International Journal of Scientific Research & Engineering Trends
Volume 11, Issue 4, July-Aug-2025, ISSN (Online): 2395-566X

Modellab: A Scalable AI Model Training Platform With Real-Time Training And Prediction Capabilities

Rushil Vijay Salian
Computer Science Engineering

Abstract- The growing demand for accessible, scalable, and automated machine learning (ML) platforms has highlighted the need for intuitive tools that democratize model development. ModelLab is a proprietary AI/ML model training platform that enables users to upload datasets, train models, and generate predictions through an interactive web interface. Built with React, Supabase, and TensorFlow.js, it supports end-to-end workflows from data ingestion to model deployment. This paper presents the architecture, features, and performance of ModelLab and explores its potential to revolutionize AI development for non-experts and professionals alike. Future work includes model versioning, expanded algorithm support, and production-level deployments.
Key words- machine learning, AutoML, web platform, TensorFlow.js, model training

I. INTRODUCTION

Machine Learning (ML) has become pivotal across industries, yet many organizations lack the tools or expertise to build models efficiently. ModelLab addresses this gap with a browser-based platform that simplifies dataset handling, model training, and prediction generation. It abstracts complex ML processes while preserving flexibility and transparency for more advanced users.

II. MOTIVATION AND BACKGROUND

Despite the abundance of ML libraries and frameworks, end-users face significant barriers in preprocessing data, tuning models, and deploying them at scale. Platforms like Google AutoML and Microsoft Azure ML offer advanced features but often present a steep learning curve or high cost. ModelLab aims to deliver core AutoML functionalities through a lightweight, proprietary, and extensible architecture.

III. SYSTEM ARCHITECTURE

ModelLab's architecture comprises three primary layers:

Frontend: Built using React, Vite, and Tailwind CSS, it offers a responsive UI for dataset uploads, progress tracking, and predictions.

Backend (Supabase): Serves as the BaaS layer with PostgreSQL, file storage, authentication, and edge functions.

Edge Functions: Handle model training using TensorFlow.js, CSV/Excel parsing, and real-time progress updates.
A polling mechanism provides updates every 3 seconds, ensuring training feedback remains timely.

Key Features
User Authentication: Secure login/logout using Supabase Auth.

Dataset Upload: Supports CSV and Excel formats with automatic schema detection.

AutoML Training: Enables training of classification, regression, and clustering models.

Model Management: Organize, view, and delete trained models.

Real-Time Progress Monitoring: Displays step-wise progress and metrics during training.

Prediction Interface: Allows users to input new data and receive results with confidence scores.

Model Analytics: Visualizes accuracy, loss, and other key training metrics.

IV. DATABASE DESIGN

Tables include models, profiles, and predictions, implementing Row Level Security (RLS) for user-specific data isolation. JSONB columns allow flexible data storage for training metrics and prediction results.

V. IMPLEMENTATION DETAILS

ML Layer: TensorFlow.js is used to train neural network models for classification and regression, and K-means for clustering.
File Handling: PapaParse and XLSX libraries parse user-uploaded datasets with type detection and error handling.

Security: All endpoints are JWT-protected, and private Supabase storage ensures secure file access.

VI. EVALUATION AND RESULTS

ModelLab was tested with multiple datasets ranging from 1,000 to 50,000 rows. Training times varied depending on the model type and dataset size but remained within acceptable limits for in-browser computation. Real-time updates provided consistent user feedback.

Comparison with Related Work

Feature	ModelLab	Google AutoML	Azure ML	H2O.ai
Open Source	No	No	No	Yes
Real-time UI	Yes	Partial	No	No
Edge Function Support	Yes	No	No	No
Custom UI	Fully Custom	Limited	Limited	Moderate
Cost	Paid	Paid	Paid	Open Core

VII. FUTURE SCOPE

Short-term (1–3 months)

- Model versioning and performance comparison
- Batch predictions and API access
- Export trained models (TensorFlow.js, ONNX, etc.)
- Improved UI for analytics and visualizations

Medium-term (3–6 months)

- Support for new data formats (JSON, Parquet)
- Additional ML algorithms: Random Forest, SVM, etc.
- Advanced hyperparameter tuning
- Monitoring and alerting for production models

Long-term (6+ months)

- Deep learning models and GPU training support
- AutoML pipeline optimization
- Real-time model A/B testing
- Enterprise-ready features (SSO, audit logs)

Conclusion

ModelLab simplifies the ML development pipeline through an integrated, proprietary platform designed for ease of use, flexibility, and real-time interaction. Its scalable architecture makes it suitable for educational, research, and early-stage commercial deployments. With ongoing enhancements in model sophistication, performance tracking, and deployment capabilities, ModelLab is poised to become a leading lightweight AutoML solution.

REFERENCES

1. Google Cloud. (2023). *AutoML documentation*. Retrieved from https://cloud.google.com/automl/docs
2. Microsoft. (2023). *Azure Machine Learning documentation*. Retrieved from https://docs.microsoft.com/azure/machine-learning/
3. H2O.ai. (2023). *H2O AutoML user guide*. Retrieved from https://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html
4. TensorFlow.js. (2023). *TensorFlow.js guide*. Retrieved from https://www.tensorflow.org/js/guide
5. Supabase. (2023). *Supabase documentation*. Retrieved from https://supabase.com/docs
6. PapaParse. (2023). *PapaParse documentation*. Retrieved from https://www.papaparse.com/docs
7. SheetJS. (2023). *SheetJS community edition (XLSX) documentation*. Retrieved from https://github.com/SheetJS/sheetjs
8. React. (2023). *React documentation*. Retrieved from https://reactjs.org/docs/getting-started.html
9. Vite. (2023). *Vite documentation*. Retrieved from https://vitejs.dev/guide/
10. Tailwind CSS. (2023). *Tailwind CSS documentation*. Retrieved from https://tailwindcss.com/docs/installation