International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802067

# Decentralized Predictive Intelligence for Multi-Tenant Task Platforms using Federated Optimization

**Mr.G.Kannan[1], Kesavan M[2], Karthik P[3], Karthik GV[4]**

Assistant Professor, Department of Artificial Intelligence and Data Science, Department of Artificial Intelligence and Data Science, RP Sarathy Institute of Technology, Poosaripatti, Salem, Tamil Nadu, India[1]

Students, Department of Artificial Intelligence and Data Science, RP Sarathy Institute of Technology, Poosaripatti, Salem, Tamil Nadu, India[2,3,4]

**Publication History:** Received: 25.02.2026; Revised: 20.03.2026; Accepted: 25.03. 2026; Published: 28.03.2026.

**ABSTRACT:** Task management platforms are widely used to coordinate work across distributed teams. However, most existing systems focus primarily on monitoring workflows rather than predicting potential delays. In modern multi-tenant cloud environments, centralized machine learning approaches may also introduce privacy concerns because task data from multiple organizations must be aggregated.

This paper proposes a decentralized predictive intelligence framework for multitenant task management platforms using federated optimization. In the proposed system, each tenant trains a local machine learning model using its own task data while a global model is constructed through federated aggregation of model parameters. This approach allows organizations to benefit from collaborative learning while preserving data privacy. Experimental evaluation demonstrates that the proposed framework improves task delay prediction accuracy while maintaining strict data isolation across tenants.

**KEYWORDS:** Federated Learning, Multi-Tenant Architecture, Task Delay Prediction, Distributed Machine Learning, Cloud Computing

## I. INTRODUCTION

Digital task management platforms play a critical role in coordinating workflows within modern organizations. These platforms allow administrators to assign tasks, monitor progress, and allocate resources effectively across distributed teams. With the increasing adoption of cloud computing technologies, many modern enterprise applications are deployed in multi-tenant environments where multiple organizations share computational infrastructure while maintaining logical data isolation [1–3].

Despite their widespread adoption, most task management systems primarily focus on workflow monitoring and reporting rather than predictive intelligence. As projects become more complex and distributed, identifying potential task delays before they occur becomes increasingly important. Machine learning techniques provide an effective approach for analyzing historical task data and identifying patterns that influence task completion outcomes [4–6].

However, in multi-tenant cloud environments, collecting data from multiple organizations in a centralized repository raises privacy and governance concerns. Organizations may be reluctant to share sensitive operational data due to security and confidentiality constraints. Federated learning has emerged as a promising approach for collaborative machine learning in such distributed environments. Instead of transferring raw data, federated learning allows participants to train models locally and share only model parameters during the training process [7–10].

Recent studies have also highlighted several challenges in federated learning systems, including communication efficiency, heterogeneous data distributions, and security risks [11–13]. Despite these challenges, federated learning has demonstrated strong potential for enabling privacy-preserving predictive analytics across distributed systems.

In this paper, we propose a federated predictive intelligence framework designed for multi-tenant task management platforms. The proposed system enables multiple organizations to collaboratively train machine learning models for task delay prediction while preserving data privacy. By combining federated optimization with task analytics, the framework aims to improve workflow visibility and support proactive decision-making in distributed project management environments.

## II. RELATED WORK

Federated learning was introduced as a distributed machine learning framework that enables multiple participants to collaboratively train a shared model while keeping their training data decentralized [7]. In this paradigm, model updates are exchanged instead of raw datasets, which significantly reduces privacy risks associated with centralized machine learning approaches. Subsequent research has focused on improving the scalability and communication efficiency of federated learning systems. For example, strategies for optimizing communication between distributed clients and aggregation servers have been proposed to reduce training overhead in large-scale deployments [10,11].
Several studies have also examined the theoretical challenges and practical limitations

of federated learning. These include issues related to heterogeneous data distributions, system scalability, and model convergence across distributed clients [8,9,12]. In addition, security and privacy concerns such as data leakage through model updates have been extensively studied, leading to the development of more secure federated learning architectures [13]. Recent research has further extended federated learning to edge computing and hierarchical architectures, where model training can occur across multiple layers of distributed infrastructure including mobile devices, edge servers, and cloud environments [14–16].

Cloud computing provides the technological foundation for deploying large-scale distributed applications such as multi-tenant task management systems. Early studies highlighted the potential of cloud computing to deliver scalable, on-demand computing resources to organizations through shared infrastructure [1]. The National Institute of Standards and Technology (NIST) later defined cloud computing as a model that enables convenient and on-demand network access to a shared pool of configurable computing resources [2]. Market-oriented cloud computing frameworks further explored how distributed infrastructure can support enterprise applications operating in multi-tenant environments [3].

Machine learning algorithms have long been used for classification and predictive analytics across a variety of application domains. Decision tree–based models and ensemble methods have demonstrated strong performance for structured datasets with heterogeneous features [17]. Foundational machine learning concepts and methodologies have been widely discussed in classical literature [5,6,18]. In addition, practical guidelines for applying machine learning techniques in real-world applications have been explored in prior research [4].

Large-scale distributed machine learning systems have also been studied in the context of cloud computing infrastructures. Early work demonstrated how distributed computing environments could be used to train complex machine learning models across multiple machines [19]. These developments laid the groundwork for modern federated learning frameworks that operate across multiple independent organizations.

Although significant progress has been made in federated learning and distributed machine learning systems, relatively limited research has focused on integrating these techniques into collaborative task management platforms. This study addresses this gap by proposing a federated predictive intelligence framework for multi-tenant task management environments, enabling organizations to collaboratively train models for task delay prediction while preserving data privacy.

## III. SYSTEM ARCHITECTURE

The proposed framework adopts a federated learning architecture designed for multitenant task management platforms. In this architecture, multiple organizational tenants collaboratively train a predictive model while maintaining strict data isolation. The system consists of three main components: tenant nodes, a federated aggregation server, and a communication layer that coordinates the training process.

Each tenant represents an independent organization operating within the multi-tenant platform. Every tenant maintains its own local database containing historical task records such as task duration, priority level, workload allocation, and task complexity. These datasets remain within the tenant environment and are never shared directly with other participants.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802067

Local machine learning models are trained independently at each tenant node using the tenant's private dataset. During the training process, the local model learns patterns that influence task delay prediction. Instead of transferring raw data to a centralized repository, only the learned model parameters are transmitted to the federated aggregation server.

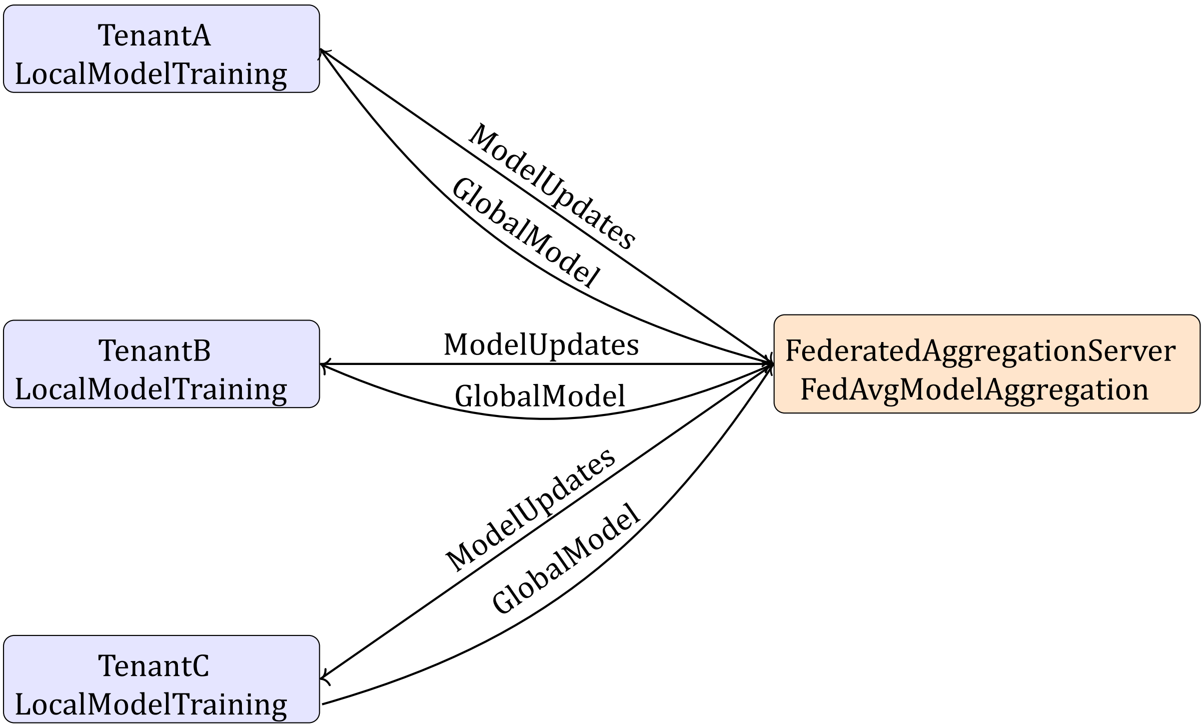



Figure 1: Federated learning architecture for multi-tenant task prediction. Each tenant trains a local model and sends model updates to the aggregation server. The server aggregates the updates using the FedAvg algorithm and redistributes the global model to the tenants.

As illustrated in Figure 1, the federated aggregation server coordinates the collaborative training process across all participating tenants. After receiving model updates from each tenant, the server aggregates the parameters using a federated optimization strategy, typically the Federated Averaging (FedAvg) algorithm. This aggregation step produces a global model that captures knowledge learned from multiple tenants.

Once the global model is generated, it is redistributed to all tenant nodes. Each tenant then updates its local model using the new global parameters and continues the next training round. Through multiple iterations of this process, the global model gradually improves while preserving the privacy of tenant data.

This architecture provides several advantages for multi-tenant task management systems. First, it ensures data privacy since raw task datasets remain within their respective organizations. Second, it allows collaborative learning across tenants, enabling the predictive model to capture broader workflow patterns. Finally, the decentralized training process reduces the risks associated with centralized data storage and improves the scalability of the predictive framework.

## IV. METHODOLOGY

The proposed predictive framework integrates machine learning techniques with a federated learning architecture to enable privacy-preserving task delay prediction in multitenant task management platforms. The methodology consists of four major phases: data preprocessing, feature representation, local model training, and federated model aggregation.

**Data Preprocessing**

Before model training, the collected task datasets are subjected to preprocessing procedures to improve data quality and ensure consistent model performance. The preprocessing stage includes handling missing values, encoding categorical attributes, and normalizing numerical features.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |
| Volume 8, Issue 2, March - April 2026 |
DOI:10.15662/IJEETR.2026.0802067

Task attributes such as priority levels are converted into numerical representations using label encoding. Numerical features such as task duration and workload allocation are scaled to maintain uniform feature distributions.

**Feature Representation**

Each task instance is represented as a feature vector containing relevant workflow attributes. Let

$$X_i = [d_i, p_i, w_i, c_i] \tag{1}$$

where
$d_i$ represents the estimated task duration
$p_i$ represents task priority
$w_i$ represents workload allocation
$c_i$ represents task complexity

The target variable $y_i$ represents the task completion outcome and is defined as

$$y_i = \begin{cases} 1, & \text{if the task is delayed} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The objective of the predictive model is to learn a function $f(X_i)$ that maps task attributes to delay predictions.

**Local Model Training**

Each tenant node independently trains a local machine learning model using its private dataset. In this study, three supervised learning algorithms were evaluated: Logistic Regression, Decision Tree, and Naive Bayes.
Given a dataset $D_k$ belonging to tenant $k$, the local loss function is defined as

$$F_k(w) = \frac{1}{n_k} \sum_{i=1}^{n_k} L(f(X_i; w), y_i) \tag{3}$$

where $n_k$ denotes the number of samples in tenant $k$ and $L(\cdot)$ represents the classification loss function.

Local training is performed using iterative optimization procedures that update the model parameters $w$ based on the training data.

## V. FEDERATED OPTIMIZATION

Federated optimization enables multiple tenants to collaboratively train a global predictive model without sharing their local datasets. Instead of transferring raw task records to a centralized server, each tenant performs local training and sends model updates to a federated aggregation server.

**Global Objective Function**

Assume that the federated system consists of $K$ tenants. Each tenant $k$ contains a dataset with $n_k$ training samples. The global optimization objective can be defined as

$$F(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \tag{4}$$

Where $K$ $k$=1

$$n = \sum n_k \tag{5}$$

represents the total number of samples across all tenants.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802067

**Federated Averaging**

The federated learning process follows the Federated Averaging (FedAvg) algorithm. During each training round, tenants receive the global model parameters, perform local training, and return their updated parameters to the aggregation server.

The global model parameters are updated using

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k \tag{6}$$

where $w_k$ represents the locally trained model parameters from tenant $k$.

**Federated Training Workflow**

The federated training process is executed iteratively across multiple rounds until model convergence. The procedure is summarized as follows:

1. Initialize the global model parameters.
2. Distribute the global model to all tenants.
3. Each tenant performs local training using its private dataset.
4. Local model updates are transmitted to the aggregation server.
5. The server aggregates model updates using the FedAvg algorithm.
6. The updated global model is redistributed to tenants.
7. Steps 3–6 are repeated until convergence.

Through this iterative collaborative learning process, the global predictive model gradually improves while maintaining data privacy across participating organizations.

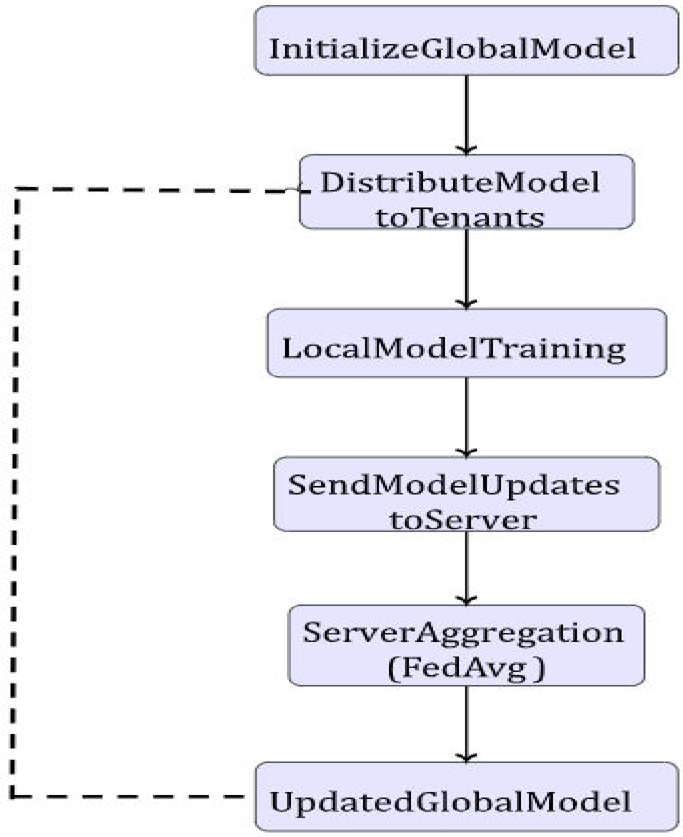



Figure 2: Federated training workflow illustrating the iterative process of global model distribution, local training at tenant nodes, parameter aggregation using the FedAvg algorithm, and global model updates.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802067

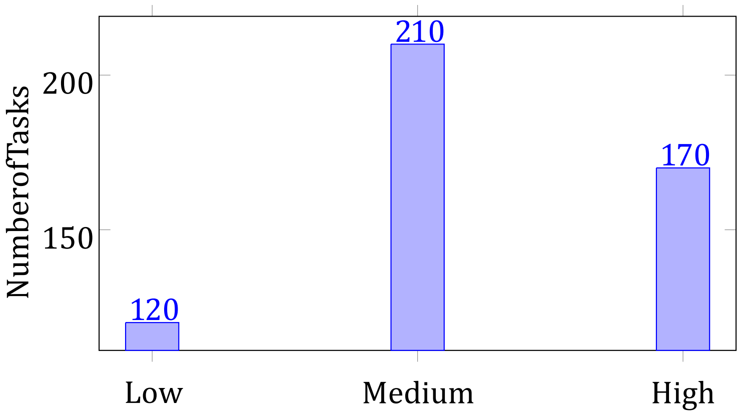
## VI. EXPERIMENTAL SETUP

This section describes the dataset used in the experiments, preprocessing procedures, machine learning models employed, and the evaluation metrics used to assess the performance of the proposed predictive framework.

### Dataset Description

The experimental evaluation was conducted using a dataset containing historical task records collected from a simulated multi-tenant task management platform. Each tenant maintains its own local dataset that contains task information generated during project workflow activities.
Each task record includes several attributes that influence task completion. The features used for training the predictive models are described below:

- **Task Duration (*d*):** Estimated time required to complete a task.
- **Task Priority (*p*):** Priority level assigned to the task.
- **Workload Allocation (*w*):** Number of tasks currently assigned to the employee responsible for the task.
- **Task Complexity (*c*):** A numerical value representing the complexity of the task.
- The output variable represents whether the task experienced a delay. Tasks that exceeded their expected completion time were labeled as 1, while tasks completed within the expected time were labeled as 0.




Task Priority Level

Figure 3: Distribution of tasks across different priority levels in the dataset used for training predictive models.

### Data Preprocessing

Before model training, several preprocessing steps were applied to improve data quality and ensure consistent model performance.

- Missing values were handled using appropriate data cleaning techniques.
- Numerical attributes were normalized to ensure uniform scaling.
- Categorical variables such as priority level were encoded into numerical representations.
- The dataset was divided into training and testing sets using a 70:30 ratio.

These preprocessing steps allow the models to learn meaningful patterns from the task data.

### Machine Learning Models

Three classification algorithms were implemented to predict task delays. These algorithms were selected due to their effectiveness in handling structured datasets and classification problems.

### Decision Tree

Decision Tree is a supervised learning algorithm that constructs a tree-like model of decisions based on feature values. The model splits the dataset into subsets according to feature conditions, allowing it to learn interpretable decision rules for classification.

### Naive Bayes

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes independence between features and calculates the probability that a given task belongs to a particular class.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802067

**Logistic Regression**

Logistic Regression is a statistical classification model that estimates the probability of a binary outcome. The model uses a logistic function to map input features to a probability value between 0 and 1.

**Evaluation Metrics**

The performance of the predictive models was evaluated using several standard classification metrics.

**Accuracy**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

**Precision**

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

**Recall**

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

**F1 Score**

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

These metrics provide a balanced evaluation of the predictive performance of each model.

**Implementation Environment**

The proposed predictive framework was implemented using the Python programming language due to its extensive ecosystem for data analysis and machine learning development. Python provides flexible tools for building predictive models and performing data preprocessing tasks efficiently.

Data preprocessing and dataset manipulation were performed using the *Pandas* library, which provides data structures for handling structured datasets. Numerical computations and matrix operations required for machine learning algorithms were carried out using the *NumPy* library.

The machine learning models used in this study, including Decision Tree, Naive Bayes, and Logistic Regression, were implemented using the *Scikit-learn* library. This library provides well-established implementations of classification algorithms and evaluation utilities for supervised learning tasks.

The development and experimentation process was conducted using the *Jupyter Notebook* environment, which allows interactive execution of code, visualization of intermediate results, and documentation of experimental workflows. Visualization of results such as model performance comparisons and evaluation metrics was performed using the *Matplotlib* and *Seaborn* libraries. The federated learning workflow was simulated by dividing the dataset into multiple tenant partitions representing independent organizational participants. Each partition was treated as a local dataset for model training. During each federated training round, locally trained model parameters were aggregated to construct a global predictive model.

The implementation environment was configured to ensure reproducibility of the experiments, allowing consistent evaluation of the machine learning models across different training iterations.

## VII. RESULTS AND DISCUSSION

This section presents the experimental results obtained from the predictive models implemented for task delay prediction in the multi-tenant task management platform. The performance of three machine learning algorithms, namely Logistic Regression, Decision Tree, and Naive Bayes, was evaluated using validation accuracy.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802067

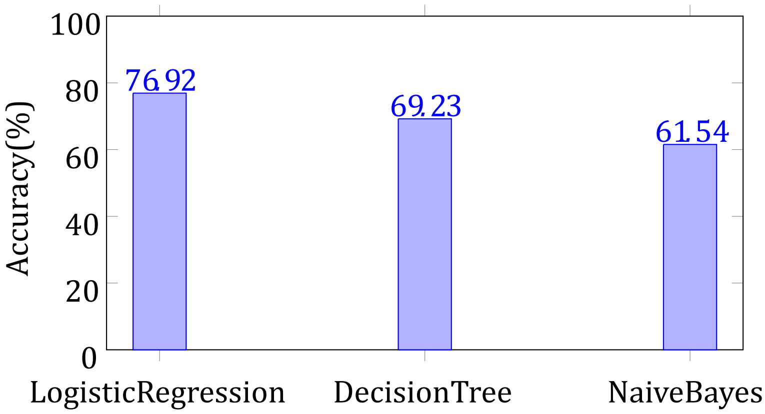
The models were trained on historical task records obtained from the system database. Each tenant dataset was divided into training and validation subsets in order to evaluate the generalization ability of the predictive models.

**Model Performance**

Table 1 summarizes the validation accuracy achieved by each algorithm.

Table 1: Performance Comparison of Prediction Models

| Model | Validation Accuracy |
|---|---|
| Logistic Regression | 76.92% |
| Decision Tree | 69.23% |
| Naive Bayes | 61.54% |




Machine Learning Models

Figure 4: presents a visual comparison of the prediction accuracy achieved by the evaluated machine learning models. Logistic Regression demonstrates superior performance with an accuracy of 76.92%, outperforming Decision Tree and Naive Bayes classifiers.

The experimental results show that Logistic Regression achieved the highest validation accuracy of 76.92%. This indicates that the relationship between the task features and the delay outcome can be effectively captured using a linear classification model. Logistic Regression models the probability of task delay using a logistic function, which allows it to handle binary classification problems efficiently.

The Decision Tree model achieved an accuracy of 69.23%. Decision Trees are capable of capturing nonlinear relationships between features and the target variable by creating hierarchical decision rules. However, the performance of tree-based models may depend heavily on dataset size and feature distribution. In this case, the model performed moderately well but slightly lower than Logistic Regression.

The Naive Bayes classifier produced the lowest accuracy of 61.54%. This outcome can be explained by the underlying assumption of the Naive Bayes algorithm, which assumes that input features are independent of each other. In practical task management scenarios, features such as workload allocation, task complexity, and estimated duration often exhibit correlations. This dependency between features reduces the effectiveness of the Naive Bayes model for this prediction task.

**Analysis of Predictive Features**

The results also suggest that certain task attributes contribute significantly to predicting task delays. Features such as task complexity and workload allocation have a strong influence on prediction outcomes. Tasks with higher complexity levels and increased workload distribution were more likely to experience delays.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802067

Task priority was also found to play an important role. High-priority tasks are often completed within expected deadlines due to increased monitoring and attention from project managers, while lower-priority tasks may be postponed when resources are limited.

**Impact of the Federated Learning Framework**

The predictive models were integrated within a federated learning framework designed for multi-tenant task management systems. Instead of aggregating raw data across organizations, each tenant trains a local predictive model using its own dataset. Only the model parameters are shared during the aggregation process.

This decentralized learning approach provides several advantages. First, it preserves organizational data privacy since sensitive task information remains within each tenant environment. Second, the global model benefits from knowledge learned across multiple tenants, enabling it to capture broader workflow patterns.

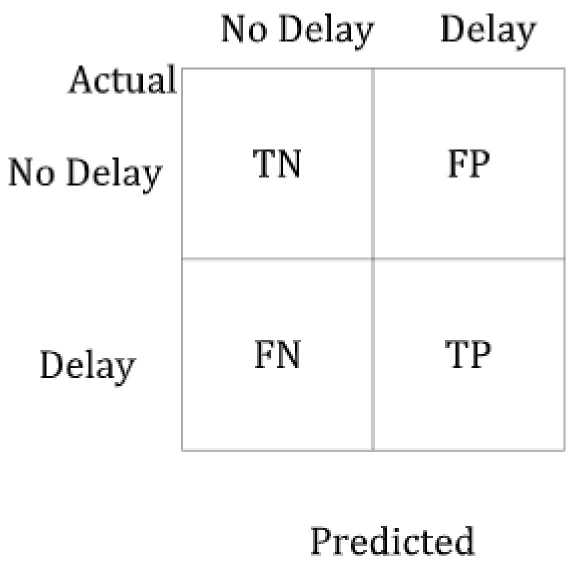



Figure 5: Confusion matrix representation for task delay prediction using the Logistic Regression model.

The results demonstrate that combining machine learning techniques with federated optimization can provide effective predictive capabilities for distributed task management systems.

**Discussion**

Overall, the experimental findings confirm that machine learning models can successfully identify patterns that influence task completion outcomes. Among the evaluated algorithms, Logistic Regression provided the best predictive performance for the task delay prediction problem.

The federated learning architecture further enhances the practicality of the proposed system by enabling collaborative model training across multiple organizations without compromising data privacy. This approach is particularly suitable for multi-tenant cloud environments where organizations share infrastructure but must maintain strict data isolation.
By integrating predictive analytics into task management platforms, project managers can proactively detect potential workflow delays and take corrective actions before project deadlines are affected.

## VIII. CONCLUSION

This research presented a decentralized predictive intelligence framework for multi-tenant task management platforms using federated optimization. The primary objective of the study was to integrate machine learning techniques into task management systems in order to predict potential task delays while preserving data privacy across participating tenants.
The proposed framework enables each tenant to train a local predictive model using its own task dataset while participating in a collaborative learning process through federated model aggregation. This approach eliminates the

need to share sensitive organizational data while still allowing the system to benefit from knowledge learned across multiple tenants.

Experimental evaluation was conducted using three classification algorithms: Logistic Regression, Decision Tree, and Naive Bayes. The results showed that Logistic Regression achieved the highest validation accuracy of 76.92%, demonstrating its effectiveness for predicting task delays based on attributes such as task duration, priority, workload allocation, and task complexity. Decision Tree produced moderate performance, while Naive Bayes achieved lower accuracy due to its feature independence assumption.

The findings indicate that predictive analytics can significantly enhance the capabilities of modern task management platforms. By identifying tasks that are likely to experience delays, the system can assist project managers in taking proactive actions to improve workflow efficiency and resource allocation.

## IX. FUTURE ENHANCEMENT

Although the proposed framework demonstrates promising results, several opportunities exist for further improvement and extension.

First, future work can explore the integration of advanced machine learning techniques such as ensemble models and deep learning architectures. These models may capture more complex relationships between task attributes and improve prediction accuracy.

Second, the federated learning framework can be enhanced by implementing secure aggregation and privacy-preserving techniques such as differential privacy and homomorphic encryption. These methods would further strengthen data protection in collaborative learning environments.

Third, future research can evaluate the proposed system using larger real-world datasets collected from enterprise task management platforms. This would allow the framework to be tested under more complex workflow scenarios involving heterogeneous data distributions.

Finally, the predictive framework can be integrated with automated task scheduling and resource optimization mechanisms. Such integration would allow the system not only to predict task delays but also to recommend corrective actions for improving project timelines and workload distribution.

## X. LIMITATIONS OF THE STUDY

Although the proposed federated predictive framework demonstrates promising results for task delay prediction in multi-tenant task management platforms, several limitations should be acknowledged.

First, the experimental evaluation was conducted using a relatively limited dataset derived from simulated task management records. While the dataset captures typical workflow attributes such as task duration, priority level, workload allocation, and task complexity, larger and more diverse datasets may reveal additional patterns that could further improve model performance.

Second, the machine learning models implemented in this study primarily consist of traditional classification algorithms such as Logistic Regression, Decision Tree, and Naive Bayes. Although these models provide interpretable and computationally efficient predictions, more advanced models such as ensemble learning methods or deep neural networks may achieve higher predictive accuracy in complex workflow environments.

Third, the federated learning process in this study was simulated within a controlled environment by partitioning the dataset into multiple tenant segments. In real-world deployments, federated learning systems may encounter additional challenges such as network latency, client availability variations, and heterogeneous data distributions across participating organizations.

Finally, the current framework focuses primarily on predicting task delays rather than automatically optimizing task allocation or workflow scheduling. Integrating predictive analytics with automated decision-making mechanisms could further enhance the practical impact of the system.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802067

Despite these limitations, the proposed framework provides a foundation for developing privacy-preserving predictive analytics systems in multi-tenant task management environments.

**REFERENCES**

1. M. Armbrust *et al.*, “A view of cloud computing,” *Communications of the ACM*,
2. 2010.
3. P. Mell and T. Grance, “The nist definition of cloud computing,” National Institute of Standards and Technology, Tech. Rep., 2011.
4. R. Buyya *et al.*, “Market-oriented cloud computing,” *Future Generation Computer Systems*, 2009.
5. P. Domingos, “A few useful things to know about machine learning,” *Communications of the ACM*, 2012.
6. T. Mitchell, *Machine Learning*.McGraw Hill, 1997.
7. C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
8. H. B. McMahan *et al.*, “Communication-efficient learning of deep networks from decentralized data,” *Proceedings of AISTATS*, 2017.
9. P. Kairouz *et al.*, “Advances and open problems in federated learning,” *Foundations and Trends in Machine Learning*, 2021.
10. T. Li *et al.*, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Processing Magazine*, 2020.
11. K. Bonawitz *et al.*, “Towards federated learning at scale: System design,” *Proceedings of MLSys*, 2019.
12. C.Nagarajan and M.Madheswaran - ‘Stability Analysis of Series Parallel Resonant Converter with Fuzzy Logic Controller Using State Space Techniques’- Taylor &Francis, Electric Power Components and Systems, Vol.39 (8), pp.780-793, May 2011. DOI: 10.1080/15325008.2010.541746
13. C.Nagarajan and M.Madheswaran - ‘Experimental verification and stability state space analysis of CLL-T Series Parallel Resonant Converter’ - Journal of Electrical Engineering, Vol.63 (6), pp.365-372, Dec.2012. DOI: 10.2478/v10187-012-0054-2
14. C.Nagarajan and M.Madheswaran - ‘Performance Analysis of LCL-T Resonant Converter with Fuzzy/PID Using State Space Analysis’- Springer, Electrical Engineering, Vol.93 (3), pp.167-178, September 2011. DOI 10.1007/s00202-011-0203-9
15. S.Tamilselvi, R.Prakash, C.Nagarajan,“Solar System Integrated Smart Grid Utilizing Hybrid Coot-Genetic Algorithm Optimized ANN Controller” Iranian Journal Of Science And Technology-Transactions Of Electrical Engineering, DOI10.1007/s40998-025-00917-z,2025
16. S.Tamilselvi, R.Prakash, C.Nagarajan,“ Adaptive sliding mode control of multilevel grid-connected inverters using reinforcement learning for enhanced LVRT performance” Electric Power Systems Research 253 (2026) 112428, doi.org/10.1016/j.epsr.2025.112428
17. S.Thirunavukkarasu, C. Nagarajan, 2024, “Performance Investigation on OCF and SCF study in BLDC machine using FTANN Controller," Journal of Electrical Engineering And Technology, Volume 20, pages 2675–2688, (2025), doi.org/10.1007/s42835-024-02126-w
18. C. Nagarajan, M.Madheswaran and D.Ramasubramanian- ‘Development of DSP based Robust Control Method for General Resonant Converter Topologies using Transfer Function Model’- Acta Electrotechnica et Informatica Journal , Vol.13 (2), pp.18-31,April-June.2013, DOI: 10.2478/aeei-2013-0025.
19. C.Nagarajan and M.Madheswaran - ‘DSP Based Fuzzy Controller for Series Parallel Resonant converter’- Springer, Frontiers of Electrical and Electronic Engineering, Vol. 7(4), pp. 438-446, Dec.12. DOI 10.1007/s11460-012-0212-0.
20. C.Nagarajan and M.Madheswaran - ‘Experimental Study and steady state stability analysis of CLL-T Series Parallel Resonant Converter with Fuzzy controller using State Space Analysis’- Iranian Journal of Electrical & Electronic Engineering, Vol.8 (3), pp.259-267, September 2012.
21. C.Nagarajan and M.Madheswaran, “Analysis and Simulation of LCL Series Resonant Full Bridge Converter Using PWM Technique with Load Independent Operation” has been presented in ICTES’08, a IEEE / IET International Conference organized by M.G.R.University, Chennai.Vol.no.1, pp.190-195, Dec.2007
22. Suganthi Mullainathan, Ramesh Natarajan, “An SPSS and CNN modelling based quality assessment using ceramic materials and membrane filtration techniques”, Revista Materia (Rio J.) Vol. 30, 2025, DOI: https://doi.org/10.1590/1517-7076-RMAT-2024-0721
23. M Suganthi, N Ramesh, “Treatment of water using natural zeolite as membrane filter”, Journal of Environmental Protection and Ecology, Volume 23, Issue 2, pp: 520-530,2022
24. J. Koneˇcny´ *et al.*, “Federated learning: Strategies for improving communication efficiency,” *arXiv preprint*, 2016.
25. Y. Zhao *et al.*, “Federated learning with non-iid data,” *arXiv preprint*, 2018.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802067

26. A. Mothukuri *et al.*, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, 2021.
27. Y. Liu *et al.*, "Client edge cloud hierarchical federated learning," *IEEE Transactions on Network Science*, 2021.
28. T. Yang *et al.*, "Federated learning in mobile edge computing," *IEEE Internet of Things Journal*, 2022.
29. Q. Li *et al.*, "Federated learning systems: Vision, hype and reality," *arXiv preprint*, 2022.
30. L. Breiman, "Random forests," *Machine Learning*, 2001.
31. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
32. J. Dean *et al.*, "Large scale distributed deep networks," in *NeurIPS*, 2012.