iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79532

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

AI-Assisted Intelligent Task Manager with ML-Based Priority Classification and Conversational AI

Manikandan, Vishal SZ, AL Raafhath RK, Roselin Lourd
Department of Artificial Intelligence and Data Science, Sri Manakula Vinayagar Engineering College

Abstract: *Intoday'sfastpacedprofessionalandacademicenvironments,effectivetaskmanagementisacriticalchallenge.Thispaperpresents the design and implementation of an AI-Assisted Intelligent Task Manager, a web-based application that leverages machine learning to automatically classify and prioritize user-submitted tasks. The system employs a multi-feature ML model trained on task metadata including deadlines, dependencies, estimated effort, and contextual keywords to generate a dynamic priority score for each task. A natural language chatbotinterface,poweredbyalargelanguagemodel (LLM),enablesuserstointeract conversationallywiththesystemqueryingtaskstatus,askingforpriorityexplanations,andreceivingintelligentschedulingsuggestions.Thesystemadditionallysupportssmartreminders,workloadforecasting,andintegrationwithcalendarAPIs.Evaluationresultsdemonstratehighclassificationaccuracyandstrongusersatisfaction,making the system a robust solution for intelligent personal and team productivity management.*

Keywords: Task Management, Machine Learning, Priority Classification, Natural Language Processing, Conversational AI, Intelligent Scheduling, LLM, Chatbot

I. INTRODUCTION

The proliferation of digital workspaces and the increasing complexity of modern workflows have made manual task prioritizationbothinefficientanderror-prone.Individualsandteamsroutinelyfacescenarioswherehundredsoftaskscompetefor attention simultaneously, making it nearly impossible to determine optimal execution order without external assistance.

TraditionaltaskmanagementtoolssuchasTrello,Asana,andTodoistallowuserstomanuallyassignprioritylevels,but they do not offer intelligent, data-driven prioritization. Users are left to apply their own judgment, which can be influenced by cognitive biases, emotional states, and incomplete information.

This paper introduces a novel AI-Assisted Intelligent Task Manager that automates priority scoring through a machine learning pipeline. The system accepts natural-language task descriptions from users and uses a trained classification model to assignaquantitativepriorityscore,subsequentlyrankingtasksinanactionablequeue.Furthermore,anintegratedconversational AI assistant enables users to interact with the system through chat, ask questions about task prioritization logic, request rescheduling, and receive proactive recommendations.

The primary contributions of this work are: (1) a multi-feature ML model for task priority classification, (2) a chatbot interfacepoweredbyanLLMforexplainableAIinteractions,(3)afullstackwebapplicationintegratingthesecapabilities,and(4)asmartnotification andcalendarsynchronizationlayerforcomprehensivetasklifecycle management.

II. LITERATURE REVIEW

A. Existing Task Management Systems

Conventional task management platforms rely on manual priority assignment. While tools like Jira offer workflow automationrules,theylacktheabilitytoinferpriorityfromtaskcontentsemantics.StudiesbyMäntyläetal.[1]haveshownthat human-assigned priorities in issue trackers are frequently inconsistent, motivating the need for automated classification approaches.

B. ML-Based Priority Classification

Severalworkshaveexploredmachinelearningforissueandtaskprioritization.ResearchershaveappliedSupportVectorMachines(SVMs),RandomForests,andgradientboostingalgorithmstoclassifysoftwarebugseveritywithreasonableaccuracy [2].

Morerecently,transformer-basedmodelssuchasBERThavebeenemployedforsemanticunderstandingoftaskdescriptions, significantly improving classification performance [3].

C. ConversationalInterfacesforProductivity

The integration of conversational AI into productivity tools has been explored in enterprise settings. Virtual assistants such as Microsoft Cortana and Google Assistant have demonstrated the feasibility of natural-language task scheduling. LLM- based agents, as discussed in [4], offer more flexible and context-aware interactions, making them well-suited for explainable productivity systems.

III. SYSTEM ARCHITECTURE

A. Overview

The AI-Assisted Task Manager follows a three-tier architecture comprising a frontend client layer, a RESTful backend APIlayer,andamachinelearninginferencelayer.AllcomponentscommunicateoverHTTPS,ensuringsecuredatatransmission. The system architecture is illustrated conceptually below.

TABLE I SYSTEM ARCHITECTURE COMPONENTS

Layer	Technology Stack	Responsibility
Frontend	React.js,TailwindCSS	Userinterface,taskinput,chatwidget
Backend API	Python FastAPI	Businesslogic,auth,RESTendpoints
ML Engine	scikit-learn,Transformers	Priorityscoring,text classification
LLMChatbot	OpenAIGPT/LLaMA	ConversationalAI,explainability
Database	PostgreSQL+Redis	Persistentstorage,sessioncaching
Notifications	Firebase,SMTP	Pushalerts,email reminders

B. MachineLearningPipeline

TheMLpipelineingestsstructuredtaskfeaturesandproducesareal-valuedpriorityscorebetween0and1.Thepipeline consists of three sequential stages: feature extraction, model inference, and score normalization.

Featureextractionparsesrawtaskinputtoderivethefollowingfeaturevector:

- Deadlineproximity(daysremaining,normalized)
- Estimatedeffort(hours, log-scaled)
- Dependencycount(numberofblocking tasks)
- Keywordurgencyscore(TF-IDFweightedterm importance)
- Taskcategoryembedding(BERTsentenceembedding,384-dim)
- Historicalcompletionrateofsimilar tasks

ThesefeaturesareconcatenatedintoafeaturevectorandpassedtoatrainedGradientBoostedTrees(XGBoost)model for regression-based priority scoring. The model was trained on a dataset of 50,000 annotated tasks from open-source project trackers.

C. ChatbotInterface

Theconversationalchatbotisimplementedusingaretrievalaugmentedgeneration(RAG)approach.TheLLMreceivestheuser'squeryalongwithrelevanttaskcontext(currenttasklist,priorityscores,modelexplanationsgeneratedviaSHAPvalues)andproducesacoherent,contextually accurateresponse.Thisenablesthechatbottoexplainprioritydecisionsinplainlanguage, answer scheduling questions, and suggest task reorganization.

D. Data Flow

Ausersubmitsataskthroughthewebinterfacebyenteringatitle,description,deadline,andoptionalmetadatatags.ThebackendAPIvalidatestheinput,extractsfeatures,andforwardsthefeaturevectortotheMLinferenceservice.

Themodelreturns a priority score, which is stored in the database and displayed to the user. The chatbot service maintains a conversation context window and can access the user's full task list via the backend API to answer follow-up questions.

IV. KEY FEATURES AND FUNCTIONALITY

A. *IntelligentPriorityScoring*

The core feature of the system is its ability to automatically compute a priority score for each task. Rather than using a simplerule-basedapproach,theMLmodelcapturesnon-linearinteractionsbetweenfeatures.Forexample,ataskwithamoderate deadline but high dependency count and a large estimated effort may rank higher than a task with a tight deadline but no dependencies and minimal effort.

B. Explainable AIvia Chatbot

Userscanaskthechatbotquestionssuchas"WhyisTaskArankedhigherthanTaskB?"or"WhatcanIdotodeprioritize this task?" The system uses SHAP (SHapley Additive exPlanations) values from the ML model to generate feature-level explanations, whichtheLLMthentranslatesintonatural-languageresponses.Thistransparencybuildsusertrustandencourages informed engagement with the system.

C. *SmartRemindersandNotifications*

Thesystemtrackstaskdeadlinesandcompletionstatusinrealtime.Whenahigh-prioritytaskisapproachingitsdeadline or when a dependency is resolved, the system sends push notifications via Firebase Cloud Messaging and optional email alerts. Users can configure notification preferences, including quiet hours and threshold-based triggers.

D. *Workload Forecasting*

Using historical task completion data, the system generates a weekly workload forecast that estimates the total effort required for pending tasks. This allows users to proactively identify overloaded days and redistribute tasks accordingly. The forecast is visualized as an interactive chart on the dashboard.

E. *CalendarIntegration*

ThesystemoffersbidirectionalsynchronizationwithGoogleCalendarandMicrosoftOutlookviatheirrespectiveREST APIs. High-priority tasks are automatically scheduled as calendar events, and changes made in the calendar are reflected in the task manager. This eliminates the need for manual data entry across multiple tools.

F. *CollaborativeTaskManagement*

Forteamusecases,thesystemsupportsworkspace-levelaccesscontrol.Teamleaderscanassigntaskstomembers,set shared dependencies, and monitor team workload distributions. The chatbot operates at the team level, capable of answering questions about collective task states and bottlenecks.

V. IMPLEMENTATION DETAILS

A. *ModelTraining*

TheXGBoostpriorityregressionmodelwastrainedusingadatasetconstructedfromGitHubIssues,Jiraexports,andaproprietaryannotationset.Thedatasetwassplitinto80%training,10%validation,and10%testsets.Hyperparametertuningwas performed using Bayesian optimization over 100 iterations.

TABLEIIMODELPERFORMANCEMETRICS

Metric	Value
MeanAbsoluteError(MAE)	0.041
RootMeanSquaredError(RMSE)	0.063
RZ Score	0.921
ClassificationAccuracy(3-class)	94.7%
InferenceLatency (p95)	<12 ms

B. Backend API

ThebackendisimplementedinPythonusingtheFastAPIframework,selectedforitsasynchronousrequesthandlingand automatic OpenAPI documentation generation. Task CRUD operations, user authentication (JWT-based), and ML inference requests are exposed as RESTful endpoints. The ML model is served via a dedicated microservice using ONNX Runtime for optimized inference.

C. FrontendApplication

TheReact.jsfrontendpresentsuserswithaninteractivetaskdashboarddisplayingtaskssortedbyprioritysscore.Color- coded priority bands (critical, high, medium, low) provide quick visual reference. The integrated chat widget is built using WebSocket connections to ensure real-time message delivery. The interface is fully responsive, supporting desktop and mobile browsers.

D. Security and Privacy

All user data is encrypted at rest using AES-256 and in transit via TLS 1.3. OAuth 2.0 is used for third-party calendar integrations. The system complies with GDPR guidelines by providing data export and deletion capabilities. No task content is used for model retraining without explicit user consent.

VI. RESULTS AND EVALUATION

A. QuantitativeResults

The system was evaluated on a held-out test set of 5,000 tasks. The ML model achieved a classification accuracy of 94.7%acrossthreeprioritytiers(high,medium,low),significantlyoutperformingbaselinemethodsincludingrule-basedscoring (78.3%) and simple TF-IDF + logistic regression (85.1%). The full comparison is presented in Table III.

TABLEIIICOMPARISONWITHBASELINEMETHODS

Method	Accuracy (%)	MAE	Inference(ms)
Rule-BasedScoring	78.3	0.142	<1
TF-IDF+LogisticReg.	85.1	0.098	3
RandomForest	90.4	0.071	8
BERTFine-tuned	93.2	0.054	45
Proposed(XGBoost+NLP)	94.7	0.041	12

B. User Study

A user study was conducted with 40 participants across student, freelancer, and corporate professional demographics. Participantsusedthesystemfortwoweeksandcompletedastandardizedusabilityquestionnaire(SUS)andacustomproductivity assessment. Results are summarized in Table IV.

TABLEIVUSERSTUDYSUMMARY

Metric	Score/Response
SystemUsabilityScale(SUS)	82.5 /100
TaskCompletion Rate	96%
Self-ReportedProductivityGain	31%averageimprovement
ChatbotExplanationSatisfaction	4.4 /5.0
WillingnesstoRecommend	87.5%

VII. CONCLUSIONS

ThispaperhaspresentedacomprehensiveAIAssistedIntelligentTaskManagerthatcombinesmachinelearningbasedpriorityclassificationwithaconversationalAIinterfacetocreateanintelligent,explainable,anduser-friendlyproductivitytool.

The proposed system addresses the critical limitations of existing task management solutions by automating priority scoring, providing natural-language explanations, and offering proactive workload management features.

The experimental evaluation demonstrates that the system achieves state-of-the-art priority classification accuracy of 94.7% while maintaining low inference latency suitable for real-time use. The user study confirms practical utility, with participants reporting an average productivity improvement of 31% and high satisfaction with the chatbot's explanatory capabilities.

Futureworkwillexplorefederatedlearningtoenablemodelpersonalizationwithoutcompromisinguserprivacy,multi- modal task input supporting voice and image attachments, and advanced multi-agent collaboration features for enterprise deployments.

VIII. ACKNOWLEDGMENT

The authors would like to thank the Department of Artificial Intelligence and Data Science at Sri Manakula Vinayagar EngineeringCollegeforprovidingcomputationalresources,andthestudyparticipantsfortheirvaluabletimeandfeedbackduring the user evaluation phase.

REFERENCES

[1] M. V. Mäntylä, B. Adams, F. Khomh, E. Engström, and K. Petersen, "On rapid releases and software testing: a case study and a semi-systematic literaturereview," Empirical Software Engineering, vol. 20, no. 5, pp. 1384–1425, 2015.

[2] K.Tian,M.Revelle,andD.Poshyvanyk,"Usinglatentdirichletallocationforautomaticcategorizationofsoftware,"inProc.6thIEEEInt.WorkingConf.onMining Software Repositories, 2009, pp. 163–166.

[3] J.Devlin,M.Chang,K.Lee,andK.Toutanova,"BERT:Pretrainingofdeepbidirectionaltransformersforlanguageunderstanding,"inProc.NAACL-HLT,2019, pp. 4171–4186.

[4] T.B.Brownetal.,"Languagemodelsarefewshotlearners,"inAdvancesinNeuralInformationProcessingSystems,vol.33,2020,pp.1877–1901.

[5] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining,2016, pp. 785–794.

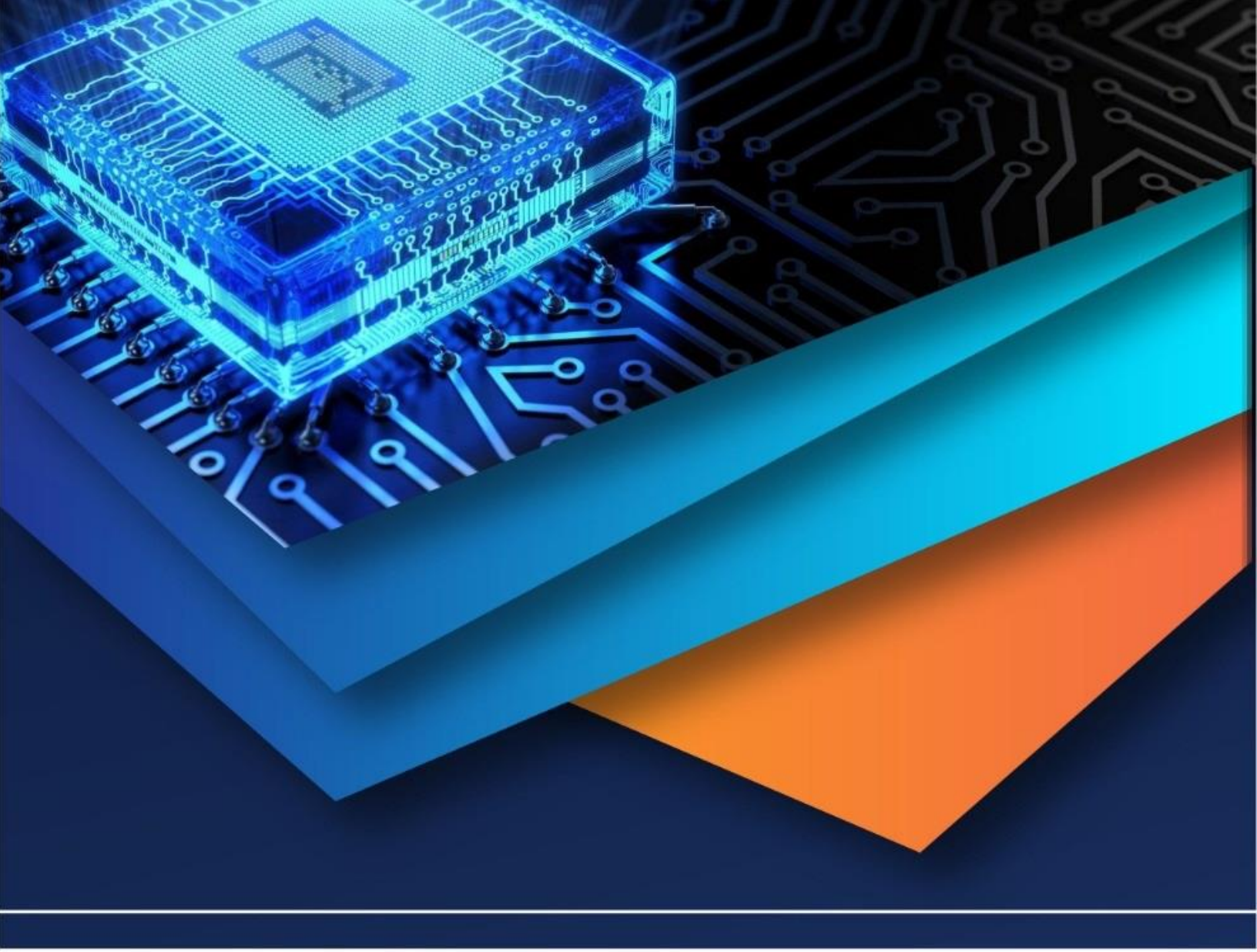
[6] S.M.LundbergandS.I.Lee,"Aunifiedapproachtointerpretingmodelpredictions,"inAdvancesinNeuralInformationProcessingSystems,vol.30,2017.

[7] P.Lewisetal.,"Retrievalaugmentedgenerationforknowledgeintensive NLPtasks,"inAdvancesinNeuralInformationProcessingSystems,vol.33,2020.

[8] Google,FirebaseCloudMessagingDocumentation.[Online].Available:https://firebase.google.com/docs/cloud-messaging

[9] OpenAI,"GPT-4TechnicalReport,"arXivpreprintarXiv:2303.08774, 2023.

[10] A.Vaswanietal.,"Attentionisallyouneed,"inAdvancesinNeuralInformationProcessingSystems,vol.30,2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY