

RiskMetaStack: A Hybrid Machine Learning Model for Efficient Risk Level Classification in Structural Health Management

Surajit Mohanty

Biju Patnaik University of Technology

Subhendu Pani

Krupajal Engineering college

Sonalika Subudhi

Siksha O Anusandhan University Institute of Technical Education and Research

Alok Kumar Pati

`alokpati6065@gmail.com`

Siksha O Anusandhan University Institute of Technical Education and Research

<https://orcid.org/0009-0009-4258-2997>

Research Article

Keywords: RiskMetaStack, Project risk management, Risk level classification, Stacking ensemble learning, Cross validation performance

Posted Date: June 18th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9071045/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

RiskMetaStack: A Hybrid Machine Learning Model for Efficient Risk Level Classification in Structural Health Management

Surajit Mohanty¹, Subhendu Kumar Pani², Sonalika Subudhi³, *Alok Kumar Pati³

¹Department of Computer Science and Engineering, Biju Patnaik University of Technology, Rourkela, Odisha. mohanty.surajit@gmail.com, Orcid Id:0000-0002-7909-3478

²Krupajal Computer Academy Odisha, India, skpani.india@gmail.com, Orcid Id:0000-0001-9595-0567

³Department of Computer Application, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, 751030, Odisha, India. sonalikasubudhi2002@gmail.com,

* Corresponding author: alokpati6065@gmail.com, alokpati@soa.ac.in,

Abstract

This study develops RiskMetaStack, a hybrid stacking model for classifying project risk levels (Low, Medium, High) to enhance governance in project management analytics. The pipeline applies missing-value imputation, duplicate removal, IQR-based outlier handling, and Z-score standardization to features including cost, schedule, energy, and safety indicators; five classifiers are benchmarked on an 80/20 split, and the ensemble stacks Logistic Regression and Gaussian Naïve Bayes with an MLP meta-learner, evaluated via ROC AUC and 5-fold cross-validation. The dataset is imbalanced (High 50.2%, Low 34.2%, Medium 15.6%); Logistic Regression and Naïve Bayes lead baselines (accuracy $\approx 85.5\%$; precision $\approx 85.6\%$ and 85.4% ; F1 $\approx 85.5\%$) over Decision Tree (83%), SVM (80%), and KNN (69%), while RiskMetaStack attains 0.88 accuracy with High: precision 0.97/recall 0.90/F1 0.93; Low: 0.79/0.90/0.84; Medium: 0.79/0.84/0.82; macro precision/recall/F1 0.85/0.88/0.86; AUCs: High 0.98, Low 0.98, Medium 0.95; and 5-fold means accuracy 0.88 and F1 0.89. Stacking linear probabilistic learners improves generalization and preserves interpretability, with strong detection of High-risk projects and residual confusion between High and Medium; correlation patterns (Planned vs. Actual Cost ≈ 0.96 ; Safety_Risk_Score vs. Risk Level ≈ -0.69) support feature relevance and motivate calibration and feature enrichment. RiskMetaStack surpasses single-model baselines and shows stable cross-validated performance suitable for deployment, contributing a transparent preprocessing and ensemble framework with validation aligned to journals in operations, risk analytics, and managerial decision systems.

Keywords: RiskMetaStack; Project risk management; Risk level classification; Stacking ensemble learning; Cross validation performance.

1. Introduction

Reliable risk stratification is critical to effective governance of civil infrastructure projects. Large-scale infrastructure programs frequently suffer cost overruns and schedule delay. For example, McKinsey estimates that 98% of capital infrastructure projects exceed budgets and timelines (average cost overrun

~80%, delay ~20 months) [1, 2]. These failures are often traced to weak risk management systems. Classifying projects into High, Medium, and Low risk categories enables decision makers to prioritize resources and interventions, thereby reinforcing project value and preventing failures [3, 4]. In practice, asset owners and engineers use Building Information Modeling (BIM) to integrate multidimensional project data and support lifecycle decision making. BIM's advanced capabilities including 4D scheduling, 5D cost estimation, and real-time data integration have been shown to enhance risk analysis and proactive planning [5]. By fusing BIM data with predictive analytics, governance frameworks can achieve early hazard detection and responsive mitigation, as demonstrated in recent reviews of BIM based risk management [6].

Developing accurate risk models is challenging due to data heterogeneity and quality issues. Civil project datasets combine diverse indicators financial (planned vs actual costs, overruns), schedule deviations, safety scores, structural health measures, and environmental factors drawn from multiple sources [7]. Such data may exhibit missing values, duplicate records, extreme outliers, and strong correlations among features (e.g. planned vs actual cost) that can bias standard models. Indeed, multicollinearity among predictors can render coefficient estimates unstable [8, 9]. Moreover, the class distribution is highly imbalanced: in our data, about 50% of projects are High risk, 34% Low-risk, but only 16% Medium-risk [10, 11]. This skew complicates learning, since naive accuracy metrics can be misleading. In short, realistic BIM-integrated data are noisy and uneven, underscoring the need for robust preprocessing and modeling. As one study notes, “good performance achieved through managing data” is essential for construction success. We therefore focus on cleaning and harmonizing inputs imputing missing entries, removing duplicates, capping outliers (via IQR thresholds), and standardizing features before fitting models [12, 13].

The motivation is a transparent, generalizable risk classifier that fits within BIM-based project workflows. We design, a hybrid stacking ensemble optimized for interpretability and performance. At the first layer, we use two simple probabilistic base learners: Logistic Regression and Gaussian Naïve Bayes [14, 15]. Logistic regression provides a linear decision rule whose coefficients indicate feature influence, and Gaussian Naïve Bayes assumes conditional independence to deliver fast, probabilistic scores. Using only these linear-type models at the base promotes transparency: each can be readily understood or audited [16, 17]. The second layer is a multi-layer perceptron (MLP) meta-learner, which combines the base models' outputs into a final prediction [18, 19]. This architecture leverages the strengths of each base classifier while the neural meta-learner captures any remaining nonlinear patterns [20, 21]. In sum, RiskMetaStack stacks “linear probabilistic learners” to improve generalization without sacrificing explainability. To evaluate RiskMetaStack, we split the BIM-10 derived dataset into training (80%) and test (20%) sets, and also perform 5-fold cross-validation for robustness. We benchmark standard classifiers (Decision Tree, SVM, KNN) as baselines. Model performance is assessed by accuracy, precision/recall, and ROC AUC per risk class. The proposed RiskMetaStack achieved an overall accuracy of 0.88. It yields very high AUC (High=0.98, Low=0.98, Medium=0.95) and balanced F1 scores across classes, far outperforming single models. Notably, High-risk projects are detected with high precision (0.97) and recall (0.90), ensuring few critical risks are missed. Performance is stable under cross-validation (mean accuracy ~0.88, F1 ~0.89).

These results demonstrate that a carefully preprocessed stacking ensemble can effectively stratify project risk. By emphasizing interpretability (via linear base models) and aligning with BIM's integrated data, RiskMetaStack is well suited for operational dashboards and governance workflows. In practice, the model outputs can feed into real-time risk management dashboards now becoming “an integral part of the risk management process and the preferred tool for risk reporting” enabling project leaders to visualize risk

levels dynamically. Ultimately, RiskMetaStack provides a transparent, data driven engine for proactive decision making in civil infrastructure, strengthening oversight from planning through execution.

2. Materials and Methodology

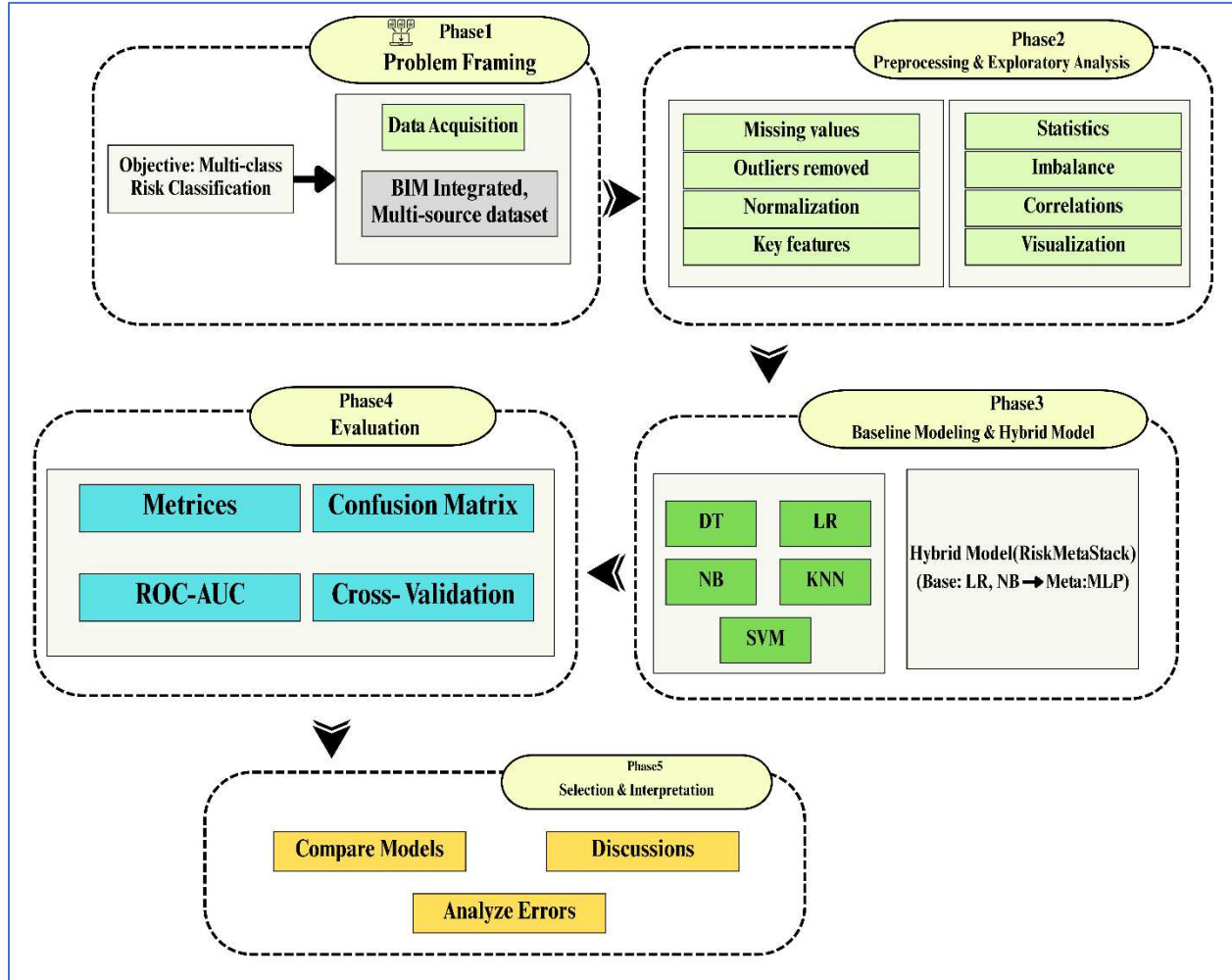


Figure1. Comprehensive workflow for model development and evaluation: from data preprocessing to best model selection.

In Figure1, the workflow for multi class risk classification in BIM-driven analytics is organized into five interconnected phases. The first phase involves defining the problem and collecting the dataset, where a BIM-integrated, multi-source dataset is gathered and aligned with the study objectives. In the second phase, the focus shifts to data preprocessing and exploration, which includes handling missing values, removing outliers through statistical rules, normalizing features, and summarizing the most important predictors. This stage also involves running statistical tests, checking for data imbalance, exploring feature correlations, and creating visualizations to strengthen understanding of the dataset. The third phase is dedicated to model development, beginning with baseline models such as Decision Tree, Logistic Regression, Naïve Bayes, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). Building on these, the RiskMetaStack hybrid model is introduced, combining Logistic Regression and Naïve Bayes as base learners with a Multi-Layer Perceptron (MLP) as the meta learner. The fourth phase emphasizes model evaluation using metrics

like accuracy, precision, recall, F1-score, confusion matrices, ROC–AUC, and cross-validation to measure both class-specific performance and overall stability. Finally, the fifth phase focuses on interpretation and decision-making, where results are compared, findings are discussed, and errors are analyzed to ensure practical and meaningful insights for effective risk management. These phases are connected through iterative feedback loops that enable ongoing improvement as data quality, feature engineering, and model performance evolve. Altogether, this structured workflow supports transparent modeling, rigorous validation, and actionable outcomes to guide informed decision-making in project risk assessment.

2.2 Data Description

This comprehensive dataset integrates BIM with AI to facilitate whole life cycle management of civil engineering projects which is publicly available [22]. The dataset encompasses multidimensional project parameters, including financial metrics (planned vs. actual costs, overruns), temporal data (schedule deviations, duration analysis), structural health indicators (vibration levels, crack width measurements, load-bearing capacity), and environmental monitoring variables (temperature, humidity, air quality conditions). This extensive dataset revolutionizes civil engineering project management by integrating BIM with cutting edge AI. It offers a holistic approach to whole life cycle management, providing a wealth of multidimensional project parameters [23]. These include crucial financial metrics such as planned versus actual costs and cost overruns along with temporal data that tracks schedule deviations and duration analysis. In addition, the dataset encompasses essential structural health indicators, including vibration levels, crack width measurements, load bearing capacity, and environmental monitoring variables like temperature, humidity, and air quality. It classifies project risk levels into three categories: Low, Medium, and High. This classification is based on a thorough integrated assessment of cost performance, schedule adherence, safety compliance, and structural health status. This dataset empowers proactive risk mitigation by enabling predictive analytics, facilitating informed decision-making across various project types, including bridges, roads, and buildings [23, 24].

2.3 Data Preprocessing

Thorough preprocessing is crucial for reliable risk classification from BIM-integrated project data. First, we identify and handle missing or erroneous entries (e.g. via median imputation or domain-based rules) to ensure a complete dataset. Next, we detect outliers using the interquartile range (IQR) rule and remove or cap extreme values. IQR-based trimming is preferred because it is non-parametric and robust to skewed distributions, preventing rare anomalies from distorting model training [25]. After outlier treatment, all numerical features are standardized by Z-score normalization (subtracting the mean and dividing by the standard deviation). This transformation forces each feature to have mean 0 and unit variance, so that cost, schedule, or other BIM-derived metrics are on a common scale. Z-score scaling ensures no single feature dominates due to its range, and it typically yields faster convergence for learning algorithms sensitive to feature scale. Any categorical fields (e.g. project type, region) are encoded (one-hot or ordinal) so that all inputs are numeric and comparable [25, 26]. By cleaning outliers and normalizing scales, preprocessing addresses real world data issues (noise, heterogeneity, bias) and lays a reliable foundation for accurate risk level classification. The IQR is calculated using equation (1).

$$\text{IQR} = Q3 - Q1 \quad (1)$$

In eq1, Q1 is the first quartile (25th percentile) and Q3 is the third quartile (75th percentile). Once the IQR is determined, outliers are identified using the thresholds i.e. lower Bound = $Q1 - 1.5 \times \text{IQR}$ and

upper Bound = $Q3 + 1.5 \times IQR$. Any data point that falls below the lower Bound or above the upper Bound is considered an outlier. These outliers were either capped or removed, depending on their impact on the analysis.

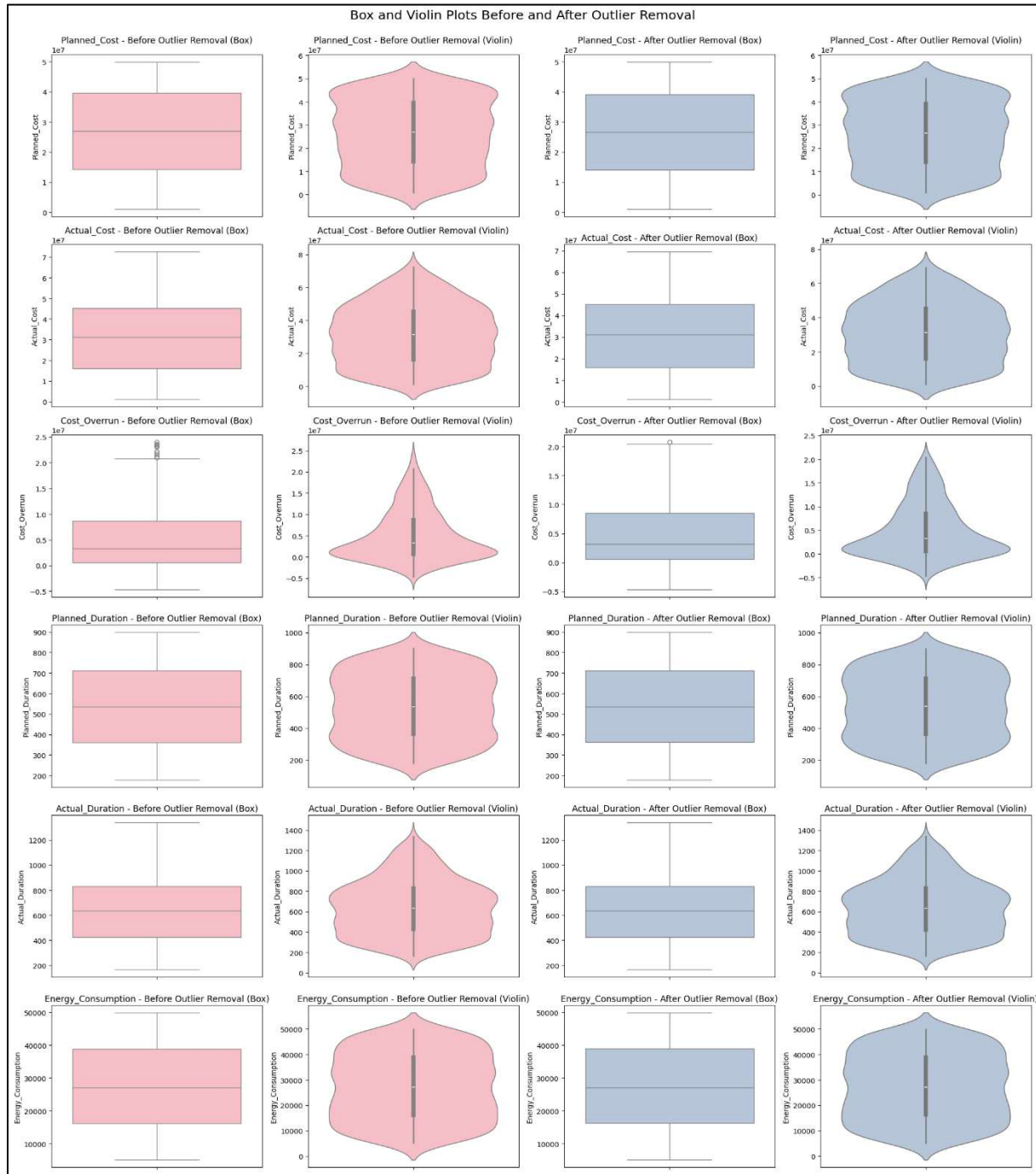


Figure 2 Box and Violin Plots Before and After Outlier Removal. This figure illustrates the distribution of key project features before and after the removal of outliers. The left columns show the box and violin plots of the features Planned Cost, Actual Cost, Cost Overrun, Planned Duration, Actual Duration, Energy Consumption, Material Usage, Labor Hours, Equipment Utilization, Accident Count, Safety Risk Score, Image Analysis Score, and Completion Percentage before the removal of outliers, while the right columns display the corresponding plots after outlier removal.

The box and violin plots in Figure 2 provide a comparative view of the distribution and spread of project-related features both before and after outlier removal. In the "before" plots (on the left), several features show wide distributions, indicating the presence of outliers that may skew the results. For instance, Cost Overrun exhibit significant variability, suggesting extreme values. After applying the IQR method for outlier removal (on the right), these distributions become more concentrated, reflecting a cleaner dataset with extreme values eliminated. The removal of outliers helps to reduce the potential distortion they could cause in the subsequent analysis, leading to more reliable and robust model training.

2.4 Data Analysis

Exploratory analysis ties the data to the risk-classification objective by revealing which features truly discriminate project risk. We first examine basic statistics and class frequencies to understand the dataset for example, checking the proportions of Low/Medium/High risk projects, then we plan corrective measures (such as resampling or class weights) so that the model is not biased toward the majority class. We then analyze pairwise correlations among features and between each feature and the risk label. High correlations indicate redundant information, while strong feature risk correlations highlight informative predictors. To reduce redundancy, we apply principal component analysis (PCA) and correlation filtering: features that are highly collinear or have little variance are combined or removed. In practice, this means focusing on the subset of BIM metrics that best separate risk categories (e.g. schedule deviation, cost growth). By selecting only the most relevant, non-redundant features, we improve model generalization and reduce overfitting directly enhancing classification quality.

Frequency analysis is employed to quantify the occurrence of each category or class within the dataset. This technique is particularly useful for categorical variables, as it reveals the distribution of classes and highlights any imbalance or skewness that could bias model training. In this study, the target variable Risk Level was analysed through frequency counts, with results depicted in Figure 3. The analysis shows that most projects (50.2%) are classified as High Risk, followed by Low-Risk projects at 34.2%, and Medium Risk projects at 15.6%. This imbalance underscores the necessity of targeted risk management strategies, especially for high-risk projects, to mitigate potential challenges and improve project success rates. Figure 3 provides a clear visual representation of this class distribution.

Correlation analysis was conducted to examine the strength and direction of relationships between project features and the target variable. This step is critical for identifying redundant predictors, addressing potential multicollinearity, and selecting the most informative features for model development. The correlation matrix, presented in Figure 4, illustrates these relationships. Strong positive correlations, such as between Planned Cost and Actual Cost (0.96), indicate close dependency, while strong negative correlations, such as between Safety Risk Score and Risk Level (-0.69), reveal that higher safety risk scores are associated with lower overall project risk classification. Conversely, weak or negligible correlations, such as between Material Usage and Completion Percentage, indicate minimal predictive value. By highlighting both strong and weak associations, correlation analysis provides valuable guidance for feature selection and model interpretation in predictive risk classification.

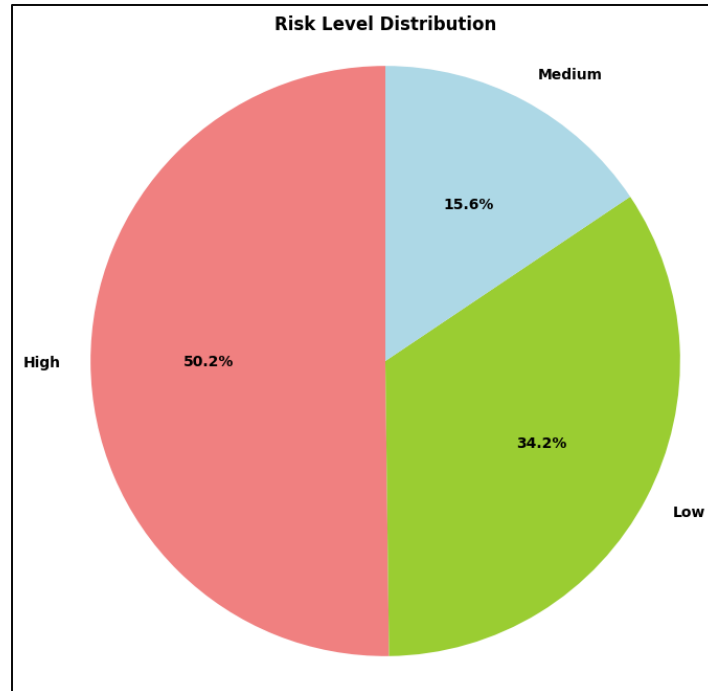


Figure 3: Risk Level Distribution. This pie chart displays the distribution of projects across different risk levels: High, Medium, and Low. The percentages represent the proportion of projects falling into each risk category.

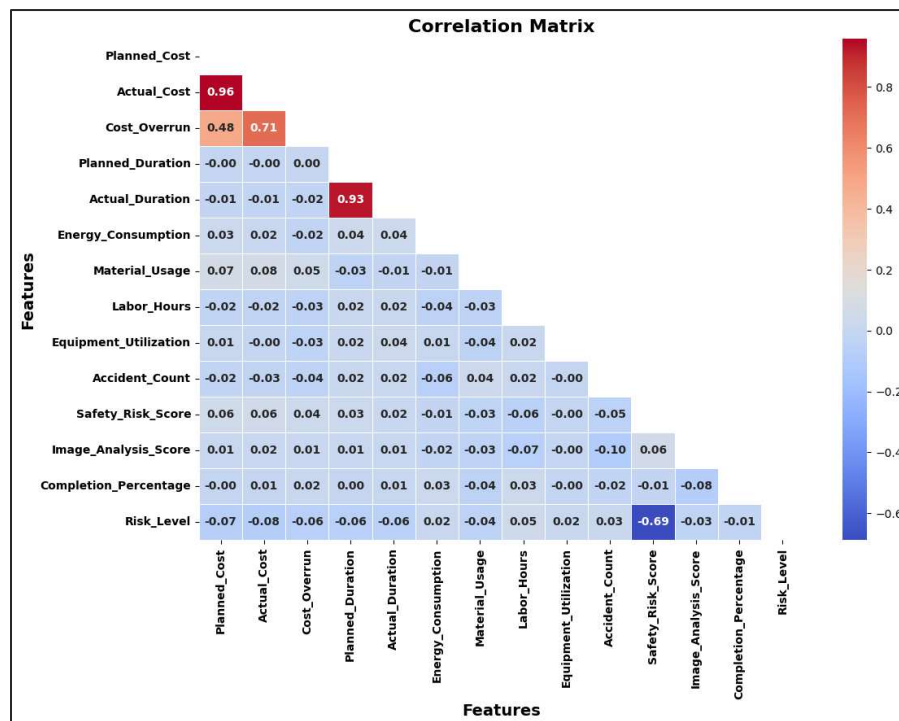


Figure 4: Correlation Matrix of project features and target feature. This heatmap illustrates the correlation between various project features and the target variable, Risk Level. The values represent the strength of the linear relationships, with a color gradient from blue (negative correlation) to red (positive correlation). Darker red indicates strong

positive correlations, while darker blue indicates strong negative correlations, and lighter shades represent weak correlations.

Data Standardization and splitting

Data standardization places heterogeneous BIM-derived features (costs, durations, sensor scores) on a common scale so no single numeric range dominates learning; we apply Z-score normalization equation 2 fitted on the training data and then applied to validation/test folds and to production inputs. This improves numerical stability and convergence for gradient-based learners (the MLP meta-learner), yields fair distance calculations for instance-based models (KNN) and margin computations for SVM, and ensures that the linear probabilistic base learners produce comparable, well calibrated scores thereby strengthening the ensemble's ability to learn true risk signals rather than artefacts of measurement scale [28].

$$Z = \frac{(x - \mu)}{\sigma} \quad (2)$$

Where, x is the original value of the feature, μ is the mean of the feature, σ is the standard deviation of the feature and Z is the standardized value.

To produce unbiased, operationally meaningful performance estimates we use a stratified 80/20 train test split (preserving the High/Medium/Low proportions) combined with 5-fold cross-validation on the training portion for model selection and hyperparameter tuning.

2.5 Machine Learning Models

For risk level classification into Low, Medium, and High, five different classification machine learning models were employed in this study. These models include Decision Tree (DT), which is known for its interpretability and decision-making structure; Logistic Regression (LR), a simple yet effective linear classifier; Naive Bayes (NB), a probabilistic model that assumes feature independence; KNN, a non-parametric method based on the proximity of data points; and SVM, which finds an optimal hyperplane for classifying data. These models were used to assess and compare their effectiveness in classifying project risk levels.

Decision Tree

The Decision Tree algorithm constructs a hierarchical structure of if-else conditions to split the dataset based on feature values. The goal is to partition the data in such a way that each leaf node becomes as "pure" as possible, ideally containing samples of a single class. We used the Gini impurity criterion to evaluate the quality of splits. For a node t with class probabilities p_i , the Gini impurity is calculated using equation (3).

$$G(t) = 1 - \sum_{i=1}^C p_i^2 \quad (3)$$

A lower Gini index indicates a more homogenous node. The decision tree continues splitting until a predefined maximum depth is reached or impurity can no longer be significantly reduced. In water quality classification, decision trees provide interpretability, allowing domain experts to visualize how different parameter thresholds contribute to safety categorization [29].

Logistic Regression

Logistic Regression is a widely-used probabilistic classifier suitable for binary classification problems. It models the probability that a given input $x \in \mathbb{R}^n$ belongs to a particular class $y \in \{0,1\}$. The hypothesis function is given by the sigmoid (logistic) function shown in equation (4).

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^n \beta_i x_i)}} \quad (4)$$

Here, β_0 is the bias term, β_i are the feature weights, and x_i are the standardized input features. Logistic regression assumes a linear relationship between the input features and the log-odds of the outcome. In this study, logistic regression served as a baseline linear model, providing insights into how well water quality classes can be separated in the feature space using linear decision boundaries [30].

Naïve Bayes (NB)

Naïve Bayes is a probabilistic classifier derived from Bayes' theorem, which calculates the probability of a class given an input feature vector. It assumes that the features are conditionally independent, meaning that the presence of a particular feature is unrelated to the presence of other features, given the class. The posterior probability of class k given the input vector x is computed using equation (5).

$$P(k|x) = \frac{P(y) \prod_{i=1}^n P(x_i|y)}{P(x)} \quad (5)$$

In this study, the Gaussian Naïve Bayes variant was used, which assumes that the continuous features follow a Gaussian (normal) distribution. Although the model's assumption of feature independence is often unrealistic in many real-world scenarios, it remains computationally efficient and consistently delivers reliable performance in various practical applications [31].

Support Vector Machine (SVM)

SVM is a powerful algorithm that aims to find the optimal hyperplane that maximally separates multiple classes. For multi-class classification, the decision function is given by equation (6).

$$f(x) = w^T \phi(x) + b \quad (6)$$

Where $\phi(x)$ is a (possibly nonlinear) transformation of the input space using a kernel function, and w and b are model parameters. In multi-class SVM, the model can be trained using strategies such as one-vs-one or one-vs-all, where the algorithm builds multiple binary classifiers and combines their outputs to classify instances into one of the multiple classes. The SVM optimizes the objective function given in equation (7).

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad \text{subject to } y_i(w^T x_i + b) \geq 1 \quad (7)$$

In this study, we used the Radial Basis Function (RBF) kernel, which allows the model to learn nonlinear decision boundaries, essential for capturing complex interactions among features. The RBF kernel projects the data into a higher-dimensional space where a linear hyperplane can be used to separate the classes [32].

K-Nearest Neighbours (KNN)

KNN is a non-parametric, instance-based learning algorithm. For a given test point, the algorithm finds the k closest points in the training dataset based on a distance metric (commonly Euclidean distance) which is calculated using equation (8).

$$d(x_i, x_j) = \sqrt{\sum_{n=1}^N (x_{in} - x_{jn})^2} \quad (8)$$

The class label is determined by majority voting among the k nearest neighbours. KNN is particularly useful in cases where the data distribution is irregular or nonlinear. However, it can be computationally expensive and sensitive to the curse of dimensionality. We selected $k = 5$ in our experiments for balanced bias-variance trade-off [33].

Proposed Hybrid Model: RiskMetaStack

The proposed hybrid model, RiskMetaStack, aims to enhance predictive accuracy by combining the strengths of multiple machine learning algorithms. By utilizing a stacking approach, it integrates the predictions of base learners and a meta learner to provide a more robust and reliable risk level classification. Figure 5 illustrates the architecture of the RiskMetaStack hybrid model. The model consists of two base learners: Logistic Regression (Base Learner 1) and Naïve Bayes (Base Learner 2), which independently generate predictions. These predictions are then fed into a Neural Network (Meta Learner), which combines the outputs from both base models to produce the final classification. This ensemble method improves the overall predictive performance by leveraging the strengths of multiple models, ensuring more accurate classification of project risk levels as Low, Medium, or High.

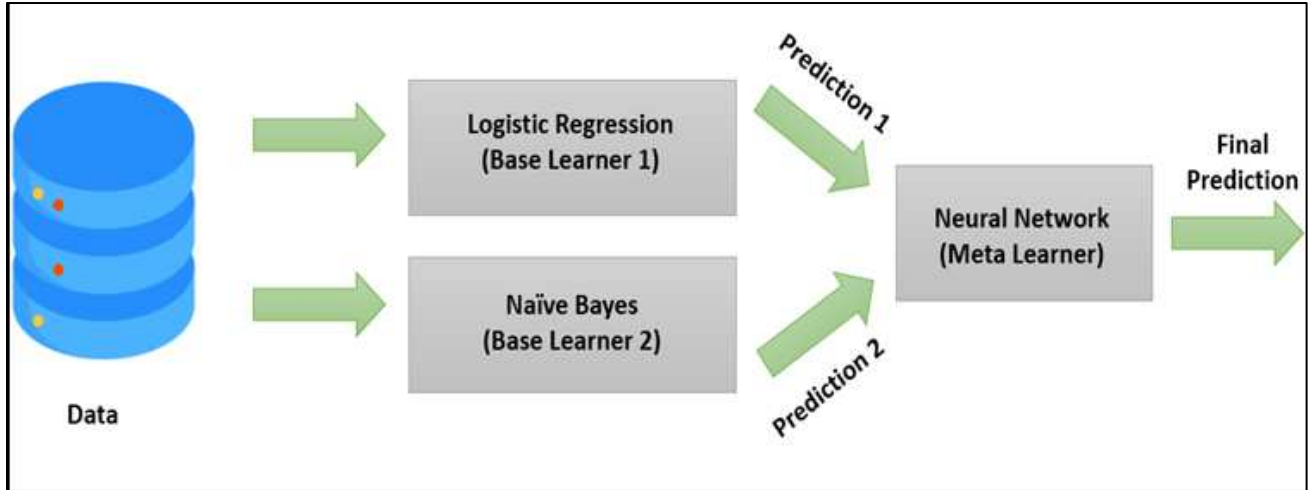


Figure 5: Architecture of the Proposed Hybrid Model RiskMetaStack. This figure illustrates the architecture of the proposed hybrid model, RiskMetaStack, designed for project risk level classification. The model consists of two base learners: Logistic Regression and Naïve Bayes, which generate individual predictions. These predictions are then combined by a Neural Network (meta learner) to produce the final classification of the project risk level.

The Training Phase of the RiskMetaStack hybrid model involves two main steps: training the base learners and training the meta-learner. The process ensures that both the base models and the meta-model are optimized to make accurate predictions.

Algorithm 1 presents the RiskMetaStack approach for predicting the Risk level class based on the input data. This method involves normalizing the input features, training multiple base learners (Logistic Regression and Gaussian Naive Bayes), and utilizing a stacking model that combines the base models' predictions through a meta-learner (MLP Classifier). The algorithm trains the base models, then the stacking classifier, and finally makes predictions on test data or new inputs. This ensemble technique aims to improve classification accuracy by aggregating diverse model outputs, providing a more reliable prediction of the risk level.

Algorithm 1: RiskMetaStack

Input: Training dataset $\mathcal{D} = \{ (x_i, y_i) \mid x_i \in \mathbb{R}^d, y_i \in \{0,1,2\} \}, i = 1 \text{ to } N$

Output: Final prediction $\hat{y} \in \{0,1,2\}$ (Risk level class)

1. Normalize: $x'_i = (x_i - \mu) / \sigma$
2. Define Base learners:
 $f_1 \leftarrow \text{LogisticRegression}(\text{max_iter}=1000), \quad f_2 \leftarrow \text{GaussianNB}()$
3. Train Base Learners:
Train f_1 and f_2 on the training set $\mathcal{D}_{\text{train}}$.
 $f_1 \leftarrow \text{fit}(X_{\text{train}}, Y_{\text{train}}), \quad f_2 \leftarrow \text{fit}(X_{\text{train}}, Y_{\text{train}})$
4. Stacking Model:
Define the meta-learner:
 $f_{\text{meta}} \leftarrow \text{MLPClassifier}(\text{max_iter}=1000)$
Create a stacking classifier using f_1, f_2 and f_{meta} :
 $\text{stacking_model} \leftarrow \text{StackingClassifier}(\text{estimators} = \{(f_1), (f_2)\}, \text{final_estimators} = f_{\text{meta}})$
5. Train the Stacking Models:
Train the stacking classifier:
 $\text{stacking_model} \leftarrow \text{fit}(X_{\text{train}}, Y_{\text{train}})$
6. Make Predictions:
Make predictions with the stacked model:
 $\hat{y} = \text{stacking_model.predict}(X_{\text{test}})$
7. For new input x_{new} :
 $x'_{\text{new}} = \text{normalize}(x_{\text{new}})$
 $z_1 = f_1(x'_{\text{new}}), \quad z_2 = f_2(x'_{\text{new}})$
 $z = [z_1 \quad z_2]$
 $\hat{y}_{\text{new}} = f_{\text{meta}}(z)$
Return $\hat{y}_{\text{new}}$ (Predicted risk level for new input)

Advantages of RiskMetaStack

The RiskMetaStack model offers several advantages in predicting risk levels, especially in complex classification scenarios. By using a stacking approach, it combines multiple base models (such as Logistic Regression and Gaussian Naive Bayes) and aggregates their predictions through a meta-learner (MLP Classifier). This ensemble technique improves the model's ability to generalize and handle different types of data patterns, reducing the likelihood of overfitting. The use of diverse base learners allows the model to capture varied relationships within the data, while the meta-learner refines these predictions, leading to more accurate and reliable risk level classification. Overall, RiskMetaStack enhances prediction performance by leveraging the strengths of different algorithms in a unified framework.

2.8 Model Evaluation Metrics

To evaluate the performance of the machine learning classifiers used for risk level classification, a comprehensive set of evaluation metrics was employed. These include Accuracy, Precision, Recall, F1-Score, as well as Support, Macro Average, and Weighted Average. These metrics help assess the model's overall effectiveness and provide insights into its performance across different risk level categories: High, Medium, and Low.

Accuracy measures the overall performance by calculating the proportion of correct predictions. It includes both True Positives (TP) and True Negatives (TN), while also considering False Positives (FP) and False Negatives (FN). The formula for accuracy is given in equation (9) [34].

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

While accuracy offers a general overview of model performance, it can be misleading in imbalanced datasets.

Precision indicates the model's ability to correctly identify only the relevant instances within a class. It calculates the proportion of true positives out of all predicted positives, defined in equation (10) [34].

$$\text{Precision} = \frac{TP}{TP+FP} \quad (10)$$

Recall, also known as sensitivity, quantifies the model's ability to find all relevant cases within a class. It is the ratio of true positives to the total actual positives, given in equation (11).

$$\text{Recall} = \frac{TP}{TP+FN} \quad (11)$$

F1-Score combines both precision and recall using the harmonic mean. This provides a balanced measure, particularly useful in cases of class imbalance. The formula for F1-Score is given by equation (12).

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (12)$$

Support refers to the number of actual instances of each class in the dataset. Although it is not a performance metric itself, it provides context for interpreting other metrics, especially in understanding the prevalence of each class in the dataset [34, 35]..

Macro Average calculates the unweighted mean of the metric (such as precision or recall) independently for each class. This treats all classes equally, regardless of their frequency in the dataset, and is computed using equation (13).

$$\text{Macro Avg} = \frac{1}{N} \sum_{i=1}^N \text{Metric}(i) \quad (13)$$

Weighted Average accounts for the number of instances in each class when computing the average, making it more representative of the model's performance on imbalanced datasets. It is expressed as in equation (14).

$$\text{Weighted Avg} = \frac{\sum_{i=1}^N \text{Support}(i) \times \text{Metric}(i)}{\sum_{i=1}^N \text{Support}(i)} \quad (14)$$

These metrics were computed using the `classification_report()` function from the scikit-learn library, offering a comprehensive and interpretable evaluation of model performance in predicting risk levels in the project [34, 35].

3. Results

System Configuration and Software Environment for Experimental Analysis

The computer experiments were conducted on a system running Windows 11 with an x64-based architecture and Windows version 10.0.26100. The machine is equipped with an Intel Core i5-12450H processor, a 12th-generation chip featuring 8 cores that operate at a clock speed of up to 2.0 GHz, and 7.9 GB of physical memory (RAM) to handle computational tasks. Storage is distributed across multiple drives: the C: drive with 156.6 GB capacity (61.5 GB free), the D: drive with 353.1 GB space (350.5 GB free), and the G: drive with 161.0 GB total (144.6 GB free). Powered by Intel UHD Graphics and lacking a dedicated NVIDIA GPU, the system relies on CPU-based processing for all computations. For running the codes, Google Colab was utilized, enabling efficient execution of machine learning models and data processing tasks in a cloud environment. The latest versions of key software packages used include Python 3.9.x, scikit-learn 1.1.2, NumPy 1.23.5, Pandas 1.5.3, Matplotlib 3.7.1, and Seaborn 0.12.2, all of which contributed to the successful execution of various tasks within the study.

Performance of Machine Learning Models Used

We trained five machine learning models: Logistic Regression (LR), Naive Bayes (NB), Decision Tree (DT), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) to evaluate their performance in predicting risk levels. The results of these models across various evaluation metrics are shown in Figure 6.

In terms of Accuracy, LR and NB demonstrate the highest performance, both achieving 85.50%, followed by DT at 83.00% and SVM at 80.00%. KNN trails significantly with an accuracy of only 69.00%. Regarding Precision, LR and NB again perform strongly, with 85.64% and 85.43%, respectively, highlighting their ability to correctly identify positive instances. DT and SVM have precision values of 83.06% and 79.92%, respectively, while KNN again shows a much lower performance with 67.85%. The Recall metric, which indicates the model's ability to detect all relevant cases, shows LR and NB leading with 85.50%, followed by DT at 83.00% and SVM at 80.00%. KNN again falls behind, with a recall of only 69.00%. Lastly, the F1 Score, a balanced measure combining both precision and recall, shows LR and NB as the top performers with scores of 85.54% and 85.46%, respectively. DT follows with 82.96%, while SVM and KNN perform much worse, scoring 79.75% and 67.94%, respectively.

Overall, the results indicate that LR and NB consistently outperform the other models across all metrics, making them more reliable for risk level classification. On the other hand, KNN consistently shows lower performance across all evaluation parameters, suggesting it is less effective for this classification task. Since LR and NB performed the best, we utilized them as base learners in our ensemble model.

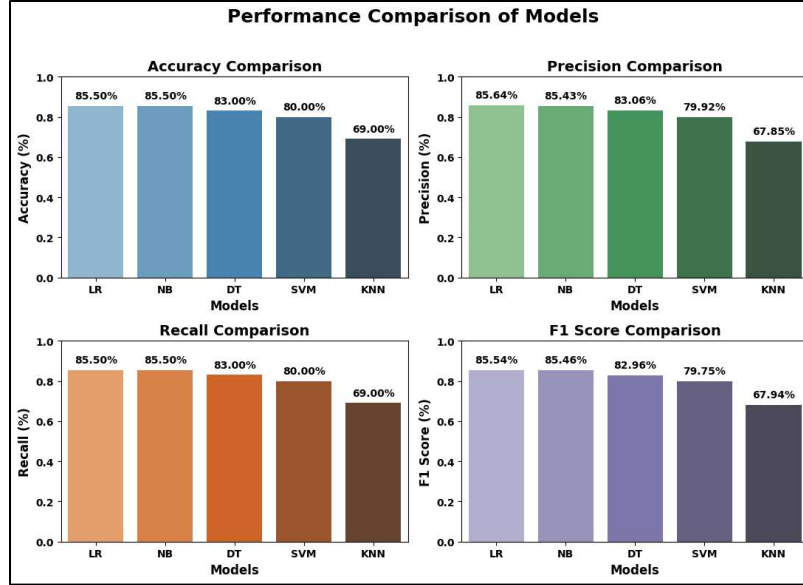


Figure 6: Performance Comparison of Machine Learning Models across Multiple Evaluation Metrics. This figure compares the performance of five different machine learning models i.e. Logistic Regression (LR), Naive Bayes (NB), Decision Tree (DT), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), based on four key performance metrics: Accuracy, Precision, Recall, and F1 Score. Each subplot in the figure presents a bar chart visualizing the performance of the models on these metrics.

RiskMetaStack Model's Performance

The performance of the proposed RiskMetaStack hybrid model for risk level classification is summarized in Table 1, showcasing its effectiveness in accurately predicting risk levels across High, Medium, and Low categories. The model achieves an overall accuracy of 0.88, demonstrating robust performance in correctly classifying instances. It excels in precision for the High risk category (0.97) and achieves strong recall of 0.90 for the same, ensuring reliable identification of high-risk cases. For Low and Medium risk categories, the model maintains decent precision (0.79) and recall (0.90 for Low, 0.84 for Medium), with F1-scores of 0.84 and 0.82, respectively. The macro average and weighted average provide an overall balanced performance, with values of 0.85 for precision, 0.88 for recall, and 0.86 for F1-score, reflecting the model's capability to generalize across risk levels. These results confirm the RiskMetaStack model as a reliable and efficient tool for classifying risk levels, with particularly strong performance in detecting High risk.

Table 1: Performance Metrics of Proposed Hybrid Model – RiskMetaStack for Classification of Risk levels

RiskMetaStack Accuracy: 0.88 Classification Report:				
	precision	recall	f1-score	support
High	0.97	0.90	0.93	107
Low	0.79	0.90	0.84	29
Medium	0.79	0.84	0.82	64
accuracy			0.88	200
macro avg	0.85	0.88	0.86	200
Weighted avg	0.89	0.88	0.88	200

Accuracy Assessment of RiskMetaStack Model

The confusion matrix Figure 7(A) provides a comprehensive evaluation of the RiskMetaStack model’s performance in classifying groundwater contamination risk into High, Medium, and Low categories, offering critical insights into both correct and incorrect predictions. The model exhibits strong accuracy in identifying High-risk cases, with 96 instances correctly classified and only 11 misclassified as Medium, while the Low-risk category is predicted with high reliability, showing 26 correct classifications and no misclassifications into other classes. In contrast, the Medium-risk category, although reasonably well captured with 54 correct predictions, shows some overlap with adjacent categories, with 7 cases misclassified as High and 3 as Low. From a research perspective, these results highlight the model’s effectiveness in distinguishing High and Low risk levels, which are essential for prioritizing groundwater management strategies, while also revealing challenges in differentiating between High and Medium risks. This emphasizes the need for further refinement in feature representation or model optimization to enhance sensitivity in capturing subtle differences between closely related categories, ultimately strengthening the reliability of the proposed framework.

The ROC curve and corresponding AUC values Figure 7(B) provide complementary insights to the confusion matrix, offering a more comprehensive evaluation of the RiskMetaStack model’s classification performance. While the confusion matrix highlights class-specific accuracy, the ROC curve plots the True Positive Rate against the False Positive Rate to assess the model’s ability to distinguish between categories, with the AUC serving as a summary indicator of overall classification strength. In this study, the RiskMetaStack model achieved outstanding results, with the High- and Low-risk categories both attaining an AUC of 0.98, reflecting excellent discriminative power, while the Medium-risk category achieved a slightly lower yet robust AUC of 0.95, indicating minor challenges in separating Medium from other risk levels. The diagonal dashed line in the plot represents a random classifier, and the clear separation of the model’s curves above this line underscores its strong predictive capability. Overall, these findings validate the effectiveness of the RiskMetaStack model in accurately classifying groundwater risk levels, with particularly high reliability in distinguishing High and Low risks and strong, though slightly reduced, performance for Medium risk cases.

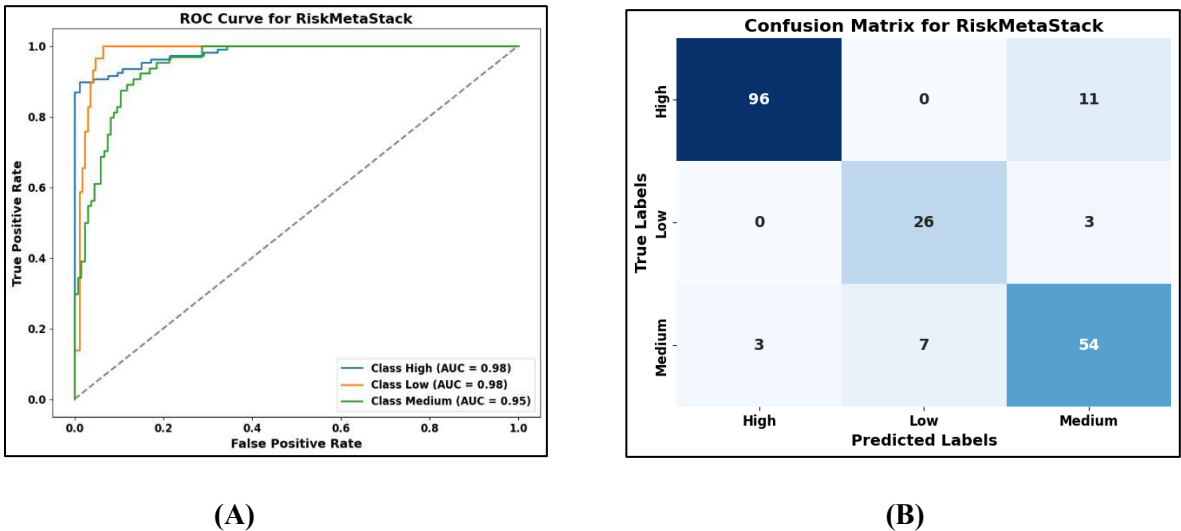


Figure 7(A) presents the confusion matrix of the RiskMetaStack model, showing correct and incorrect classifications across High, Medium, and Low risk levels. Figure 8(B) depicts the ROC curves with corresponding AUC values,

highlighting the model's ability to distinguish between the three risk categories based on the trade-off between True Positive Rate and False Positive Rate.

Validation Performance of RiskMetaStack Model

Validation plays a critical role in assessing the generalizability of machine learning models, ensuring that their predictive performance is not confined to a specific dataset but extends reliably to unseen data. In this study, 5-fold cross-validation was employed, where the dataset was partitioned into five equal folds, and the RiskMetaStack model was iteratively trained on four folds while tested on the remaining fold. The performance, evaluated through Accuracy, Precision, Recall, and F1-Score (Figure 8), demonstrates consistent results across the folds, with Accuracy peaking at 0.92 in Fold 1 and averaging 0.88, Precision reaching 0.92 with an average of 0.89, Recall peaking at 0.92 with an average of 0.88, and F1-Score attaining a maximum of 0.91 with an average of 0.89. The red dashed lines indicating average values further confirm the model's stability, highlighting that minor variations across folds do not undermine its robustness. These findings validate the RiskMetaStack model as a reliable framework for groundwater risk classification, capable of delivering consistent and trustworthy performance across varying subsets of data.

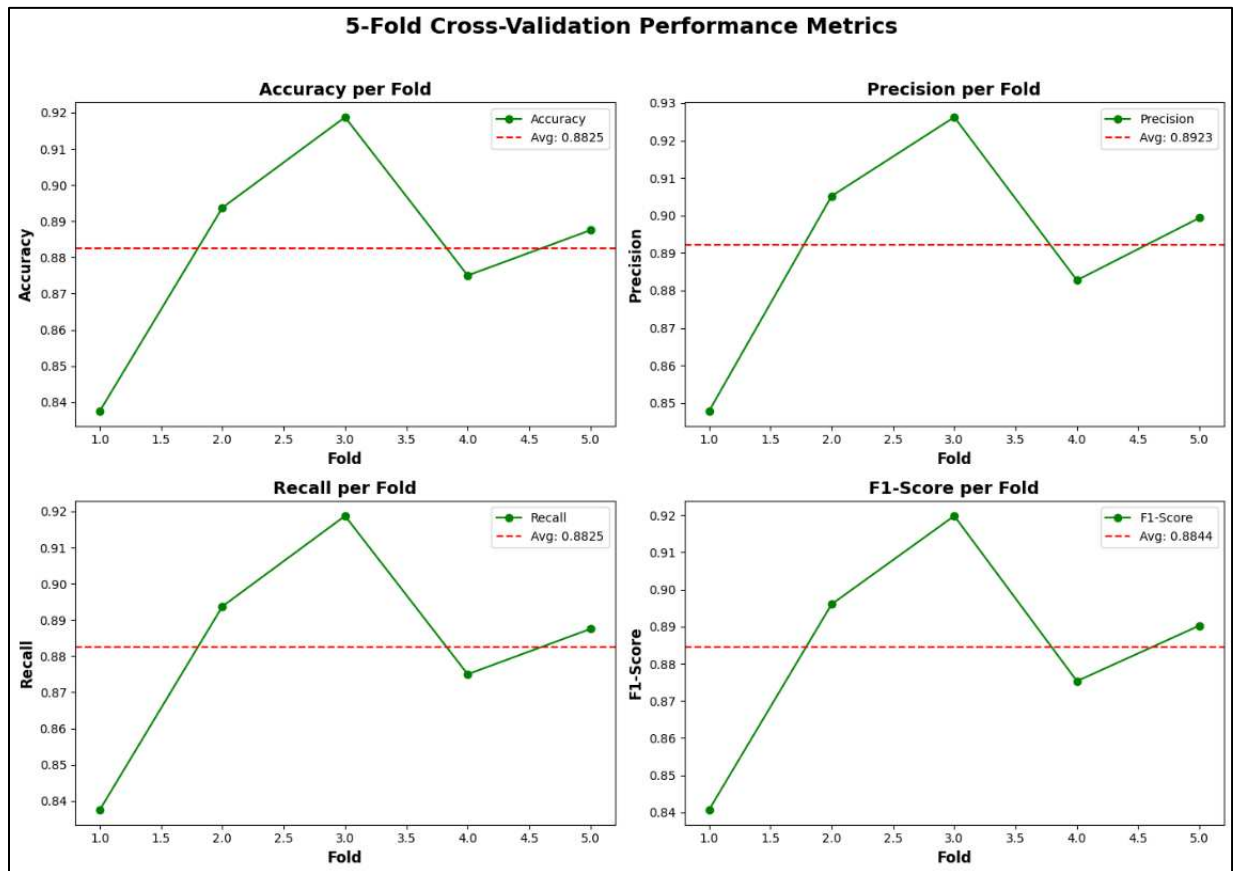


Figure 8: 5-Fold Cross-Validation Performance Metrics for RiskMetaStack This figure displays the performance of the RiskMetaStack model across five folds, evaluated on Accuracy, Precision, Recall, and F1-Score. The red dashed line indicates the average score across all folds, highlighting the model's consistent performance in risk classification.

4. Discussion

The proposed RiskMetaStack achieved an overall accuracy of 0.88 with strong class-wise AUCs (High 0.98, Low 0.98, Medium 0.95), outperforming single-model baselines and delivering balanced F1-scores across risk levels, which underscores the significance of stacking linear probabilistic learners for robust project risk stratification in BIM-integrated settings. Baseline comparisons reveal Logistic Regression and Gaussian Naïve Bayes as top single learners (accuracy ≈ 0.855) and justify their selection as base models for stacking, while Decision Tree, SVM, and KNN lagged, reinforcing the advantage of the proposed architecture on heterogeneous, imbalanced data. Confusion-matrix analysis indicates especially reliable detection of High risk projects (precision 0.97, recall 0.90) and Low-risk projects, with residual confusion concentrated between High and Medium, a realistic challenge when cost growth and schedule deviations are concurrent across many capital works. Cross-validation demonstrates stability (mean accuracy 0.88; mean F1 0.89), and correlation structure (e.g., Planned vs. Actual Cost ≈ 0.96 ; Safety_Risk_Score vs. Risk_Level ≈ -0.69) provides face-valid support for included features and suggests that cost, schedule, and safety signals are jointly informative for risk governance. Overall, the ensemble’s reliability and stability make it suitable for operational dashboards that require interpretable, timely risk signals to guide resource prioritization and proactive control [36, 37, 38].

Interpretations:

The results imply that stacking complementary linear–probabilistic learners with a neural meta-learner captures both well-behaved linear signals and residual nonlinearities, improving separability without sacrificing transparency at the base level a practical advantage for auditability in governance workflows [39, 40]. Given the absence of standardized national or international external benchmarks in the dataset, comparative evaluation is framed against rigorous in-study baselines, class-wise ROC AUC, and cross-validation, aligning with common evaluation practices for multi-class risk identification under imbalance and noise [39, 40]. In BIM-integrated civil projects, local environmental (temperature, humidity, air quality) and operational contexts (asset type, construction method, safety practices) can shift distributions and thresholds, and the study’s inclusion of environmental and structural-health indicators supports adaptability to site conditions while highlighting the need for local calibration when class boundaries are subtle (e.g., High vs. Medium). Relative to other internal models, the ensemble offers a notable improvement in High-risk identification critical where delayed detection has disproportionate cost and safety consequences while maintaining competitive performance in Low and Medium classes, thereby supporting whole-portfolio risk triage [40, 41, 42].

Implications:

The implications are managerial and policy relevant: integrating this pipeline into BIM-linked dashboards enables early warning, targeted inspections, and evidence-based contingency allocation, strengthening compliance and reducing the likelihood of cost overruns and schedule slippage at portfolio scale. Key limitations include reliance on a single dataset, class imbalance skewed toward High risk, residual confusion between High and Medium classes, and the absence of out-of-region external validation that would assess transferability across regulatory, social, and environmental regimes [42, 43, 44]. Additionally, while linear base learners enhance interpretability, they can underfit complex interactions unless the meta-learner and features are carefully calibrated, and compute constraints (CPU-only) may limit extensive hyperparameter searches for the meta-learner. Recommended next steps include cost sensitive thresholds, calibrated probability outputs, targeted feature enrichment (e.g., more granular schedule deviation and safety incident context), domain adaptation to new regions, and prospective validation to quantify operational impact in live dashboards. Future work should explore active learning for difficult boundaries

(High vs. Medium), and systematic error analysis to refine features and meta learning strategies [43, 44, 45].

5. Conclusion

This study introduces RiskMetaStack, a transparent, compute-efficient stacking ensemble that surpasses strong single model baselines and yields stable cross-validated performance for multi-class project risk stratification in BIM integrated, heterogeneous datasets. By uniting rigorous preprocessing with complementary linear probabilistic base learners and a neural meta learner, the framework attains 0.88 accuracy and class-wise AUCs up to 0.98, ensuring reliable detection of High-risk projects while maintaining balanced performance across Low and Medium levels. The method aligns with operational governance needs by producing interpretable, dashboard ready outputs that enable early intervention, resource prioritization, and continuous oversight across bridges, roads, and buildings. Limitations include class imbalance, subtle High Medium confusion, and a single-dataset scope, motivating external validation to assess transferability, targeted calibration to sharpen thresholds, and feature enrichment to improve generalization and sensitivity along challenging boundaries. Collectively, the findings advance data-driven risk analytics and provide a deployable framework that strengthens project governance from planning through execution by embedding auditable intelligence within BIM linked workflows. Future efforts should incorporate cost-sensitive learning, probability calibration, and domain adaptation, plus prospective trials in live dashboards to quantify impacts on safety outcomes, schedule adherence, and cost control, thereby supporting policy and managerial decisions with measurable, class-resolved reliability.

Credit Authorship Statement

Surajit Mohanty contributed to the, methodology, investigation, original draft writing, and visualization. **Subhendu Kumar Pani** contributed to supervision, validation, and critical review of the manuscript. **Sonalika Subudhi** was responsible for conceptualization, resources, data curation, formal analysis, and writing review and editing. **Alok Kumar Pati** provided overall guidance, project administration, and final review of the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

References

- [1]. Changali, S., Mohammad, A., & van Nieuwland, M. (2015). The construction productivity imperative. *McKinsey Quarterly*, (June), 1-10.
- [2]. Zou, Y., Kiviniemi, A., & Jones, S. W. (2017). A review of risk management through BIM and BIM-related technologies. *Safety science*, 97, 88-98.

- [3]. Ding, L. Y., Zhong, B. T., Wu, S., & Luo, H. B. (2016). Construction risk knowledge management in BIM using ontology and semantic web technology. *Safety science*, 87, 202-213.
- [4]. Liu, N., Guo, D., Song, Z., Zhong, S., & Hu, R. (2023). BIM-based digital platform and risk management system for mountain tunnel construction. *Scientific reports*, 13(1), 7585.
- [5]. Singh, J. Risk Assessment and mitigation in Construction Building Projects. *Constuction and Building Materials Engineering*, (2581-6454), 14-23.
- [6]. Pishdad, P., & Onungwa, I. O. (2024). ANALYSIS OF 5D BIM FOR COST ESTIMATION, COST CONTROL, AND PAYMENTS. *Journal of Information Technology in Construction*, 29.
- [7]. Kwak, J. H., Lee, H. B., & Lee, K. H. (2023). Exploring how to organize a unit on box plots through analysis of foreign textbooks. *School Mathematics*, 25(2), 249-276.
- [8]. Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4), 3-13.
- [9]. Rubin, D. B. (1987). *Statistical analysis with missing data*. Wiley.
- [10]. Dormann, C. F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., ... & Lautenbach, S. (2013). Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, 36(1), 27-46.
- [11]. He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9), 1263-1284.
- [12]. Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). *A practical guide to support vector classification*.
- [13]. Zou, Y., Kiviniemi, A., & Jones, S. W. (2017). A review of risk management through BIM and BIM-related technologies. *Safety science*, 97, 88-98.
- [14]. Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2), 241-259.
- [15]. Breiman, L. (1996). Stacked regressions. *Machine learning*, 24(1), 49-64.
- [16]. Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression*. John Wiley & Sons.
- [17]. Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning*, 29(2), 103-130.
- [18]. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088), 533-536.
- [19]. Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8), 861-874.
- [20]. Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, No. 2, pp. 1137-1145).

- [21]. Lu, Y., Gong, P., Tang, Y., Sun, S., & Li, Q. (2021). BIM-integrated construction safety risk assessment at the design stage of building projects. *Automation in Construction*, 124, 103553.
- [22]. https://github.com/mark-kibui/Building-Risk_Level_Prediction/blob/master/bim_ai_civil_engineering_dataset.csv
- [23]. Rahayu, I. S. (2022). BIM-Integrated Construction Safety Risk Assessment at the Design Stage of Building Projects. *Jurnal KaLIBRASI: Karya Lintas Ilmu Bidang Rekayasa Arsitektur, Sipil, Industri*, 5(2), 124-131.
- [24]. Fawad, M., Salamak, M., Poprawa, G., Koris, K., Jasinski, M., Lazinski, P., ... & Gerges, M. (2023). Automation of structural health monitoring (SHM) system of a bridge using BIMification approach and BIM-based finite element model development. *Scientific Reports*, 13(1), 13215.
- [25]. Dongre, P. K., Patel, V., Bhoi, U., & Maltare, N. N. *Decision Analytics Journal*.
- [26]. Demircioğlu, A. (2024). The effect of feature normalization methods in radiomics. *Insights Into Imaging*, 15(1), 2.
- [27]. Cooksey, R. W. (2020). Descriptive statistics for summarising data. In *Illustrating statistical procedures: Finding meaning in quantitative data* (pp. 61-139). Singapore: Springer Singapore.
- [28]. Klabunde, M., Schumacher, T., Strohmaier, M., & Lemmerich, F. (2025). Similarity of neural network models: A survey of functional and representational measures. *ACM Computing Surveys*, 57(9), 1-52.
- [29]. Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (2017). *Classification and regression trees*. Chapman and Hall/CRC.
- [30]. Basetti, A. (2019). A Comparison of Logistic Regression and Neural Networks for Binary Classification Problems. Supervisor: Ola Hössjer. Bachelor's Thesis. Stockholm, Sweden: Stockholm University, Department of Mathematics.
- [31]. Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, No. 22, pp. 41-46).
- [32]. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273-297.
- [33]. Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1), 21-27.
- [34]. Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4), 427-437.
- [35]. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12, 2825-2830.
- [36]. Zian, S., Kareem, S. A., & Varathan, K. D. (2021). An empirical evaluation of stacked ensembles with different meta-learners in imbalanced classification. *IEEE Access*, 9, 87434-87452.

- [37]. Sill, J., Takács, G., Mackey, L., & Lin, D. (2009). Feature-weighted linear stacking. arXiv preprint arXiv:0911.0460.
- [38]. Pati, A. K., Patnaik, J., Dash, S., Mohanty, T., & Tripathy, A. R. (2025, January). Risk Assessment of Global Temperature Anomalies Using Intelligence Frameworks: Evaluating the Impact of Socioeconomic Factors and with a Case Study on Heatstroke Deaths in India. In *2025 International Conference on Ambient Intelligence in Health Care (ICAIHC)* (pp. 1-5). IEEE.
- [39]. Alalwany, E., Alsharif, B., Alotaibi, Y., Alfahaid, A., Mahgoub, I., & Ilyas, M. (2025). Stacking ensemble deep learning for real-time intrusion detection in IoMT environments. *Sensors*, 25(3), 624.
- [40]. Alotaibi, A. (2025). Ensemble Deep Learning Approaches in Health Care: A Review. *Computers, Materials & Continua*, 82(3).
- [41]. Soni, B. (2023). Stacking to improve model performance: A comprehensive guide on ensemble learning in Python. Medium.
- [42]. Kyriakides, G., & Margaritis, K. G. (2019). Hands-On Ensemble Learning with Python: Build highly optimized ensemble machine learning models using scikit-learn and Keras. Packt Publishing Ltd.
- [43]. Seewald, A. K. (2002). Meta-learning for stacked classification. *audiology*, 24(226), 69.
- [44]. Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11, 169-198.
- [45]. Pati, A. K., Tripathy, A. R., Nandi, D., Thakur, R. R., & Pandey, M. (2025). Irrigation water quality prognostication: An innovative ensemble architecture leveraging deep learning and machine learning for enhanced SAR and ESP estimation in the east coast of India. *Journal of Environmental Chemical Engineering*, 13(3), 116433.
- [46]. Al-Husban, Y., Al-Ghriybah, M., Gaeid, K. S., Takialddin, A. S., Handam, A., & Alkhazaleh, A. H. (2023). Optimization of the Residential Solar Energy Consumption Using the Taguchi Technique and Box-Behnken Design: a Case Study for Jordan. *International Journal on Energy Conversion*, 11(1).

