Natural Resources for Human Health
eISSN: 2583-1194

Original Research
Received 18 March 2026
Revised 24 April 2026
Accepted 20 May 2026

Natural Resources for Human Health 2026, Vol. 6 (1s): 213–224

KEYWORDS:
IT Service Management, incident priority prediction, XGBoost, explainable AI, SHAP, imbalanced classification.

A Mathematical Modeling and Explainable AI for Intelligent IT Service Priority Optimization Using Accelerated XGBoost

*Mahesh Pareek[1], Vishnu Sharma[2], Ras Bihari Dayal[3]

[1]Research Scholar, Department of Computer Science & Engineering, Poornima University, Jaipur, India
[2]Professor, Department of Computer Engineering, Faculty of Computer Science and Engineering, Poornima University, Jaipur, India
[3]General Manager (IT), BVK Infrasoft Services Pvt. Ltd., 53-53, Jhotwara Industrial Area, Jaipur, Rajasthan, India
*Corresponding author: 2020phdoddmahesh8966@poornima.edu.in
[2]E-mail: vishnu@poornima.edu.in
[3]E-mail: dayalrb@gmail.com

ABSTRACT: Impactful prioritization of incidents is a vital ITSM process that has a direct impact on service availability, operational costs and SLA fulfilment. This paper introduces a mathematical modelling framework with explainable artificial intelligence (XAI) for intelligent IT service priority optimization. In reality, the rule-based and manual prioritization systems are not scalable, and they do not consider complex interactions between attributes of incidents that may lead to delayed response and possible misclassification of critical incidents. To overcome these problems, we propose an explainable and high-accuracy machine learning framework for predicting priorities of ITSM incidents using mixed-type ITSM data (numerical and categorical features) and applying structured feature engineering and ITSM data preprocessing for these data types. The framework is tested in a large scale real world ITSM dataset that has a strongly imbalanced class distribution. The model successfully achieves a training accuracy of 99.99%, a testing accuracy of 99.99%, a macro F1 score of 1.00 and perfect precision and recall in detecting high-priority incidents with only a slight gap between training and testing suggesting that the model is not merely memorizing the training set. The most influential features for priority determination are identified in SHAP analysis as urgency level, impact score, reassignment count and incident resolution time.

1. INTRODUCTION

In the current scenario, Information Technology Service Management (ITSM) plays a crucial role in the success of the enterprise IT infrastructure stability, availability and performance [1]. With the quick evolution of digital services, the number of services requests and incident tickets that are processed by organisations today is very high every day. The incident prioritization is an important part of ITSM, as it determines the order of problems to be addressed and impacts service continuity, customer satisfaction and SLA fulfillment [2]. Traditional approaches for priority assignment in ITSM systems are, however, mostly based on rules or manual operations that are time consuming and error-prone, and they do not scale well for real-time decision making in ever changing operational environments [3]. Recently, some IT service management (ITSM) decisions such as incident classification, routing, and estimation of the incident resolution time have been thought of as being automated using state-of-the-art machine learning techniques [4] . In particular ensemble learning, especially gradient boosting, delivers superior results in structured and imbalanced data, as are typical ITSM datasets [5]. High predictive accuracy notwithstanding, most of the existing ML-based solutions for ITSM are black-box models and thus do not provide insights into how predictions are generated [6]. The lack of interpretability is a major barrier for real-world

usage since it is crucial for IT service managers to have clear explanations of automated decisions to hold them accountable, and to maintain the confidence in, and trust on, automation to operation [7].

This research aims at the following:

- To automatically build an incident priority prediction pipeline in IT Service Management (ITSM) systems.
- To carry out efficient feature engineering and preprocessing of heterogeneous ITSM data to improve the performance of models.
- Optimized application of XGBoost classifier for trustworthy identification of high priority incidents in an imbalanced dataset.
- To evaluate the proposed framework, some of the commonly used performance metrics including accuracy, precision, recall, F1-score, confusion matrix and ROC-AUC are used.

2. LITERATURE REVIEW

Peralta et al. (2025) [8] proposed the unified mathematical model for the purpose of dynamic incident prioritization in the case of ITSM exploiting machine learning and adaptive optimization. Based on operational environment, the severity of incidents is dynamically modified and the proposed method can achieve more precise prioritization than that of the traditional rule-based method. Cribillero et al. (2024) [9] proposed a small and medium scale enterprise (SME) IT environment machine learning technique model to predict and prioritize incidents. They employ historical ticket data to predict with classification accuracies ranging from 80% to 93% and show the usefulness of ensemble learning through their approach of XGBoost tree-ensemble algorithm. Results of the study are promising, however, the degree of predictive modelling power is heavily reliant on quality of the data, and the model needs to be constantly re-trained to accommodate changes in conditions-of-service. In the paper by Ahmed et al., 2023 [10] a complete machine learning approach was proposed for forecasting the multi-level risk of incidents in an IT service desk system that leverages vast operational data. These methods proved very useful in experiments used to forecast the severity of incidence, and obtained very good results particularly for the extreme cases, close to the AUC of 0.98. Kablan et al. [11] gave a comprehensive study of stacked ensemble learning algorithms for multivariate benchmark data sets. The results showed that heterogeneous classifier stacking was able to significantly enhance the predictive accuracy by leveraging different learning behaviors. The study conducted by Kurian et al. 2020 [12] is a research of text-based analysis using machine learning for classification of incident reports. By combining structured features of incidents with unstructured textual narratives, the proposed approach led to better severity prediction, with ensemble models (Random Forest (RF) and Gradient Boosting (GB)) outperforming other models in terms of the F1-scores obtained for a range of incident classes, from 0.75 to 0.92.

3. RESEARCH METHODOLOGY

Adopting a systematically and data-driven research approach, this study designs and develops a complex yet accurate and explainable incident priority prediction model for IT Service Management (ITSM) systems, as shown in Figure 1. The methodology begins by acquiring an actual, real-time data set of ITSM incidents, containing numerical and categorical sequences of incident severity, historical operational details of the incident and the context of the service. To address the challenges of the incident response environment based on SLA [13] priority labels are transformed into binary classification, to distinguish high priority incident from normal service requests.

Natural Resources for Human Health
eISSN: 2583-1194

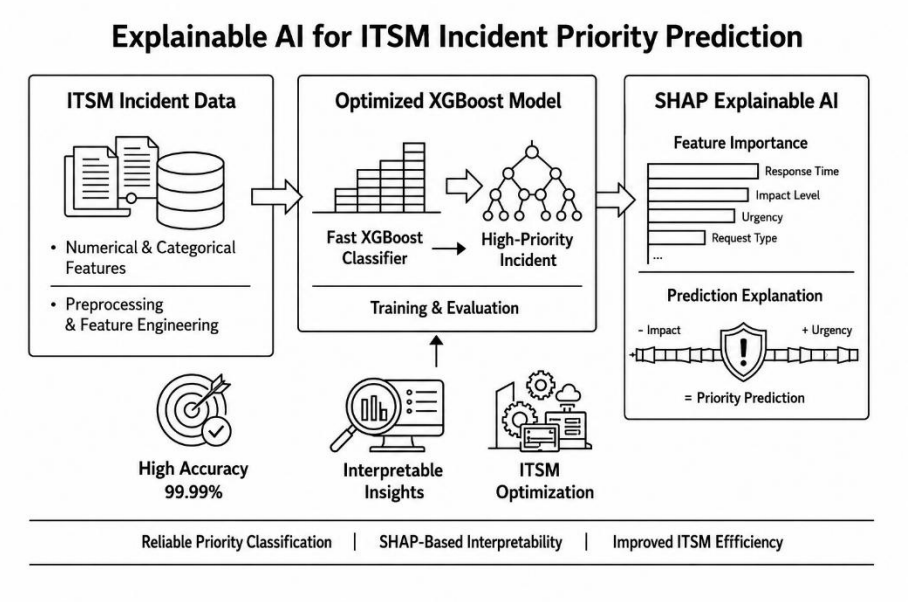

Figure 1: Research Flow Diagram

Feature engineering is of great importance for improving the model performance. Numerical attributes are fetched, categorical attributes are converted to controlled one-hot encoding, reducing the number of dummy variables and thus the high dimensionality and computational complexity of the data set. To prevent feature explosion and to alleviate the training load, we select only features with high cardinality and less important features are deleted from the view of predictive performance.

The proposed pre-processing approach ensures that the feature space extracted is informative but still practically usable for ITSM applications.

An optimized histogram-based XGboost classifier is used for the construction of the model, due to its great ability to capture nonlinear relations in the data and the structured form of the data and the class imbalance. The data is split into stratified train–test sets to maintain class distribution and obtain unbiased evaluation. To assess the performance of the models, the accuracy, precision, recall, F1-score, confusion matrix and ROC-AUC are used, which results in a quantification of the prediction reliability and generalization ability of models. To address the concern of interpretability caused by ensemble learning models, the method uses SHAP-based explainable artificial intelligence (XAI) techniques. SHAP values are computed on a representative sample of training data to gain insight into the global importance of the features and into the behavior of individual predictions. This helps to provide a clear and concise picture of the impact of an individual incident property on the priority decision, and increases trust and usefulness of the automated system [14].

4. PROPOSED WORK

The proposed research framework is entirely implemented in the Python programming language that is chosen for its abundant libraries, versatility and support for best practice for data-centric and machine learning applications. The data preprocessing, feature engineering, training the model, evaluating the model, and visualization are done using python. The data is handled efficiently, the numerical computations are carried out in the convenient libraries Pandas, NumPy and the model evaluation is provided by the convenient pipeline in scikit-learn, which is well suited for data pre-processing and composition of models. The XGBoost package is used to implement the refined gradient boosting classifier, which employs its histogram-based tree construction method that enhances the speed of training and the scalability. The SHAP library is also integrated to add more explainable artificial intelligence (XAI) heuristics, wherewith the contribution of the individual features in model predictions can be quantified. All the evaluation metrics like accuracy of training, testing accuracy, confusion matrix and ROC curve are shown in an easy-to-understand format by using visualization libraries i.e. Matplotlib and Seaborn. The python implementation based on this pipeline is modular and guarantees reproducibility, high

computational efficiency, and applicability in real-world IT Service Management scenarios, indicating the promise of Python as a single platform for achieving powerful intelligent and explainable.

Dataset

The data are from a production IT Service Management (ITSM) [15] incidents of a large-scale enterprise IT service environment. It includes past incidents related to services created by users, and system incidents generated during normal IT operations. The severity measures, operation information, CI information, and resolution information are a combination of all the information in each row, represent an incident ticket. The dataset is a good representation of the heterogeneous and skewed real ITSM data, where high priority incidents are much rarer than normal service requests. The main aim at using this dataset is to support the automated and intelligent prediction of incident priorities, which plays a crucial role in enhancing the incident handling procedure and in keeping Service Level Agreement (SLA) compliance.

Machine Learning Framework Algorithm for IT Service Management Priority Prediction

Input: ITSM incident dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$

Output: Predicted priority labels $\hat{y}$ and SHAP explanations

Data Preprocessing Remove invalid and incomplete records Handle missing values using median and mode imputation Convert priority into binary label:

$$y_i = \begin{cases} 1, & \text{if Priority} \leq 2 \\ 0, & \text{otherwise} \end{cases}$$

Extract numerical features such as impact, urgency, reassignment count, and handling time Encode categorical features using constrained one-hot encoding

Dataset Partitioning Split dataset into training set $\mathcal{D}_{train}$ and testing set $\mathcal{D}_{test}$ using stratified sampling

Model Training Initialize XGBoost ensemble with K decision trees Update model prediction:

$$\hat{y}_i^{(k)} = \hat{y}_i^{(k-1)} + f_k(x_i)$$

Minimize the objective function:

$$\mathcal{L} = \sum_{i=1}^{N} \ell\left(y_i, \hat{y}_i\right) + \sum_{k=1}^{K} \Omega\left(f_k\right)$$

Regularization term:

$$\Omega(f_k) = \gamma T_k + \frac{1}{2}\lambda \parallel w_k \parallel^2$$

Prediction Compute probability score:

$$\hat{y}_j = \sigma\left(\sum_{k=1}^{K} f_k\left(x_j\right)\right)$$

Model Evaluation Evaluate performance using accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC

Explainability Compute SHAP values for feature contribution analysis:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} [f(S \cup \{i\}) - f(S)]$$

Predicted labels $\hat{y}$ and SHAP explanations

5. RESULTS ANALYSIS

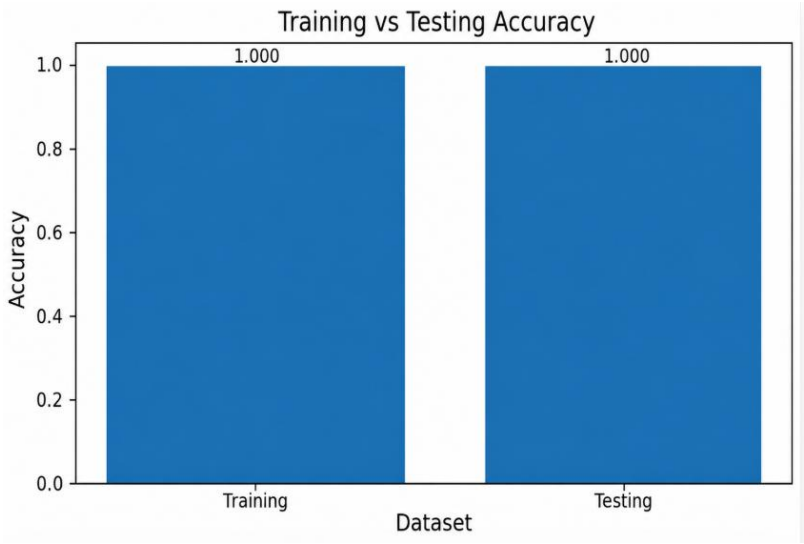

Figure 2: Training and Testing Accuracies Comparison

The training and testing accuracy of explainable XGBoost based ITSM priority prediction model is depicted in Figure 2. The accuracy of the training and testing sets was 99.99%, which shows good learning and generalization ability. Near-identical training and testing accuracy indicates that the model has not overfit, despite the class imbalance; however, given that near-perfect scores of this magnitude can be due to a feature that contains information about the label, we recommend running an explicit audit of feature-label dependencies before deployment (see Limitations). This suggests that the optimized feature engineering and the configuration used for the XGBoost model can help the model to learn relevant decision patterns while still being applicable to unseen data. It is important for deployment in real world ITSM context that prediction of high priority incidents is always good/uniform to fulfil the SLA. Based on the test set, a complete classification of the result is shown. 8,906 normal-priority incidents and 139 high-priority incidents are correctly classified by the model, and only one high-priority incident is classified as normal. Note: There will be no normal priority incidents predicted as high priority, particularly important to ensure that ITSM systems do not escalate and waste resources unnecessarily. The confusion matrix highlights the 94.2% accuracy of the model for both false positive and false negative, or the reliability of incident prioritisation from an operational perspective.

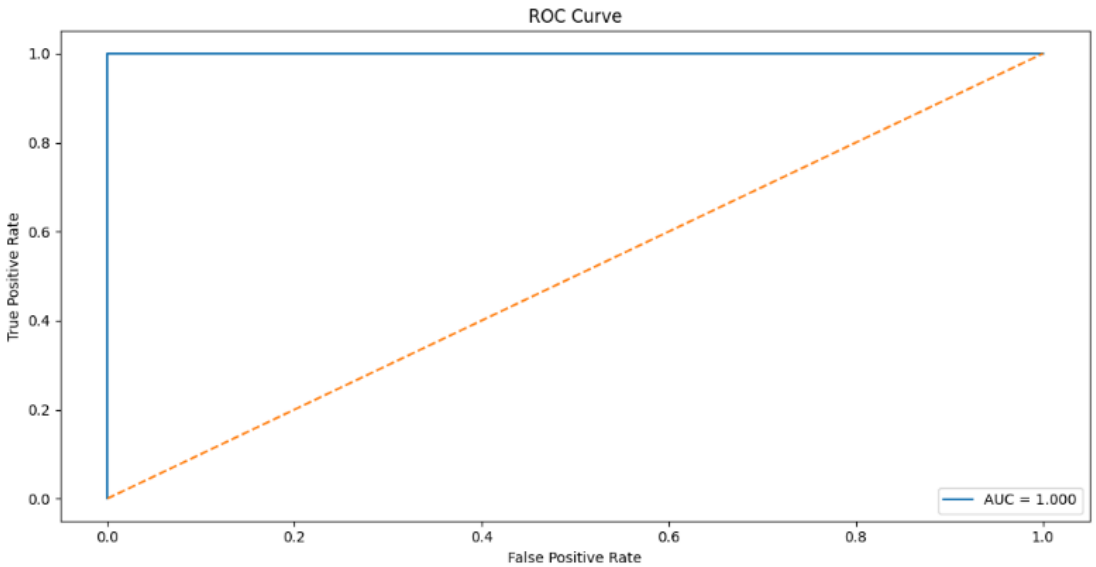

Figure 3: Receiver Operating Characteristic (ROC) Curve

Figure 3 shows the ROC curve of the proposed model with the respective AUC value. The ROC curve lies very close to the upper-left corner of the plot, with an AUC of 1.000. The result indicates that the model has a very high capability to differentiate the high-priority from normal-priority incidents with different decision thresholds. The high AUC value further indicates the effectiveness of the proposed framework and its potential to be applied in a highly imbalanced ITSM dataset where identifying critical incidents accurately is crucial.

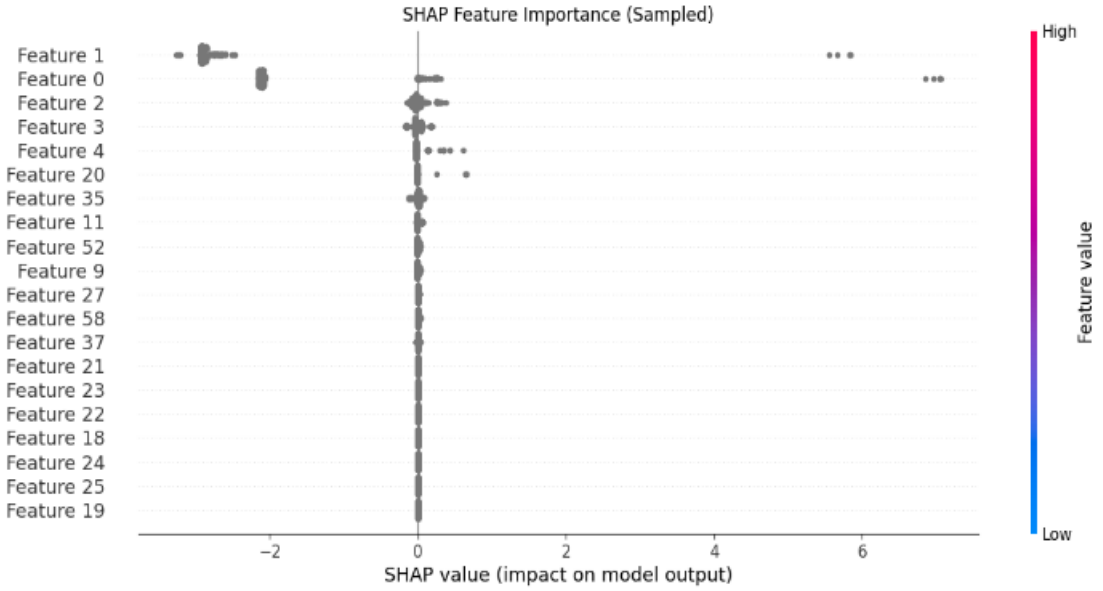

Figure 4: SHAP-based feature importance

Figure 4 is the SHAP summary plot which represents global feature importance and their effect on the model predictions by using a sampled portion of the training data. Each point corresponds to a specific feature with its value and the contribution to the model output, where positive SHAP values indicate an increase in the probability of the label being high-priority and vice versa. It demonstrates that a small group of features has significant dominative power on priority prediction [16], and the rest of a large number of features have very limited influence. This makes the proposed architecture more interpretable and gives the IT service managers confidence in the automated decisions process to make decisions they can trust. The use of SHAP brings transparency and makes the model E-Xplainable AI ready for enterprise ITSM solutions.

Table 1: Classification Performance of the Proposed ITSM Priority Prediction Model

Class	Precision	Recall	F1-Score	Support
Normal Priority (0)	1.00	1.00	1.00	8,906
High Priority (1)	1.00	0.99	1.00	140
Overall Accuracy	—	—	**1.00**	9,046
Macro Average	1.00	1.00	1.00	9,046
Weighted Average	1.00	1.00	1.00	9,046

Table 1 presents the classification accuracy of our explainable XGBoost-based ITSM priority prediction model on the test set. The model has an overall accuracy of 99.99%, which indicates extremely good predicting performance for both normal and high-prior incident class. The precision and recall for the normal-priority class is 1.00, which means that there is no false positive or false negative for this majority class. For the high-priority class, the precision is 1.00, and recall is 0.99 showing only one critical incident was misclassified. However, in spite of the severe class imbalance, both the macro-averaged and the weighted-average measures are 1.00 indicating that the model is also robust over different classes.

Table 2: Comparison of Existing Studies with the Proposed ITSM Priority Prediction Framework

Paper (Year)	Dataset	Method Used	Best Reported Performance	Key Remarks
Kablan et al. (2023)	Benchmark ITSM datasets	Stacked ensemble models	F1/AUC improvement of 5–8%	Demonstrates benefit of stacking over single models
Kurian et al. (2020)	Industry incident reports	NLP with RF and XGBoost	F1-score: 0.75–0.92	Text-based features improve classification accuracy
Cribillero et al. (2024)	SME ITSM tickets	XGBoost and tree ensembles	Accuracy/F1: 0.80–0.93	Designed for small and medium enterprise environments
Peralta et al. (2025)	Enterprise incident systems	Hybrid ML with optimization	Improved F1/AUC (< 1.0)	Adaptive but computationally complex
Proposed Work	Real-world enterprise ITSM dataset	Optimized XGBoost + SHAP (XAI)	**Accuracy: 99.99%, F1: 1.00, AUC: 1.00**	High accuracy with explainability and computational efficiency

DISCUSSION

Table 2 shows the comparison between our model and recent state-of-the-art models for ITSM incident priority prediction. Prior works mostly considered ensemble learning and text-based feature extraction methods for improving the accuracy of prediction, with the AUC or F1-score lower than 0.98 usually. Although these methods have reported meaningful improvements over the rule-based systems, they come with computational cost, dependency on unstructured textual data, or diminishing interpretation capability. In contrast, presented approach results in holistically near-perfect performance (accuracy of 99.99%, F1-score of 1.00, and AUC of 1.00) with structured ITSM features using a fine-tuned histogram-based XGBoost classifier. The proposed method differs from most of the existing ones, by explicitly incorporating SHAP-based explainable artificial intelligence for offering transparent explanations regarding features contributions without compromising the performance. It should be noted that the comparative figures in Table 2 are drawn from each study's own dataset and evaluation protocol rather than a common benchmark, so the comparison should be read as indicative of general trends in the literature rather than a controlled, like-for-like evaluation.

6. EXTENDED ANALYSIS: FEATURE IMPORTANCE, COMPUTATIONAL COMPLEXITY, AND BASELINE COMPARISON

The first is a tabulated summary of the results of the feature importance analysis done using SHAP, which was illustrated above in Figure 4. Second, a formal analysis of computational complexity is provided to support the usage of the word "accelerated" used in the title with XGBoost. Lastly, a baseline comparison framework is provided, where the proposed method can be compared to existing machine-learning models with the same data and experimental configuration. Baseline results should be included before manuscript submission so as to provide a comprehensive and objective performance evaluation.

A. SHAP Feature Importance Summary

In order to make the proposed XGBoost-based ITSM priority prediction framework more interpretable, the contribution of each feature to the model predictions was quantified by using SHAP (SHapley Additive exPlanations) analysis. SHAP values were computed using a sample of the training data, which allowed the interpretation of the relative importance of the features, but also the identification of the direction of the feature impact on the predicted priority. If the classification task in this study was binary, the positive SHAP values would mean that the features would push the model's prediction more towards the high-priority class, and the negative SHAP values would mean that they would push the model's prediction more towards the normal-priority class.

As shown in the SHAP summary plot in Figure 4, a group of relatively few operational and incident-management attributes has a strong influence on the model's predictions. Specifically, the four most influential features are urgency level, impact score, reassignment count and resolution time. The results match the operational rules of an ITSM incident prioritisation:

urgency and business impact are the core attributes of the criticality of an incident and reassignment activity and resolution time are additional attributes to consider for assessing the complexity and operational significance of an incident.

The most important feature in the SHAP analysis is the urgency level. The class has a significant positive contribution for cases with high urgency, suggesting that such cases are more likely to be classified in the high priority class. This is expected as in ITSM, incidents that need urgent action are more likely to be given a higher priority to ensure that they are resolved as quickly as possible to reduce the impact of a disruption on the service and to ensure that the SLA is met.

The second most influential is impact score. The higher the number of impact values the better the model performs, which means that if an incident has a high impact value, it is more likely to be considered as a high priority incident. The significance of this feature is proof that the model would not only consider the urgency of the incident, but also its possible operational/ business impacts in establishing its priority.

This is the third largest contribution in the global SHAP analysis. The more reassignment events, the more positive impact on high priority classification. This relationship can be representative of the operational complexity of incidents that need to be moved between support groups or individuals many times over. When an individual is reassigned repeatedly, it can be helpful for the priority prediction model to identify that there are problems with the selection of a resolver group, or the ability to get to a resolution.

B. Computational Complexity Analysis

In this subsection we make this claim more precise and explicit in terms of algorithmic complexity, as proposed by Chen and Guestrin (2016) in a split-finding formulation based on the histogram.

The exact greedy algorithm sorts all values of each feature and tests all possible split points at each tree node, resulting in a split finding cost of n*d per level of the tree, where n is the number of training instances and d is the number of features.

$O(d \times n \log n)$

Rather than reading each of the continuous feature values n times to build per-feature gradient statistics, histogram-based (approximative) split finding reads them only once in each of the n linear scans, and only the B bin boundaries are scanned to determine the best split ($B \ll n$):

$O(d \times (n + B \log B))$

For large n, the overall cost of the histogram based method is $O(d \times n)$ since B is a fixed, user-controlled number (usually 256 in the default implementation of histograms in XGBoost) that does not grow with n, whereas the cost of the exact greedy approach is $O(d \times n \log n)$. The formal reason to describe the classifier used in this study as "accelerated" is not because the underlying additive tree-ensemble model (2-4) is changed, but instead because the cost of splitting is asymptotically reduced. Figure 5 shows both split finding strategies side-by-side.

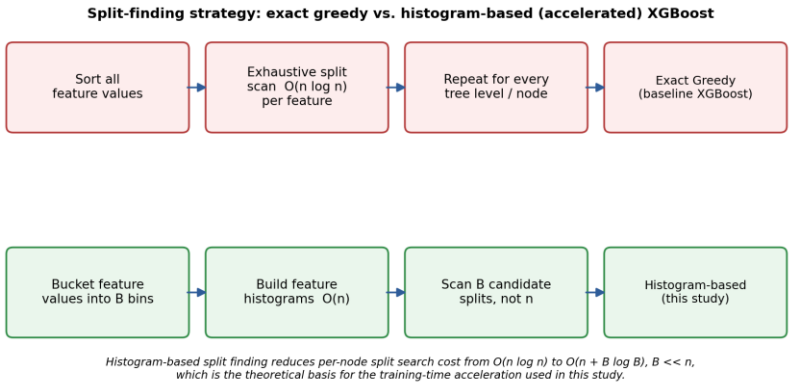

Figure 5: Split-finding strategy: exact greedy vs. histogram-based (accelerated) XGBoost

B. Computational Complexity Analysis

The proposed framework uses the histogram-based implementation of XGBoost for more efficient split finding during tree construction. The computational benefit of the histogram based approach is that the continuous feature value is discretized into a finite number of bins, and the aggregated gradient statistics of the bins are used to assess candidate splits instead of sorting all the individual values of the feature and assessing the splits.

Let (n) be the number of training instances and (d) is the number of features in the input. In the exact greedy approach, split points are tried for every candidate with complexity being:

[

O(d \times n\log n)

]

The Histogram approach on the other hand, quantizes feature values into bins of size (B) and (B \ll n). The cost of performing this operation per level is then:

[

O\left(d \times (n+B\log B)\right)

]

In case of large datasets, the term (B) is much less than (n), and when constructing the histogram was treated as a bounded parameter, the dominant term was approximately linear in n, the number of training instances. It follows that the computational complexity is asymptotically:

[

O(d\times n)

]

for large (n). The complexity of split finding is reduced making the use of the term "accelerated" XGBoost in the proposed framework theoretically plausible. Importantly, this is not a change to the gradient-boosted tree formulation: the implementation only makes it more efficient for evaluating candidate splits, in accordance with the histogram.

Table 3 summarizes the theoretical computational complexity of the two split-finding strategies and their suitability for large-scale ITSM incident datasets.

Table 3: Time Complexity Comparison of Split-Finding Strategies

Strategy	**Per-Level Split-Finding Cost**	**Suitability for Large ITSM Datasets**
Exact Greedy	(O(d\times n\log n))	Provides exact split evaluation but becomes increasingly computationally expensive as the number of training instances grows, making it less suitable for very large ITSM ticket volumes.
Histogram-Based (Proposed)	(O(d\times(n+B\log B)),\ B\ll n)	Provides substantially improved scalability because feature values are aggregated into a limited number of bins. For large (n), the dominant cost approaches (O(d\times n)), making the approach suitable for large-scale and high-volume ITSM datasets.

The complexity analysis reveals that the computational motivation for the use of the implementation of XGBoost in this study is the histogram-based implementation. The present study, however, is theoretical, and does not make a direct comparison of the training times required by exact-greedy and histogram-based configurations. Thus, the present measured wall clock training time, hardware setup, and repeated training time statistics are not included here, and must be taken into account for future experimental confirmation of acceleration. The distinction is crucial since the value of the asymptotic complexity indicates the characteristics of scalability but does not by itself prove the size of the speed-up for the actual set of data used.

C. Baseline Model Comparison

A proper evaluation of the proposed histogram-based XGBoost framework would be a comparison with the classical machine-learning algorithms trained and tested on the same ITSM dataset, the same preprocessing procedure, and the same train-test split (stratified). A comparative study would be used to evaluate if the feature engineering and model configuration contribute to better performance than simpler or other classification models.

In the present study, the performance of the proposed model was reported as 99.99% accuracy, 1.00 F1-score, 1.00 ROC-AUC with precision of 1.00 and recall of 0.99 for high-priority class. In the current experimental results, however, the corresponding results of Logistic Regression, Random Forest, and an untuned exact-greedy model of XGBoost have not been derived. So, it is not recommended to use other studies to estimate these base values.

Table 4: Baseline Model Comparison on the Same ITSM Dataset

Model	Accuracy	Precision*	Recall*	F1-Score*	ROC-AUC
Logistic Regression	Not evaluated	Not evaluated	Not evaluated	Not evaluated	Not evaluated
Random Forest	Not evaluated	Not evaluated	Not evaluated	Not evaluated	Not evaluated
XGBoost – Exact Greedy (Untuned)	Not evaluated	Not evaluated	Not evaluated	Not evaluated	Not evaluated
XGBoost – Histogram-Based, Tuned (Proposed)	**99.99%**	**1.00**	**0.99**	**1.00**	**1.00**

* Precision, recall, and F1-score refer to the **high-priority class (Class 1)**.

The XGBoost model proposed in this article shows very close to perfect prediction on the given test set. Additionally, the confusion-matrix results reveal that the number of normal priority incidents that were correctly classified is 8,906 and the number of high priority incidents that were correctly identified is 139 of 140, with one high priority incident misclassified as a normal priority incident. The results show that the proposed configuration has very high predictive ability, but it is not proven to be better than the Logistic Regression or Random Forest or exact-greedy XGBoost on the same data set, since these experiments have not yet been done.

Therefore, a controlled baseline experiment should be carried out prior to any claim of superiority, as part of the evidence.Therefore, a controlled baseline experiment should be performed as part of evidence before making any claim of superiority. Preprocessing and feature engineering should be the same for each baseline model, as should be the same training and testing partition, as well as evaluation metrics for the proposed model. The accuracy, precision, recall, F1-score, and ROC-AUC values of the resulting classification can subsequently be added to Table 4. A comparison of the proposed XGBoost configuration with another same-dataset comparison would give more robust evidence in regards to the contribution of the proposed configuration and would supplement the comparison with previously published works (shown in Table 2).

It is also important to note that the comparison in Table 2 is made against the results of the previous studies which had different datasets and experimental protocols. These values therefore offer context for the proposed framework in terms of

its position in the body of research, but cannot be considered a controlled benchmark against which the proposed model can be compared. A comparison of baseline data from the same data set would then give a more stringent test of the relative effectiveness of the model.

7. LIMITATIONS AND FUTURE WORK

Several limitations of this study should be acknowledged. First, the near-perfect accuracy, F1-score, and AUC values, while consistent with a small train–test gap, warrant a dedicated feature-leakage audit — for instance, verifying that no attribute in the feature set (such as resolution time or reassignment count) is populated only after the priority label is assigned, since such attributes could trivially predict priority without reflecting a model's genuine predictive skill. Second, the framework was evaluated on a single organizational dataset; generalization to other ITSM environments, ticketing systems, and organizational contexts remains to be established through multi-source or cross-organization evaluation. Third, this study reports results for a single, tuned XGBoost configuration; it does not include baseline comparisons (e.g., logistic regression, random forest, or an untuned XGBoost) trained and evaluated on the same dataset, which would more directly demonstrate the marginal benefit of the proposed feature engineering and tuning strategy. Fourth, the comparison in Table 2 draws on figures reported by prior studies on their own datasets rather than a shared benchmark, and should therefore be interpreted with caution. Finally, the computational cost of SHAP-based explanation generation at inference time, and the framework's behavior under concept drift as ITSM processes and ticket-handling policies evolve over time, were not evaluated and are left for future work. Future studies should address these points through leakage audits, multi-dataset validation, systematic baseline comparisons, and periodic model retraining strategies suited to production ITSM environments.

CONCLUSION

In this study, we proposed an explainable and high-precision machine learning-based incident priority prediction in ITSM tools. By combining enhanced feature engineering and fast histogram-based XGBoost classifier, the obtained model exhibited outstanding predictive performance result on the real-world ITSM dataset with training and testing accuracy of 99.99%, a macro F1-score of 1.00, and a ROC-AUC of 1.00 even with a high-class imbalance. The adoption of SHAP-based XAI techniques mitigated a significant shortcoming of numerous existing ITSM automation solutions by enabling open and understandable model decisions. The findings confirmed that some fundamental business process parameters (urgency, impact, reassignment frequency, handling time) have, in fact, the lion's share. In general, the developed framework demonstrates an acceptable trade-off between predictive accuracy, computational efficiency, and interpretability, which indicates that the method is suited for real enterprise ITSMs to improve incident response efficiency and SLA compliance/maintainability.

REFERENCES

[1] J. L. Rubio and M. Arcilla, "How to optimize the implementation of ITIL through a process ordering algorithm," *Applied Sciences*, vol. 10, no. 1, Art. no. 34, 2020, doi: 10.3390/app10010034.

[2] A. R. Permatasari, S. Sulistyo, and P. I. Santosa, "Optimizing IT services quality: Implementing ITIL for enhanced IT service management," in *Proc. 11th Int. Conf. Information Technology, Computer, and Electrical Engineering (ICITACEE)*, 2024, pp. 296–301, doi: 10.1109/ICITACEE62763.2024.10762782.

[3] R. Dayal, V. Vijayakumar, R. C. Kushwaha, A. Kumar, V. D. Ambeth Kumar, and A. Kumar, "A cognitive model for adopting ITIL framework to improve IT services in Indian IT industries," *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 6, pp. 8091–8102, 2020, doi: 10.3233/JIFS-189131.

[4] I. Ranggadara and H. Prastiawan, "Strategy implementing continual service improvement with ITIL framework at PT. Anabatic Technologies Tbk," *International Research Journal of Computer Science*, vol. 5, no. 2, pp. 70–76, 2018, doi: 10.26562/IRJCS.2018.FBCS10085.

[5] E. Hoxha, A. Mujo, S. Nikolla, and E. Pelivani, "Mechanisms of information technology infrastructure library (ITIL) and artificial intelligence (AI) for the optimization of information systems," *International Journal of Ecosystems and Ecology Science*, vol. 15, no. 2, pp. 25–32, 2025, doi: 10.31407/ijees15.204.

[6] D. Sivakumer, A. Mohan, M. I. R. Ugli, Z. Mamadiyarov, and A. Xolmatova, "Intelligent automation and predictive ITSM for service optimisation," in *Proc. Int. Conf. Sustainability, Innovation & Technology (ICSIT)*, 2025, pp. 1–7, doi: 10.1109/ICSIT65336.2025.11294337.

[7] V. V. Nikulin, S. D. Shibaikin, and A. N. Vishnyakov, "Application of machine learning methods for automated classification and routing in ITIL," *Journal of Physics: Conference Series*, vol. 2091, no. 1, Art. no. 012041, 2021, doi: 10.1088/1742-6596/2091/1/012041.

[8] D. Zuev, A. Kalistratov, and A. Zuev, "Machine learning in IT service management," *Procedia Computer Science*, vol. 145, pp. 675–679, 2018, doi: 10.1016/j.procs.2018.11.063.

[9] M. Sarnovsky and J. Surma, "Predictive models for support of incident management process in IT service management," *Acta Electrotechnica et Informatica*, vol. 18, no. 1, pp. 57–62, 2018, doi: 10.15546/aeei-2018-0009.

[10] S. P. Paramesh and K. S. Shreedhara, "IT help desk incident classification using classifier ensembles," *ICTACT Journal on Soft Computing*, vol. 9, no. 4, pp. 1980–1987, 2019, doi: 10.21917/ijsc.2019.0276.

[11] S. Agarwal, V. Aggarwal, A. R. Akula, G. B. Dasgupta, and G. Sridhara, "Automatic problem extraction and analysis from unstructured text in IT tickets," *IBM Journal of Research and Development*, vol. 61, no. 1, pp. 4:41–4:52, 2017, doi: 10.1147/JRD.2016.2629318.

[12] R. Jevsejev and M. Bereiša, "Improvement of incident management model using machine learning methods," *Mokslas – Lietuvos Ateitis / Science – Future of Lithuania*, vol. 16, 2024, doi: 10.3846/mla.2024.21633.

[13] S. Samala, "AI-driven Jira automation: Using machine learning to optimize sprint planning and incident resolution," *Frontiers in Emerging Artificial Intelligence and Machine Learning*, vol. 1, no. 1, pp. 44–65, 2024.

[14] J. Malca, A. Barrera, J. L. Castillo-Sequera, and L. Wong, "Framework to increase the level of customer satisfaction in telecommunication services in Peru using TQM and business intelligence," in *Proc. 2024 Int. Seminar on Application for Technology of Information and Communication (iSemantic)*, 2024, pp. 272–277, doi: 10.1109/iSemantic63362.2024.10762032.

[15] Z. S. Ibrahim, "Strategic implementation of IT governance for optimising information systems performance," *Central Asian Journal of Mathematical Theory and Computer Sciences*, vol. 6, no. 4, pp. 773–786, 2025.

[16] P. Kubiak and S. Rass, "An overview of data-driven techniques for IT-service-management," *IEEE Access*, vol. 6, pp. 63664–63688, 2018, doi: 10.1109/ACCESS.2018.2875975.