SKE IntelliPower: An AI-Powered Platform for Electrical Contracting, Project Management and Business Intelligence

Anushka Bhosle[1] Arohi Mandekar[2] Anjali Mankar[3] Prof. Vaishnavi Kumbhar[4]
[1,2,3,4]Department of Artificial Intelligence and Machine Learning
[1,2,3,4]Zeal Polytechnic, Narhe, Pune, Maharashtra, India

Abstract — Small electrical contracting firms usually plan projects from personal experience and keep records in registers or scattered spreadsheets, so estimates differ from person to person and past work is hard to reuse. This paper presents the design of SKE IntelliPower, a web platform proposed for Shrikrishana Electricals that joins a public company website, a secured project-management area and a machine-learning (ML) decision-support module. Given project details such as type, built-up area, workforce, complexity and material availability, the module is designed to estimate cost and duration with Random Forest regression, assign a low, medium or high delay-risk class with a Random Forest classifier, and retrieve similar completed projects using TF-IDF and cosine similarity. A dashboard summarises project status. The paper describes the requirements, architecture, data model, interface design, data preparation, modelling approach, evaluation protocol, test plan and limitations, and includes a small worked example of the similarity method. The work is a prototype design: no prediction accuracy is claimed, and any demonstration data will be labelled as such until validated on real company records.

Keywords: Project Cost Estimation, Delay-Risk Classification, Random Forest, TF-IDF, Business Intelligence, Electrical Contracting, Decision Support

I. Introduction

A. Background

Electrical contracting covers wiring, panel installation, lighting, earthing and maintenance work for homes, shops, offices and factories. Each job has its own scope, site conditions and material needs, and a small firm often prices and schedules a new job by recalling similar earlier ones. This is quick, but it is subjective, it is hard to teach to new staff, and it is lost when the experienced person is unavailable. Cost overruns are a well-documented problem in large public works [1], and the same pattern of optimistic early estimates is a concern for any organisation that plans work in advance. Standard project-management guidance treats cost and schedule as separate knowledge areas that need systematic estimating and control [2].

A second difficulty is how information is kept. Many small contractors have a static website that describes the firm, while the actual project data stay in paper registers or separate spreadsheets. As a result, no single place shows which projects are running, how far they have progressed, or how earlier projects compared with their plans. When a new enquiry arrives, the owner cannot quickly search for comparable jobs.

B. Motivation and Contribution

The authors' family business, Shrikrishana Electricals in Pune, faces exactly these problems. This paper therefore describes SKE IntelliPower, a platform with three parts that share one backend and one database: a public website, a login-protected project-management area, and an ML-based intelligence module. The contributions are: (i) a requirement analysis for a small electrical contractor; (ii) an architecture and data model that combine web, database and ML components; (iii) a feature and target definition for cost, duration and delay-risk prediction; (iv) a similar-project search method with a worked example; and (v) an evaluation and test plan that avoids unsupported accuracy claims.

C. Scope and Paper Organization

The paper is a design and methodology report. It does not present prediction results, because the models have not yet been trained and validated on the firm's historical records. Section II reviews related ideas, Section III states the problem, Section IV analyses requirements, Section V gives the system design, Section VI describes data and features, Section VII explains the ML methods, Section VIII sets out the evaluation design, Section IX covers implementation, Section X the testing strategy, Section XI expected outcomes, limitations and ethics, and Section XII concludes.

II. Literature Review

A. Cost and Schedule Estimation

Studies of public works show that actual costs frequently exceed early estimates, and that the cause may be technical error or deliberate optimism [1]. For a small contractor the lesson is practical: estimates should be compared with final outcomes so that the gap becomes visible and can be reduced. This requires records that link each project's plan with its actual cost and duration, which is the first thing the proposed platform provides. Project-management guides describe estimating techniques such as analogous estimating, where a new job is estimated from a similar earlier one [2]. The similar-project module of SKE IntelliPower automates the search step of that technique.

B. Delay-Risk Prediction

Delay prediction has been treated as a classification problem in construction research. Gondia et al. trained and compared supervised learning algorithms on project attributes to predict whether delay risk would be low or high, and reported that such models can support early planning decisions [3]. Their data came from larger construction projects, so their numerical results cannot be transferred to a small electrical contractor. What carries over is the idea of expressing delay as a labelled class and learning it from attributes known at the start of a project.

C. Random Forests

A random forest builds many decision trees, each trained on a bootstrap sample of the data and, at each split, a random subset of features. The predictions of the trees are averaged

All rights reserved by www.ijsrd.com

for regression or combined by voting for classification [4]. Averaging many partly independent trees reduces variance compared with a single tree, and forests need little feature scaling, accept both numeric and encoded categorical inputs, and give a measure of feature importance. General treatments of tree ensembles, bias and variance are given in [5]. These properties suit small tabular data sets such as a contractor's project register.

D. Text Similarity for Case Retrieval

Weighting a term by how rare it is across a collection was introduced as inverse document frequency [6], and combining term frequency with inverse document frequency became a standard weighting scheme in text retrieval [7]. Documents and queries are represented as vectors and ranked by the cosine of the angle between them, a method described in standard information-retrieval texts [8]. Because project descriptions are short and use a limited vocabulary (wiring, panel, lighting, earthing), this simple method is expected to work without large language resources.

E. Research Gap

Among the sources reviewed for this paper, delay and cost research concentrates on large construction projects, while general web platforms for small firms rarely include any prediction. The gap that SKE IntelliPower addresses is the combination, for a small electrical contractor, of a company website, structured project records and lightweight, interpretable ML on those records, delivered with open-source tools and an honest evaluation plan.

III. Problem Statement and Objectives

The problem addressed is the lack of an integrated, data-driven tool that lets a small electrical contractor present its work online, store project records consistently, and obtain a first estimate of cost, duration and delay risk for a new job. The objectives are:

1) to build a responsive public website with services, completed projects and an enquiry form;
2) to provide a login-protected area for adding, updating and reviewing project records;
3) to estimate project cost and duration with regression models;
4) to classify delay risk as low, medium or high;
5) to recommend similar completed projects;
6) to show project summaries on a dashboard; and
7) to evaluate the models honestly and report limitations.

IV. Requirement Analysis

A. Existing practice and its limitations

At present, cost and duration are judged from experience, records live in registers or separate files, and the company website is a static brochure. Consequently estimates vary between people, earlier projects are hard to search, delays and overruns are noticed late, and no dashboard gives an overall view. These observations come from the project sponsor's description of the current workflow and will be confirmed during requirement collection.

B. Functional requirements

Table I lists the main functional requirements.

ID	Requirement
F1	Show company profile, services and project gallery
F2	Accept and store customer enquiries
F3	Authenticate staff and administrators
F4	Create, view, update and search project records
F5	Estimate cost and duration for a new project
F6	Classify delay risk as low, medium or high
F7	List similar completed projects
F8	Display summary charts on a dashboard
F9	Manage content, users and backups

Table I: Functional Requirements

V. System Design

A. Architecture

The proposed system follows a three-tier design, shown in Fig. 1. Visitors and administrators use a React frontend, which calls a Flask REST API. The API follows the representational state transfer style [9] and is built with the Flask framework [10]. It validates inputs, handles authentication, reads and writes a relational database, and calls the ML module. Results are returned to the frontend and shown as forms, tables and charts.

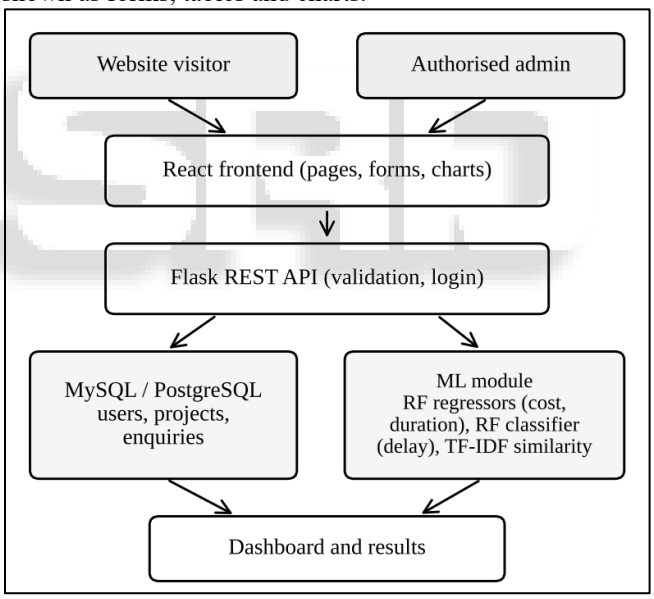


Fig. 1: Proposed architecture of SKE IntelliPower.

B. Functional modules

- Public website. Home, About, Services, Projects and Contact pages with an enquiry form, laid out for desktop and mobile screens.
- Authentication and access. Role-based login limits internal records to approved users.
- Project management. Staff add and update project type, location, area, workforce, budget, planned dates, progress and status.
- Cost and duration estimation. Regression models return an estimate for a new project from its entered attributes.
- Delay-risk analysis. A classifier assigns a low, medium or high risk class using labels derived from past planned and actual dates.

All rights reserved by www.ijsrd.com

– Similar-project recommendation. Past projects with comparable descriptions and attributes are listed for reference.
– Dashboard and administration. Counts of ongoing and completed projects, progress charts, content management, validation and backups.

A chat assistant for frequently asked questions is kept as a later enhancement, to be added only after the core modules are stable.

The modules interact through the shared database and the API and not through each other's code. For instance, the prediction module reads a project's attributes from the request, calls the saved models and writes the result to the predictions table, while the dashboard module only reads from the project and prediction tables. This separation means that a module can be improved, replaced or switched off without breaking the rest. If the models are unavailable, for example during an update, the planning page still lets the user save the project and shows a message that estimates are temporarily unavailable.

C. Data Model

A relational database is suitable because project records are structured and related [11]. Fig. 2 shows the main entities. A project has many progress updates and media files, and each prediction made for a project is stored so that it can later be compared with the real outcome. Storing predictions is important for evaluation, because it allows the firm to measure how close estimates were once a project finishes.

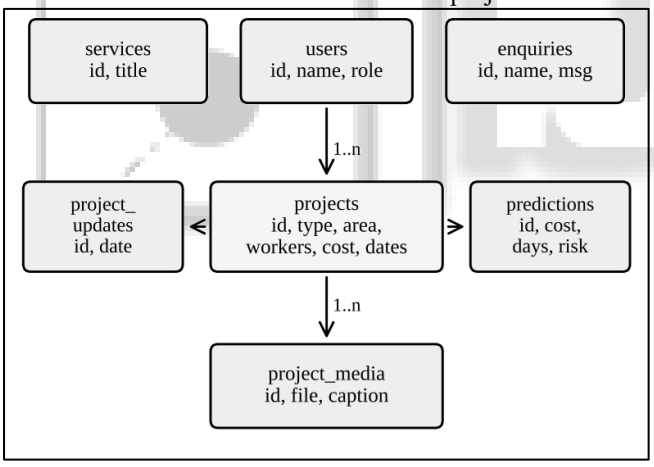


Fig. 2: Main entities of the proposed database.

D. Application Interface

The REST interface exposes the endpoints in Table II. Prediction and similarity calls are made only by logged-in users, and every request is validated before it reaches the model.

Endpoint	Purpose
POST /api/login	Authenticate a user
GET /api/projects	List and search projects
POST /api/projects	Add a project record
PUT /api/projects/{id}	Update project data
POST /api/predict	Cost, duration, risk
GET /api/similar	Similar past projects
GET /api/dashboard	Summary figures
POST /api/enquiry	Store a visitor enquiry

Table II: Planned API Endpoints

VI. Data and Feature Engineering

A. Data Sources

With the owner's permission, historical records will be collected for completed jobs from registers, bills and spreadsheets. Photographs and descriptions will also be collected for the public portfolio. Confidential client details are removed before analysis. If too few real records exist at the time of the first demonstration, a small synthetic data set may be created to test the software; it will be labelled as synthetic, will not be used to claim accuracy, and will be replaced as real records are entered.

B. Input Features and Targets

The features are those known when a project is being planned. Table III lists the proposed inputs and the three targets.

Item	Type	Example
Project type	Category	Residential, commercial, industrial
Built-up area	Number	Square metres
Workers	Number	Count
Complexity	Ordinal	1 (simple) to 5
Materials	Category	Ready, partial, delayed
Start month	Category	January to December
Planned days	Number	Days
Description	Text	Short job summary
Cost (target)	Number	Final cost in rupees
Duration (target)	Number	Actual days
Delay (target)	Class	Low, medium, high

Table III: Proposed Features and Targets

C. Cleaning and Encoding

Records are checked for missing values, inconsistent units (for example square feet and square metres), duplicate entries and outliers. Missing numeric values may be filled with the median of the training data, and rows whose target values are missing are excluded. Categorical features are converted to numeric columns by one-hot encoding. All cleaning statistics are computed on the training data only and then applied to the test data, so that no information from the test set leaks into training.

D. Defining Delay Risk

The delay ratio of a completed project is

$$d = (D_{actual} - D_{planned}) / D_{planned} \quad (1.1)$$

where D is duration in days. The ratio is converted to a class using thresholds, such as those in Table IV. The values shown are initial proposals for discussion; the final cut-offs will be agreed with the owner before training, and not changed afterwards to improve results.

Class	Delay ratio	Meaning
Low	$d \le 0.05$	On or near plan
Medium	$0.05 < d \le 0.20$	Noticeable slip
High	$d > 0.20$	Serious slip

Table IV: Proposed Delay Classes

All rights reserved by www.ijsrd.com

VII. Machine-Learning Methodology

A. Overall pipeline

Fig. 3 shows the pipeline. Records are cleaned, features and targets are defined, and the data are split into training and test sets. Models are trained on the training part and evaluated once on the test part. The trained models are saved to files and loaded by the web service at run time. The models are built with the scikit-learn library [12].

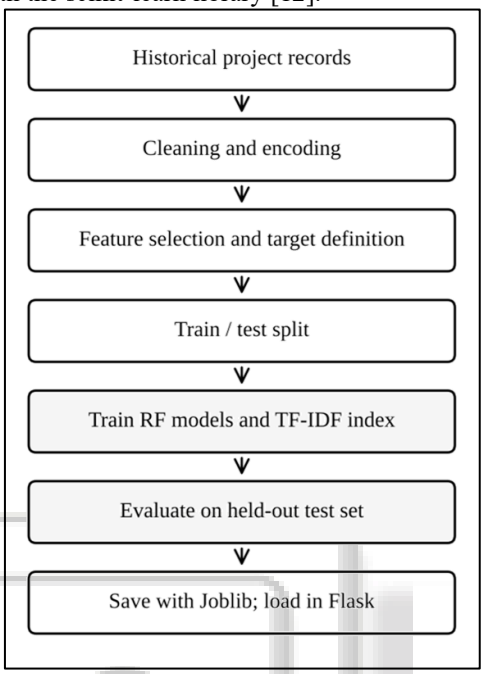


Fig. 3: Model development pipeline.

B. Cost and Duration Regression

Two Random Forest regressors will be trained, one for cost and one for duration. A forest of T trees predicts the average of the individual tree outputs:

$$\hat{y} = (1/T)\ \Sigma\ h_t(x),\ t = 1 \ldots T \qquad (1.2)$$

where x is the vector of project features and h_t is the t-th tree. Each tree is grown by repeatedly choosing the feature and split value that most reduce the squared error in the resulting groups. A simple baseline, such as predicting the training-set mean, will be included so that the benefit of the model can be judged.

C. Delay-Risk Classification

A Random Forest classifier will assign each project to one of three classes. Its trees choose splits that reduce impurity, commonly measured by the Gini index

$$G = 1 - \Sigma\ p_k^2,\ k = 1 \ldots K \qquad (1.3)$$

where p_k is the share of class k in a node. Each tree votes and the most frequent class is returned; the vote shares may also be shown to the user as a rough indication of confidence, with the caution that they are not calibrated probabilities. Serious delays may be fewer than on-time jobs, so class counts will be inspected. If the classes are clearly unbalanced, class weights or resampling methods such as synthetic minority oversampling [13] may be applied to the training data only.

D. Similar-Project Recommendation

Each project description is converted into a TF-IDF vector. The weight of term t in description d is

$$w(t,d) = tf(t,d) \times \log(N / df(t)) \qquad (1.4)$$

where tf is the count of the term in the description, N is the number of descriptions and df is the number of descriptions containing the term. The similarity between a query vector q and a stored vector p is the cosine of the angle between them:

$$sim(q,p) = (q \cdot p) / (\|q\|\ \|p\|) \qquad (1.5)$$

The projects with the highest similarity are shown, and numeric attributes such as area and project type may be used to filter them.

E. Worked Example of the Similarity Method

To make the method concrete, four short invented descriptions were used as a toy collection, and the query "shop wiring lighting" was compared with each using the scikit-learn implementation. The values in Table V are from this illustration only; they are not results from company data.

Toy description	Cosine
office wiring lighting panel installation	0.33
shop lighting wiring led fittings	0.69
factory motor panel earthing cabling	0.00
house wiring earthing lighting	0.39

Table V: Illustrative Similarity (Toy Data)

The shop description ranks first because it shares all three query words, while the factory description shares none and scores zero. The same ranking logic will be applied to real descriptions.

VIII. Evaluation Design

A. Data Split and Validation

The data will be divided into a training set and a held-out test set, for example in the ratio 80:20, with the split fixed by a stored random seed so that results can be repeated. Because the data set may be small, cross-validation on the training set will be used to check that performance is stable across splits.

B. Regression Metrics

For cost and duration, the mean absolute error, root mean square error and coefficient of determination will be reported [14]:

$$MAE = (1/n)\ \Sigma\ |y_i - \hat{y}_i| \qquad (1.6)$$

$$RMSE = \sqrt{[(1/n)\ \Sigma\ (y_i - \hat{y}_i)^2]} \qquad (1.7)$$

$$R^2 = 1 - \Sigma\ (y_i - \hat{y}_i)^2 / \Sigma\ (y_i - \bar{y})^2 \qquad (1.8)$$

MAE is easy to explain in rupees or days, RMSE penalises large errors more strongly, and R^2 compares the model with always predicting the mean. The baseline model is scored in the same way.

C. Classification Metrics

For delay risk, a confusion matrix is built and the following measures are computed for each class [15]:

$$P = TP / (TP + FP),\ \ R = TP / (TP + FN) \qquad (1.9)$$

$$F1 = 2PR / (P + R) \qquad (1.10)$$

Here TP, FP and FN are true positives, false positives and false negatives. Recall for the high-risk class is especially important, because a missed serious delay is more costly than a false alarm.

All rights reserved by www.ijsrd.com

D. Acceptance Criteria

Before results are known, the team and the owner will agree what counts as useful, for example that the regression model must beat the baseline on the test set and that estimates must be accurate enough to help the owner plan. If a model does not meet the agreed criteria, the platform will say that no reliable estimate is available and will fall back to showing similar past projects. Table VI summarises the evaluation plan.

Task	Model	Measures
Cost	RF regression	MAE, RMSE, R^2, baseline
Duration	RF regression	MAE, RMSE, R^2, baseline
Delay risk	RF classifier	Precision, recall, F1, confusion matrix
Similar projects	TF-IDF, cosine	Expert review of top results
Platform	Whole system	Functional and access tests

Table VI: Planned Evaluation Measures

IX. Implementation Plan and Technologies

A. Technology Stack

The technologies in Table VII are open source and suit a small team. They may be adjusted to institute requirements, and only those actually implemented will be reported in the final project.

Component	Technology
Frontend	HTML, CSS, JavaScript, React
Backend	Python with Flask
Database	MySQL or PostgreSQL
ML libraries	Pandas, NumPy, scikit-learn, Joblib
Analysis	Jupyter Notebook
Charts	Chart.js
Tools	Visual Studio Code, Git, GitHub, Figma

Table VII: Proposed Technology Stack

X. Testing Strategy

Testing covers the software and the models separately. Software testing checks each requirement in Table I, using valid, invalid and missing inputs; examples are shown in Table VIII. Model testing follows Section VIII. Usability will be checked by asking the owner and a few staff to complete typical tasks and by viewing the site on phone and desktop screens.

ID	Test	Expected
T1	Log in with correct password	Access granted
T2	Log in with wrong password	Access refused
T3	Open dashboard without login	Redirect to login
T4	Submit empty enquiry form	Validation message
T5	Predict with a missing field	Error, no result
T6	Predict with valid input	Estimate shown
T7	Search with no match	Empty list, no crash
T8	Open site on a phone	Layout fits screen

Table VIII: Sample Test Cases

XI. Expected Outcomes, Limitations and Ethics

A. Expected outcomes

When complete, the platform should let staff open a project, enter its details and see an estimated cost and duration, a delay-risk class and a list of comparable earlier jobs, and then record progress and view it on the dashboard. Visitors should find the firm's services and past work easily and be able to send an enquiry. Benefits for the firm include more consistent estimates, earlier warning of delay, a single store of records, reuse of past projects as references and a clearer view of overall performance.

B. Limitations

Several limits apply. First, a single firm may have only a modest number of completed projects, so models may be unstable and could be judged unsuitable; in that case the system will state this clearly. Second, delay labels depend on the thresholds chosen. Third, estimates cannot capture sudden events such as price changes or site access problems, and a model trained on past prices will become less accurate as prices move. Fourth, similarity search depends on how consistently descriptions are written. Finally, any synthetic or demonstration values used while building the prototype will be marked as examples and not presented as verified predictions.

Table IX lists the main risks and how each will be handled.

Risk	Mitigation
Too few records	Show similar jobs only; collect more
Inconsistent labels	Fix thresholds with owner first
Model becomes outdated	Retrain and monitor live error
Data entry mistakes	Server-side validation, review
Staff do not adopt	Simple forms, short training
Privacy	No personal data in models

Table IX: Risks And Mitigation

C. Ethical and Responsible Use

The system supports human judgement and does not replace it. Every estimate is shown with a reminder that it must be reviewed by the responsible person. Personal data of customers are kept out of the models and protected by access control. The authors also commit to reporting results as obtained, including poor ones.

XII. Conclusion and Future Scope

This paper presented the design of SKE IntelliPower, which combines a company website, secure project records, a dashboard and ML-based estimation, delay-risk classification and similar-project retrieval for an electrical contracting firm. The approach relies on well-established, interpretable methods and a transparent evaluation plan. Its usefulness will depend on the quality and quantity of the historical data, and results will be reported only after testing.

Several extensions are possible once the core system is stable.

1) Mobile site updates with photograph upload, so that supervisors can report progress from the site.
2) Notifications by message or e-mail when a project falls behind schedule or enters a high-risk class.

All rights reserved by www.ijsrd.com

3) Inventory tracking, linking material stock to project needs so that shortages are seen before they cause delay.
4) Report export of project summaries and quotations in printable form.
5) Better forecasting that takes material price changes and season into account.
6) A chat assistant that answers common questions about services and past work.

As more projects are recorded, the models can be retrained and compared with the live error measured by the platform. The same design could also be adapted by other small contractors in related trades, such as plumbing or interior fit-out, by changing the project types and features.

ACKNOWLEDGMENT

The authors thank Prof. Vaishnavi Kumbhar for guidance, and Shrikrishana Electricals for the business context for this project.

REFERENCES

[1] B. Flyvbjerg, M. S. Holm, and S. Buhl, "Underestimating costs in public works projects: Error or lie?," J. Amer. Plan. Assoc., vol. 68, no. 3, pp. 279–295, 2002.
[2] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK Guide), 6th ed. Newtown Square, PA: PMI, 2017.
[3] M. Gondia, A. Siam, W. El-Dakhakhni, and A. H. Nassar, "Machine learning algorithms for construction projects delay risk prediction," J. Constr. Eng. Manage., vol. 146, no. 1, Art. no. 04019085, Jan. 2020.
[4] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, Oct. 2001.
[5] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2nd ed. New York: Springer, 2009.
[6] K. Spärck Jones, "A statistical interpretation of term specificity and its application in retrieval," J. Doc., vol. 28, no. 1, pp. 11–21, 1972.
[7] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Inf. Process. Manage., vol. 24, no. 5, pp. 513–523, 1988.
[8] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[9] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, Univ. California, Irvine, 2000.
[10] M. Grinberg, Flask Web Development, 2nd ed. Sebastopol, CA: O'Reilly Media, 2018.
[11] R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 7th ed. Hoboken, NJ: Pearson, 2016.
[12] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, Oct. 2011.
[13] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.
[14] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature," Geosci. Model Dev., vol. 7, no. 3, pp. 1247–1250, 2014.
[15] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," J. Mach. Learn. Technol., vol. 2, no. 1, pp. 37–63, 2011.

All rights reserved by www.ijsrd.com