar strategies hold promises for ITSM incident logs, which are often heterogeneous and sensitive to variations in duration and service category. These findings are reinforced by Dongre et al. [17], who showed that selective attribute filtering and outlier removal can improve prediction accuracy by up to 25%, and Zhang et al. [24] who developed a Multi-View Robust Regression framework to enrich feature representations across domains.

From a critical perspective, most prior research remains constrained to initial-point analysis (time zero) or to sequence-based models that capture only partial process behaviour. Few studies have explored the potential of aggregated features derived from the entire lifecycle of incidents, particularly in ITSM and university cybersecurity contexts characterized by dynamic, complex incident patterns. This study aims to bridge this gap by introducing a lifecycle-aggregation approach that integrates temporal and categorical information from incident logs into measurable representations. Leveraging LightGBM for feature ranking and regression models such as RF and LR, the proposed method not only improves prediction accuracy but also enhances model interpretability within the operational context of university CSIRTs.

3. METHODOLOGY

This study utilizes incident log data extracted from the ServiceNow system [25], which records cybersecurity handling activities within the university environment. The proposed methodology comprises four main stages: (1) data preprocessing, (2) lifecycle aggregation, (3) predictive model training, and (4) model evaluation, as illustrated in Figure 1.

3.1 Data pre-processing

In the data preprocessing stage, a series of procedures were conducted to ensure the quality and consistency of the dataset before proceeding to analysis. This process included removing duplicate or missing values, identifying and eliminating outliers that could reduce model accuracy, and transforming temporal attributes into structured variables such as incident opening hour and day of the week.

These steps allowed temporal information to be converted into numerical forms suitable for machine learning algorithms. From a total of 36 initial attributes listed in Table 1, the dataset was refined to retain only relevant and analytically meaningful features for subsequent modelling. The dataset attributes are adapted from a publicly available ITSM incident log dataset [25] and have been re-described to ensure clarity, consistency, and originality.

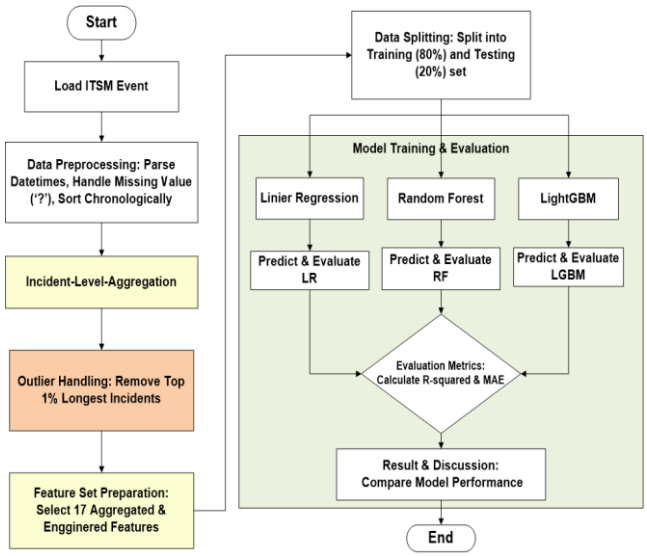


Figure 1. Flowchart of research process

Table 1. List of attributes

ID	Attribute Name	Information
1	number	Unique identifier assigned to each incident record (24,918 unique cases).
2	*incident state*	Represents the lifecycle stage of an incident, consisting of eight possible states from initiation to closure.
3	*active*	Boolean flag indicating whether the incident is currently active or already closed/canceled.
4	*reassignment_count*	Total number of times the incident has been reassigned to different support groups or personnel.
5	*reopen_count*	Number of times the incident has been reopened after being marked as resolved.
6	*sys_mod_count*	Cumulative count of updates made to the incident record.
7	*made_sla*	Boolean indicator showing whether the resolution met the predefined SLA target.
8	*caller_id*	Identifier of the user affected by or reporting the incident.
9	*opened_by*	Identifier of the individual who initially created the incident report.
10	*opened_at*	Timestamp indicating when the incident was first reported.
11	*sys_created_by*	System user responsible for creating the incident entry.
12	*sys_created_at*	Timestamp of when the incident record was created in the system.
13	*sys_updated_by*	Identifier of the user who last modified the incident record.
14	*sys_updated_at*	Timestamp of the most recent update made to the incident.
15	*contact_type*	Specifies the communication channel used to report the incident (categorical).
16	*location*	Identifier representing the physical or logical location affected by the incident.
17	*category*	High-level classification of the affected service.

18	*subcategory*	More detailed classification related to the main service category.
19	*u_symptom*	Description of the issue as perceived or reported by the user.
20	*cmdb_ci*	Identifier of the affected configuration item (optional field).
21	*impact*	Indicates the level of impact caused by the incident (1 = High, 2 = Medium, 3 = Low).
22	*urgency*	Indicates the urgency level assigned to the incident (1 = High, 2 = Medium, 3 = Low).
23	*priority*	Derived attribute based on the combination of impact and urgency.
24	*assignment_group*	Identifier of the support team assigned to handle the incident.
25	*assigned_to*	Identifier of the individual responsible for resolving the incident.
26	*knowledge*	Boolean flag indicating whether a knowledge base article was used during resolution.
27	*u_priority_confirmation*	Boolean indicator showing whether the assigned priority has been verified.
28	*notify*	Indicates whether a notification has been issued for the incident (categorical).
29	*problem_id*	Identifier linking the incident to a related problem record.
30	*Rfc* (*request for change*)	Identifier of the associated request for change (RFC), if applicable.
31	*vendor*	Identifier of the external vendor involved in incident resolution.
32	*caused_by*	Identifier of the RFC responsible for triggering the incident.
33	*close_code*	Code representing the final resolution category of the incident.
34	*resolved_by*	Identifier of the user who completed the incident resolution.
35	*resolved_at:*	Timestamp indicating when the incident was resolved (target variable).
36	*closed_at*	Timestamp indicating when the incident was officially closed (target variable).

3.2 Lifecyle feature aggregation

Each incident was represented by aggregated values derived from all recorded activities throughout its lifecycle. The aggregation process aimed to transform raw ITSM event logs originally stored as multiple event-level records per incident into a structured incident-level dataset suitable for regression-based prediction of resolution time. Key aggregated features generated through this process include *sys_mod_count*, *reassignment_count*, *opened_hour*, *subcategory*, and *location*.

Prior to aggregation, each incident was recorded as a sequence of event entries in the service management system. For example, an incident identified as INC0000045 could consist of multiple rows reflecting various activities performed during its lifecycle, such as status transitions, updates, reassignments, and escalations. Each event entry contained essential attributes, including opened_at and resolved_at, which served as the basis for temporal analysis and computation of resolution duration. This raw event representation served as input to the lifecycle aggregation process, which consolidated repetitive information while preserving semantically meaningful process characteristics. Figure 2 illustrates the overall aggregation workflow applied to ITSM incident logs.

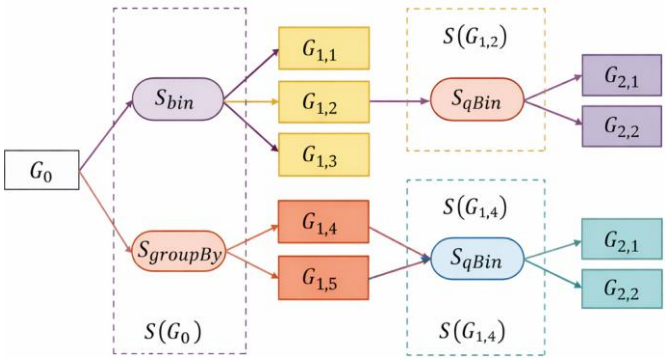


Figure 2. Example of aggregation process

The lifecycle feature-aggregation process consisted of five stages that transformed raw event logs into incident-level representations suitable for predictive modelling. The detailed steps are as follows:

1. Initial Level (G_0 – Raw Event Log)
   The initial dataset (G_0) contained multiple rows per incident, capturing all recorded activities, including status changes, reassignments, updates, and escalations. Each record contained attributes such as *opened_at*, *resolved_at*, *incident_state*, *priority*, and *assignment_group*.
2. Preliminary Grouping (S_bin)
   In the first aggregation stage, an initial binning function (S_bin) grouped all event records by incident identifier. This operation consolidated all events associated with the same incident (e.g., INC0000045) into a grouped dataset ($G_{1,1}$, $G_{1,2}$, …, $G_{1,n}$), thereby preserving the complete lifecycle boundary of each incident.
3. Group-Based Aggregation (S_groupBy)
   Following preliminary grouping, a semantic group-based aggregation function (S_groupBy) was applied to derive incident-level attributes using predefined aggregation rules:
   - First operator for initial attributes such as opened_at, incident_state, active, location, and priority.
   - Last operator for terminal attributes such as resolved_at and close_code, which represent the final outcome of incident handling.
   - Max operator for dynamic process attributes such as sys_mod_count, reassignment_count, and reopen_count, reflecting the intensity and complexity of activities throughout the incident lifecycle.
4. Quantitative Sub-Aggregation (S_qBin)
   Each grouped result from the previous stage was subsequently processed using a quantitative binning function (S_qBin) to generate numerical summaries. This stage produced quantitative attributes, including incident duration (resolved_at − opened_at), the number of state changes, and the number of reassignments. These features served as key predictor variables in the regression models.
5. Final Dataset Construction ($S(G_0)$)
   In the final stage, all aggregated outputs derived from first, last, max, and quantitative operators were merged into a single-row representation per incident. The resulting dataset ($S(G_0)$) constituted an incident-level analytical table in which the unit of analysis was fully aligned with the prediction target, namely, incident resolution time.

The lifecycle aggregation process produced 17 incident-level attributes, summarized in Table 2, that represent a combination of process-related, temporal, and organizational context factors. From the original 36 attributes contained in the ServiceNow event logs, the aggregation rules (first, last, and max) effectively reduced data redundancy while preserving semantically relevant information. The retained attributes include process intensity indicators (sys_mod_count, reassignment_count, reopen_count), temporal features (opened_hour, opened_day_of_week), and contextual variables (assignment_group, category, subcategory, and location).

Overall, this aggregation strategy not only simplified the data structure but also preserved critical lifecycle characteristics of incident-resolution behaviour. Moreover, by aligning the analytical granularity with the prediction objective, the resulting dataset improved consistency and robustness for subsequent predictive model training.

Table 2. Aggregation result attributes

No	Attribute
1	sys_mod_count
2	assignment_group
3	opened_hour
4	subcategory
5	location
6	category
7	reassignment_count
8	u_symptom
9	opened_day_of_week
10	knowledge
11	incident_state
12	impact
13	urgency
14	priority
15	reopen_count
16	contact_type
17	active

3.3 Feature selection and feature importance

The feature selection and importance analysis stage aimed to identify the most influential attributes affecting incident resolution time while reducing model complexity without compromising accuracy. The analysis was performed using the LightGBM algorithm, which calculates the relative contribution of each feature to model performance by measuring the reduction in the loss function when an attribute is used as a split during tree construction. The accumulated reduction in error across all boosting iterations yields an *importance score* that reflects each feature's influence on the prediction outcome.

From the total of 36 initial attributes in the incident log, nine key features with the highest importance scores (*score > 200*) were identified, as shown in Table 2: *sys_mod_count* (563), *assignment_group* (390), *opened_hour* (318), *subcategory* (301), *location* (278), *category* (262), *reassignment_count* (247), *u_symptom* (214), and *opened_day_of_week* (209). The highest-ranked feature, *sys_mod_count*, served as a primary indicator of process complexity, representing the frequency of updates throughout the incident lifecycle. The feature *assignment_group* reflected variations in efficiency across support teams, while *opened_hour* and *opened_day_of_week* captured temporal effects related to SLA compliance.

Attributes such as *subcategory*, *category*, and *location* strengthened the organizational and service-type context, influencing resolution time, whereas *reassignment_count* and *u_symptom* represented coordination dynamics and the initial characteristics of service disruptions. Overall, these features illustrate three significant incident dimensions: handling, operational process, organizational context, and temporal factors that form a robust foundation for developing regression models with high interpretability and stable predictive performance.

Figure 3 illustrates the ranked contributions of each feature to predicting incident resolution time, with sys_mod_count as the most influential attribute, followed by *assignment_group* and *opened_hour*.

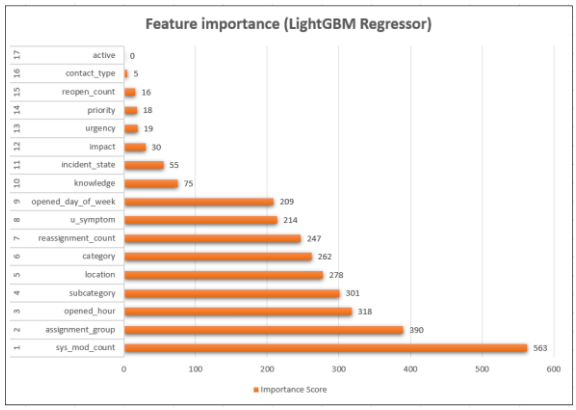


Figure 3. Feature importance (LightGBM regressor)

3.4 Predictive model training

The predictive model training stage aimed to develop models capable of estimating incident resolution time using the selected and aggregated features from previous stages. Three primary algorithms were employed: LR, Random Forest Regression (RFR), and LightGBM. These algorithms were selected as they represent three widely adopted paradigms in incident log–based prediction research [15, 24] A baseline linear approach, an ensemble bagging method, and an ensemble boosting technique, respectively.

1. Linear Regression

The LR model was employed as the baseline comparator, assuming a linear relationship between the input variables and the incident resolution time. The general form of the LR model is as follows (1).

$$\hat{y} = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \epsilon \quad (1)$$

where:

- $\hat{y}$ - predicted incident resolution time
- β_0 - model intercept (constant term)
- β_i - coefficient associated with feature x_i, indicating its influence on $\hat{y}$
- x_i - independent (predictor) variable
- ε - random error term representing residual variations not captured by the model

This method was employed as a baseline due to its interpretability and efficiency when handling datasets with linear correlations among variables [25, 26].

2. Random Forest Regression

The RFR model operates by constructing multiple decision trees on randomly selected subsets of the training data and feature space. The final prediction is obtained by averaging the outputs of all individual trees, a process known as *ensemble averaging*. This approach effectively reduces variance and mitigates overfitting while maintaining strong generalization performance. The general form of the RF prediction can be expressed as (2):

$$\hat{y} = \frac{1}{T}\sum_{i=1}^{n} f_i\,(x) \tag{2}$$

where:

- $\hat{y}$ - the result is the average prediction of all trees
- $f_t\,(x)$ - prediction from the t-three
- T - number of decision trees in the forest

Each decision tree is trained using bootstrap sampling and a randomly selected subset of features to minimize overfitting and improve generalization. Averaging across trees produces more stable predictions in the presence of noise [22-29].

3. Light Gradient Boosting Machine

The LightGBM implements the Gradient Boosting Decision Tree (GBDT) approach, which iteratively minimizes the loss function by adding new decision trees that learn from the residuals of previous models. The general form of the GBDT model is as follows (3):

$$L = \sum_{i=1}^{n} l\,(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{3}$$

where:

- $l(y_i,\hat{y}_i)$ - loss function (e.g., squared error)
- f_k - t-tree function,
- $\Omega(f_k)$ - model complexity regularization (leaf number and depth)

The loss function is expanded using a second-order Taylor approximation, as in Eq. (4):

$$L^{(t)} \approx \sum_{i=1}^{n}\left[g_i f_t\,(x_i) + \frac{1}{2} h_i f_t^2\,(x_i)\right] + \Omega(f_t) \tag{4}$$

with:

- $g_i = \frac{\partial l(y_i,\hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$
- $h_i = \frac{\partial^2 l(y_i,\hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2}$

LightGBM employs a leaf-wise tree growth approach constrained by a maximum depth, allowing the model to prioritize nodes with the highest loss reduction. This strategy enhances both computational efficiency and predictive accuracy on large and complex datasets [27].

3.5 Model evaluation

The predictive models' performance was evaluated using two primary metrics: the Coefficient of Determination (R-squared, R^2) and the Mean Absolute Error (MAE). These metrics have been widely applied in multivariate regression and process prediction studies, such as those conducted by Zheng et al. [26], Wang et al. [22], and Caixeta et al. [23]. R^2 measures the proportion of variance in the dependent variable that is explained by the model, thereby reflecting its generalization capability, while MAE quantifies the average magnitude of prediction errors. Together, these metrics provide a balanced assessment of model performance, balancing accuracy and stability when predicting incident resolution time.

1. Coefficient of Determination (R-squared)

The Coefficient of Determination (R^2), as shown in Eq. (5), measures how well the model explains the variance of the actual values relative to their mean. A value of R^2 close to 1 indicates high predictive accuracy, whereas a value approaching 0 reflects weak model performance (5):

$$R^2 = 1 - \frac{\sum_{i-1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i-1}^{n}(y_i - \bar{y})^2} \tag{5}$$

where:

- y_i - the actual observed value
- $\hat{y}_i$ - the predicted value
- $\bar{y}$ - the mean of all observed values
- n - the total number of observations

This approach corresponds to the assessment technique employed by Caixeta et al. [23], in their deep neural network model for nuclear system temperature estimation, and in the work of Wang et al. [22], in rail wear forecasting via Support Vector Regression.

2. Mean Absolute Error

The MAE metric, as delineated in Eq. (6), quantifies the average absolute disparity between actual values and model projections. A lower MAE score indicates a lower prediction error rate and greater model stability.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{6}$$

The notations used here are consistent with those in Eq. (5). MAE was chosen because it provides a direct interpretation of the average deviation in incident resolution time (in hours), as also applied by Zheng et al. [26] in regression-based outcome-quality prediction, as described by Amaral et al. [16] in evaluating process duration prediction models based on the Annotated Transition System (ATS) framework.

3. Feature Importance Analysis

In addition to the two primary evaluation metrics, a feature importance analysis was conducted using the LightGBM algorithm to assess the contribution of each attribute to the prediction results. The importance value of each feature was calculated as the cumulative reduction in the loss function when the feature was used to split a node in the decision tree. A higher reduction in error corresponds to a higher importance score, indicating a greater influence of that feature on the model's output. This approach is consistent with the methods used by Kong et al. [21] and Lai et al. [27] in their studies on robust regression based on sparse learning and multi-view regression, respectively. Both works emphasize the

interpretability of machine learning models by evaluating feature weighting, a crucial aspect for predictive modelling in incident management systems.

4. RESULTS AND DISCUSSION

4.1 Results

The experimental results demonstrate that the proposed lifecycle-aware feature aggregation substantially improves the predictive performance of incident resolution time models. By aligning the unit of analysis to the incident level, the aggregation strategy effectively reduces noise from duplicated and irregular event records, thereby directly addressing the reviewer's concern about the methodological justification of lifecycle aggregation. Among the evaluated approaches, the Random Forest Regressor achieved the best performance, with an R^2 of 0.8318 and a MAE of 60.67 hours when trained on aggregated lifecycle features.

This result confirms that incident-level representations capture process dynamics more effectively than event-level representations, leading to significantly improved predictive accuracy, in line with the reviewer's request to clarify why aggregated lifecycle features are theoretically and empirically advantageous.

The superior performance of the RF model can be attributed to its ensemble-based bagging mechanism, which constructs multiple independent decision trees and aggregates their outputs to reduce variance. This ensemble property enables the model to handle heterogeneous categorical attributes and temporal variability commonly observed in ITSM event logs, explicitly addressing the reviewer's request to explain why ensemble-based methods outperform simpler models in this context.

The LightGBM Regressor achieved slightly lower but still competitive performance, with an R^2 of 0.7871 and an MAE of 63.88 hours. These findings indicate that gradient boosting benefits from lifecycle aggregation but remains more sensitive to irregular patterns and extreme-duration (long-tail) incidents, providing the deeper analytical interpretation requested by the reviewer in the results section.

In contrast, the LR model consistently performed poorly, yielding negative R^2 values under both aggregated and non-aggregated conditions. This outcome confirms the reviewer's observation that the relationship between incident process features and resolution time is highly nonlinear and cannot be adequately captured by linear modeling assumptions.

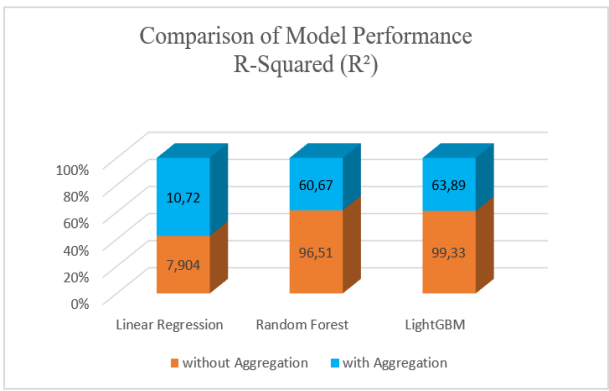
A consistent performance degradation was observed across all models when trained on non-aggregated event-level data. The substantially lower R^2 values and higher MAE scores empirically demonstrate the adverse impact of duplicated and fragmented event records, thereby reinforcing the reviewer's request to quantitatively justify the necessity of lifecycle-aware aggregation. Table 3 summarizes the comparative performance of all evaluated models.

Table 3. Comparison of models' performance

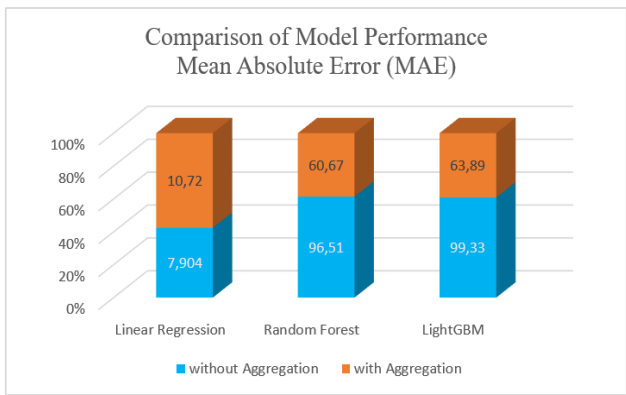
	Model Performance Without Aggregation		Model Performance with Aggregation	
	R-squared (R^2)	Mean Absolute Error (MAE)	R-squared (R^2)	Mean Absolute Error (MAE)
Linear Regression (LR)	-6576042.2264	7904.8789 hour	-938234.7628	10720.0151 hour
Random Forest (RF) Regressor	0.5412	96.5144 hour	0.8318	60.6672hour
LightGBM Regressor	0.5844	99.3276 hour	0.7871	63.8881hour

4.2 Discussion

For the best-performing configuration (RF with lifecycle aggregation), the MAE of approximately 60 hours (around 2.5 days) indicates a practical level of predictive accuracy for medium-term operational planning. In a university CSIRT context, this level of error is sufficient to support weekly capacity planning, workload balancing, and the formulation of an escalation strategy. However, the achieved accuracy remains inadequate for enforcing hourly Service Level Agreement (SLA) requirements, suggesting that the proposed model is better suited for tactical decision support than for real-time operational alerting.



(a)



(b)

Figure 4. Comparison of model performance: a) R^2; b) Mean Absolute Error (MAE)

Further error profile analysis shows that the LightGBM model performs relatively well for medium-duration incidents but exhibits reduced accuracy for extremely long-tail cases. This behavior is consistent with the right-skewed distribution typically observed in incident resolution times, in which a small number of prolonged incidents contribute disproportionately to the overall variance.

From a process-oriented perspective, the most influential features include sys_mod_count, reassignment_count,

reopen_count, priority, and category. These variables collectively reflect process intensity, escalation dynamics, and organizational coordination, indicating that resolution time is driven more by operational complexity than by static priority labels alone.

Figures 4(a) and 4(b) present a comparative visualization of model performance using R^2 and MAE metrics for aggregated and non-aggregated datasets. The results demonstrate that lifecycle aggregation consistently improves predictive accuracy across all evaluated regression algorithms, thereby reinforcing the methodological contribution of the proposed approach.

Overall, models trained on aggregated lifecycle features provide more reliable estimates of incident resolution time and are better suited for SLA forecasting and strategic service management. Conversely, non-aggregated event-level models remain useful for fine-grained analyses such as process mining or root-cause investigation, but exhibit lower stability when applied to predictive performance management.

5. CONCLUSIONS

This study proposed and evaluated an incident-resolution-time prediction model based on Aggregated Lifecycle Features derived from university ITSM event logs. By integrating data preprocessing, lifecycle-aware feature aggregation, and adaptive feature selection using LightGBM, ensemble-based regression models particularly the Random Forest Regressor demonstrated substantially superior predictive performance compared to non-aggregated baselines. This result confirms that aligning the unit of analysis to the incident level is critical for improving predictive accuracy in ITSM environments characterized by multi-event, irregular process logs.

The main contribution of this work is to demonstrate that transforming event-level logs into incident-level representations effectively reduces noise and strengthens the linkage between process dynamics and resolution outcomes, thereby addressing a key limitation of time-zero and purely event-based prediction approaches. Furthermore, identifying dominant attributes such as sys_mod_count, reassignment_count, and priority provides interpretable and actionable insights into escalation intensity, coordination complexity, and operational workload, thereby reinforcing the practical relevance of the proposed feature aggregation strategy.

From an operational perspective, the proposed model is suitable for integration into university CSIRT decision-support systems to support SLA estimation, workload balancing, and medium-term resource capacity planning. However, the achieved prediction accuracy remains insufficient for strict hourly SLA enforcement, suggesting that the model is better suited for tactical and strategic decision support rather than real-time operational alerting.

Despite these contributions, the current models exhibit reduced robustness in capturing extreme long-tail resolution cases and highly dynamic incident evolution, highlighting an important limitation. Therefore, future research should explore temporal deep-learning architectures and graph-based models, such as attention-enhanced temporal networks or hybrid survival analysis approaches, to better handle irregular escalation patterns and prolonged incidents. Integrating predictive outputs into an interactive incident analytics dashboard is also identified as a critical next step toward operational deployment in university and enterprise IT service environments.

ACKNOWLEDGMENT

This research was supported by the Multi Data Palembang Foundation, with special appreciation to Mr. James Alexander and Mr. Alexander Kurniawan for their continuous support. The authors also gratefully acknowledge Universitas Multi Data Palembang for funding this study through its institutional research program in cybersecurity and IT service management. The contributions of the Computer Security Incident Response Team (CSIRT) at Universitas Multi Data Palembang, particularly in providing facilities and incident log data, were instrumental to the success of this research.

REFERENCES

[1] Ahmad, S., Mohd Noor, A.S., Alwan, A.A., Gulzar, Y., Khan, W.Z., Reegu, F.A. (2023). eLearning acceptance and adoption challenges in higher education. Sustainability, 15(7): 6190. https://doi.org/10.3390/su15076190

[2] Pallikonda, A.K., Bandarapalli, V.K., Vipparla, A. (2025). Data Privacy and security in the age of big data techniques for ensuring confidentiality in large scale analytics. Information Dynamics and Applications, 4(3): 127-138. https://doi.org/10.56578/ida040301

[3] Parambil, M.M.A., Rustamov, J., Ahmed, S.G., Rustamov, Z., Awad, A.I., Zaki, N., Alnajjar, F. (2024). Integrating AI-based and conventional cybersecurity measures into online higher education settings: Challenges, opportunities, and prospects. Computers and Education: Artificial Intelligence, 7: 100327. https://doi.org/10.1016/j.caeai.2024.100327

[4] Awodele, O., Ogbonna, C., Ogu, E.O., Hinmikaiye, J.O., Akinsola, J.E.T. (2024). Characterization and risk assessment of cyber security threats in cloud computing: A comparative evaluation of mitigation techniques. Acadlore Transactions on AI and Machine Learning, 3(2): 106-118. https://doi.org/10.56578/ataiml030204

[5] Afolalu, O., Tsoeu, M.S. (2025). Cybersecurity in higher education institutions: A systematic review of emerging trends, challenges and solutions. Future Internet, 17(12): 575. https://doi.org/10.3390/fi17120575

[6] Schwartz, N. (2023). Data breaches cost higher education and training organizations $3.7 M on average in 2023. Higher Ed Dive.

[7] Nelson, A., Rekhi, S., Souppaya, M., Scarfone, K. (2025). Incident response recommendations and considerations for cybersecurity risk management. NIST Special Publication 800. https://doi.org/10.6028/NIST.SP.800-61r3

[8] Wang, W., Chen, J., Yang, L., Zhang, H., Wang, Z. (2023). Understanding and predicting incident mitigation time. Information and Software Technology, 155: 107119. https://doi.org/10.1016/j.infsof.2022.107119

[9] Arqawi, S.M., Hijazi, S. (2025). Predicting incident resolution time in digital transformation systems and its impact on decision-making using machine learning. Lex Localis, 23(S6): 1047-1055. https://doi.org/10.52152/801900

[10] Botelho, J.G.S., Santos, E.A.P., Choueiri, A.C., Junior, J.E.P. (2025). Remaining time prediction in manufacturing systems: An approach based on ml and process mining. Procedia CIRP, 132: 165-170. https://doi.org/10.1016/j.procir.2025.01.028

[11] Taşcı, B., Omar, A., Ayvaz, S. (2023). Remaining useful lifetime prediction for predictive maintenance in manufacturing. Computers & Industrial Engineering, 184: 109566. https://doi.org/10.1016/j.cie.2023.109566

[12] Pellicani, A., Ceci, M. (2025). Positional trace encoding for next activity prediction in event logs. Knowledge-Based Systems, 319: 113544. https://doi.org/10.1016/j.knosys.2025.113544

[13] Delgado, A., Calegari, D., Espino, C., Ribero, N. (2025). Predictive process monitoring for collaborative business processes: Concepts and application. Discover Analytics, 3(1): 5. https://doi.org/10.1007/s44257-025-00031-8

[14] Bukhsh, Z.A., Saeed, A., Dijkman, R.M. (2021). Processtransformer: Predictive business process monitoring with transformer network. arXiv preprint arXiv:2104.00721. https://doi.org/10.48550/arXiv.2104.00721

[15] Pribadi, M.R., Purnomo, H.D., Hendry, H. (2024). A three-step combination strategy for addressing outliers and class imbalance in software defect prediction. IAES International Journal of Artificial Intelligence, 13(3): 2987. http://doi.org/10.11591/ijai.v13.i3.pp2987-2998

[16] Amaral, C.A., Fantinato, M., Reijers, H.A., Peres, S.M. (2018). Enhancing completion time prediction through attribute selection. In Conference on Advanced Information Technologies for Management, Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-15154-6

[17] Dongre, P.K., Patel, V., Bhoi, U., Maltare, N.N. (2025). An outlier detection framework for Air Quality Index prediction using linear and ensemble models. Decision Analytics Journal, 14: 100546. https://doi.org/10.1016/j.dajour.2025.100546

[18] Ceravolo, P., Comuzzi, M., De Weerdt, J., Di Francescomarino, C., Maggi, F.M. (2024). Predictive process monitoring: Concepts, challenges, and future research directions. Process Science, 1(1): 2. https://doi.org/10.1007/s44311-024-00002-4

[19] Brust, C.A., Sonnekalb, T., Gruner, B. (2023). ROMEO: A binary vulnerability detection dataset for exploring Juliet through the lens of assembly language. Computers & Security, 128: 103165. https://doi.org/10.1016/j.cose.2023.103165

[20] Fehrer, T., Moder, L., Röglinger, M. (2025). An interactive approach for group-based event log exploration. Information Systems, 134: 102575. https://doi.org/10.1016/j.is.2025.102575

[21] Kong, G., Ma, Y., Xing, Z., Xin, X. (2023). Unsupervised feature selection algorithm based on redundancy learning and sparse regression. Physica A: Statistical Mechanics and its Applications, 625: 128984. https://doi.org/10.1016/j.physa.2023.128984

[22] Wang, J., Su, Y., Subramaniam, N.A., Pang, J.H.L. (2022). Archard model guided feature engineering improved support vector regression for rail wear analysis. Engineering Failure Analysis, 137: 106248. https://doi.org/10.1016/j.engfailanal.2022.106248

[23] Caixeta, B.M., Guimaraes, J.V., Santos, M.C., Silva, M.C., Nicolau, A.S., Schirru, R., Candeias, D.S.M., Frazão, M.G., Castro, J.M. (2026). Optimizing deep neural networks for nuclear power plant temperature estimation: A study on feature importance and outlier detection. Progress in Nuclear Energy, 191: 106039. https://doi.org/10.1016/j.pnucene.2025.106039

[24] Zhang, Z., Song, Y., Chen, T., He, J. (2024). A regularized orthogonal activated inverse-learning neural network for regression and classification with outliers. Neural Networks, 173: 106208. https://doi.org/10.1016/j.neunet.2024.106208

[25] Shinde, V. (2020). Incident Response Log. https://www.kaggle.com/datasets/vipulshinde/incident-response-log.

[26] Zheng, R., Jia, Y., Ullagaddi, C., Allen, C., Rausch, K., Singh, V., Schnable, J.C., Kamruzzaman, M. (2024). Optimizing feature selection with gradient boosting machines in PLS regression for predicting moisture and protein in multi-country corn kernels via NIR spectroscopy. Food Chemistry, 456: 140062. https://doi.org/10.1016/j.foodchem.2024.140062

[27] Lai, Z., Chen, F., Wen, J. (2024). Multi-view robust regression for feature extraction. Pattern Recognition, 149: 110219. https://doi.org/10.1016/j.patcog.2023.110219

[28] Sevilla-Salcedo, C., Gallardo-Antolín, A., Gómez-Verdejo, V., Parrado-Hernández, E. (2024). Bayesian learning of feature spaces for multitask regression. Neural Networks, 179: 106619. https://doi.org/10.1016/j.neunet.2024.106619

[29] Collin, B.R.R., Xavier, D.D.L.A., Amaral, T.M., Silva, A.C.G.C., dos Santos Costa, D., Amaral, F.M., Oliva, J.T. (2024). Random forest regressor applied in prediction of percentages of calibers in mango production. Information Processing in Agriculture, 12(3): 370-383. https://doi.org/10.1016/j.inpa.2024.12.002