B.Tech 2020-24 CSE- Project Phase 1

Proposal

I. Group No: D6

Project Title.: Efficient Task Allocation and Failure Detection in Crowdsourcing Platforms

Team members :

Roll No	Name
AM.EN.U4CSE20326	GELAM SINDHU SAHITHI
AM.EN.U4CSE20370	THOTA MANIKANTA
AM.EN.U4CSE20371	T D M N SURYA TEJA
AM.EN.U4CSE20375	VOLETI SAI ANUSHREE

II. Abstract

This project aims to tackle the challenges of task allocation in crowdsourcing platforms, ensuring efficient assignment of tasks to workers based on their skills and task difficulty. Additionally, the project seeks to develop a failure detection system using machine learning to identify patterns associated with task failures. The problem is significant due to the increasing popularity of crowdsourcing platforms and the need to improve efficiency and quality by reducing task allocation errors and preventing task failures. The motivation lies in enhancing platform efficiency, client satisfaction, and freelancer success. Persisting challenges include designing an accurate task allocation algorithm and creating a robust failure detection system that adapts to varying worker behaviors.

Background Study

Title &year	Problem	Contributions	Limitations	Open problems/Future work
Aggregating Reliable Submissions in Crowdsourcing Systems [Ayswarya R Kurup; G P Sajeev; J. Swaminathan, (2021)]	In crowdsourcing systems, the quality of submissions from workers can vary due to differences in expertise and knowledge background. Existing task aggregation methods mainly focus on structured submissions and do not consider the cost incurred for completing tasks. Additionally, probabilistic methods for answer aggregation may be sensitive to sparsity in the data.	Estimating submission quality based on worker reliability, task difficulty, and similarity. EM approach is used to improve results. The method effectively estimates submission quality and addresses the cold-start problem by leveraging submission similarity. Comparisons with state-of-the-art techniques validate its effectiveness, emphasizing the importance of worker and submission features in inferring reliable answers.	Adaptive threshold for cost minimization and improved expertness estimation using diverse similarity approaches. Future research could explore generalizing our approach to different crowdsourcing tasks.	Some of the Future work and problems for task aggregation in crowdsourcing: Explore adaptive cost minimization, refine worker expertness estimation with diverse similarity metrics, extend the approach to diverse task types, handle data sparsity, validate on real platforms, and improve user interface for better engagement and task completion rates.
Task Recommendation in Reward-Based [Ayswarya R Kurup	Addressing task recommendation challenges in dynamic crowdsourcing environments. Exploring hybrid models combining implicit and	This paper introduces a task recommendation model for reward-based crowdsourcing, combining implicit feedback and explicit features. The model outperforms matrix factorization and reward-based models, reducing	Limited evaluation on larger datasets and comparison with state-of-the-art methods. Generalization to diverse crowdsourcing platforms and investigation of the model's scalability remain unexplored.	Some of the Future work and problems for Task Recommendation in Reward-Based are Extending evaluation on larger datasets and comparing with

,G P Sajeev(2023)]	explicit feedback for improved accuracy. Investigating user interface and user experience enhancements to encourage worker engagement. Adapting the model to handle different types of crowdsourcing tasks. Evaluating real-world deployment and validation on various crowdsourcing platforms.	data sparsity. Utilizing participation data and worker-task feature vectors enhances reward gain prediction. Further, incorporating additional features could improve recommendation accuracy.	Further analysis required to handle cold-start problem for new workers effectively.	state-of-the-art methods. Generalizing the model for diverse crowdsourcing platforms and addressing scalability concerns. Handling cold-start problem for new workers effectively.
Outlier Detection for Streaming Task Assignment in Crowdsourcing [Yan Zhao; Xuanhao Chen; Liwei Deng et al.(2022)]	The proposed framework's problem is the lack of validation in real-world crowdsourcing platforms, limiting its practical applicability and understanding of its performance under diverse and dynamic crowdsourcing environments.	The paper proposes an efficient outlier detection framework for streaming task assignment in crowdsourcing, considering both malicious workers and invalid tasks. It introduces a SA-Generative Adversarial Network (GAN) outlier detector to identify outliers in the worker and task multivariate time series, and it addresses the class-imbalanced problem using negative sampling. The proposed framework shows improvements in	The proposed outlier detection framework's limitations include the lack of real-world deployment, limited exploration of anomalies beyond malicious workers and invalid tasks, scalability concerns for large crowdsourcing platforms, adaptability to evolving dynamics, potential overfitting, impact on task assignment, and the need for user feedback for refinement.	Real-world deployment and validation, exploring diverse anomalies, using semi-supervised or active learning, evaluating under various streaming scenarios, integrating contextual information to enhance outlier detection, and improving task assignment efficiency.

		outlier detection accuracy, task assignment accuracy, and computational efficiency.		
Failure Prediction in Crowdsourced Software Development [Abdullah Khanfor, Ye Yang, Gregg Vesonder, Dave Messinger]	This paper highlights the need to address non-competitive crowdsourcing tasks and explore additional metrics for failure prediction, while considering potential internal threats in the evaluation process.	This paper proposes a failure prediction framework for software crowdsourcing, achieving high accuracy using machine learning. It identifies influencing factors for failure prediction and presents practical recommendations for managing task failure risks.	The study focuses only on competitive tasks on the Top Coder platform, overlooking non-competitive or collaborative tasks. Some competition factors and metrics need further investigation, and internal threats should be considered.	Future work includes studying supply and demand for technologies and workers, exploring social network analysis, and gathering broader data for evaluation. Text mining approaches to understand task descriptions' impact on failure prediction are also suggested.
Task Personalization for Inexpertise Workers in Incentive Based Crowdsourcing Platforms (Ayswarya R Kurup,G P Sajeev; Dept of Computer Science 2023)	The paper presents a task recommendation model for inexpert workers in crowdsourcing, but it needs further exploration to handle multiple skills and enhance the recommendation accuracy.	The paper proposes a task recommendation model for inexpert and new workers in crowdsourcing systems, using skill taxonomy and participation probability of expert workers.	This Model's limitation: Assessing workers and tasks with single skills may overlook real-world complexity. Future work includes multi-skill considerations, improved accuracy, and advanced algorithms to expand the model's utility across diverse crowdsourcing platforms.	Future work includes extending the model to handle multiple skills, improving the recommendation accuracy, and exploring more complex recommendation algorithms.

III. Challenges

Some of the challenges that still exist in task allocation and failure detection in crowdsourcing platforms are:

Dynamic Worker Skills: Worker skills can change over time, and new workers with unique expertise join the platform. The challenge lies in continuously updating the task allocation algorithm to accommodate these dynamic changes and ensure accurate matching of tasks with suitable workers.

Complex Task Dependencies: Some tasks may have dependencies on others, making it crucial to consider task sequences and interdependencies during allocation. Handling complex task structures requires advanced algorithms and efficient data management to avoid task bottlenecks and ensure smooth workflow distribution.

Worker Behavior Analysis: Detecting task failures requires understanding various worker behavior patterns associated with potential failures. Developing a failure detection system that can accurately differentiate between genuine difficulties and poor-quality work poses a challenge, as behavior patterns can be diverse and context dependent. Analyzing and interpreting worker behavior data effectively is essential to create a robust failure detection mechanism.

IV. Deliverables of Phase I

Objectives:

- Designing a task allocation algorithm that considers the skills and abilities of workers, as well as the difficulty of tasks, to optimize task assignments on crowdsourcing platforms.
- Develop a system for detecting task failures using machine learning techniques to identify patterns of worker behaviour associated with unsuccessful task completion.
- Improve the efficiency, reliability, and quality of crowdsourcing platforms by addressing task allocation challenges and implementing effective failure detection mechanisms.

Outcomes/Deliverables

1. An efficient task allocation algorithm that considers worker skills, workload distribution, and task complexity to optimize task assignments on crowdsourcing platforms.

2. A machine learning-based task failure detection system that analyses worker behaviour patterns and identifies early signs of potential task failures, enabling timely intervention and resolution.
3. Experimental results and performance evaluation of the task allocation algorithm and failure detection system, demonstrating their effectiveness in improving platform efficiency and work quality.
4. Implementation of interventions and remediation strategies based on detected task failures, leading to enhanced client satisfaction and freelancer success on the crowdsourcing platform.
5. A comprehensive project report detailing the methodology, findings, and recommendations for future improvements in task allocation and failure detection mechanisms on crowdsourcing platforms.

V. Assumptions/Declarations:

1. Crowdsourcing Platform: It is assumed that the crowdsourcing platform in question is a well-established and reliable platform with a significant user base. This platform allows users to participate in various tasks, and there are mechanisms in place to collect data on task performance and outcomes.
2. Task Types: The tasks on the platform are assumed to be well-defined and can be categorized into specific types or domains. This assumption is essential for designing effective task allocation and failure detection strategies based on the characteristics of different task types.
3. Task Execution Metrics: The dataset assumes the availability of relevant task execution metrics, such as completion time, accuracy, complexity, and any other performance-related indicators. These metrics are crucial for evaluating the success or failure of tasks and for identifying potential patterns.
4. User Profile Information: The dataset is assumed to include user profile information, such as user demographics, past performance history, experience, and expertise. This information is crucial for effective task allocation and matching suitable tasks to appropriate users.
5. Failure Definition: There is a clear definition of what constitutes a failure in the context of the crowdsourcing platform. This definition might vary based on task type, and it is important to have a common understanding of failure for accurate failure detection.

The dataset may include the following:

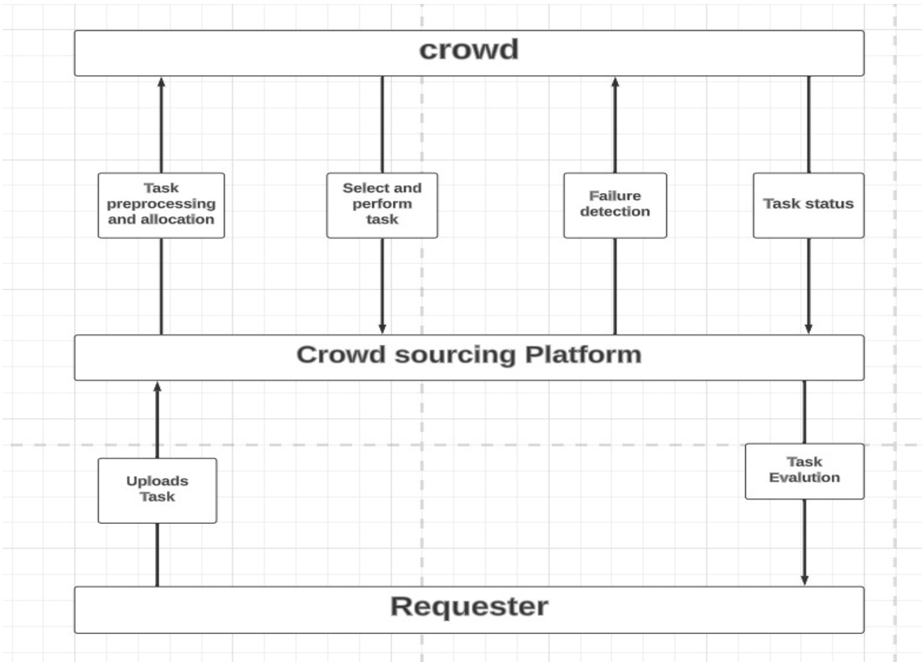
1. Task Information: Information about each task, such as task type, task description, complexity level, reward, and deadline.

2. User Information: Profiles of the users participating in the crowdsourcing platform, including demographics, past performance metrics, experience, and expertise in different task domains.
3. Task Execution Data: Data capturing the execution of tasks by users, including the start time, end time, completion status, and any intermediate progress.
4. Task Outcome: The outcome of each task, whether it was successfully completed or marked as a failure.
5. User Feedback and Ratings: Feedback provided by task requesters or other users on the quality of completed tasks.
6. Task Allocation Strategy: If available, data on the previous task allocation strategies employed by the platform.
7. Task Failure Causes: Any available data or annotations related to the reasons behind task failures.

VI. Tools to be used

Software/Hardware Tools	**Specifications**
Programming Language: Python Machine Learning Libraries: Scikit-learn, TensorFlow, Keras Data Analysis Libraries: Pandas, NumPy	A Standard Computer or Laptop with a decent CPU and GPU The project can be developed on Windows, macOS, or Linux operating systems. Python version 3.x is recommended for compatibility with the required libraries.
Web Framework: Flask or Django (if developing a web-based platform) Database Management System: SQLite or MySQL (for storing worker and task data) Integrated Hardware Requirements: Processor: Intel Core i5 or higher, RAM: 8 GB or higher, Storage: At least 256 GB SSD (Solid State Drive) or higher for faster data processing Graphics Processing Unit (GPU): NVIDIA GPU (optional, for faster training of machine learning models)	The machine learning models may benefit from a GPU for faster training, but they can also be trained on CPUs. The web-based platform, if developed, should be responsive and user-friendly for both workers and platform administrators. The database should be capable of efficiently storing and querying large volumes of worker and task data.

VII. High Level Design



Students' Name and Signature

NAME	SIGNATURE
GELAM SINDHU SAHITHI	
THOTA MANIKANTA	
T D M N SURYA TEJA	
VOLETI SAI ANUSHREE	

Guide's Signature

LEKSHMI S NAIR