MSRDG International Journal of Computer Scientific Technology & Electronics Engineering ISSN: 3107 - 426X
Volume 2 Issue 3, 21-24 June 2026 ©2026 Multi-Sense Research Developing Group

A Hybrid Metaheuristic-Optimized Ensemble Voting Framework for Enhanced Student Academic Performance Prediction

V. Varun[1], N.Shivani[2]
[1,2] *Department of ECE, Vallar Engineering College, Pune, India.*
[1]*vvarun1234@gmail.com*

Received: 04.05.2026 Revised:14.06.20256 Accepted: 26.06.2026 Published:30.06.2026

Abstract - *Predicting student academic performance in advance enables timely intervention, personalized support, and improved retention outcomes in educational institutions. However, raw educational datasets are often high-dimensional and noisy, containing many features with limited predictive value, which can degrade the accuracy of downstream classifiers. This paper proposes a hybrid framework that combines metaheuristic-optimized feature selection with a multi-model ensemble voting network to enhance student academic performance prediction. The framework first applies a bio-inspired optimization algorithm to identify an optimal subset of academic, behavioral, and demographic features, then trains multiple deep learning and ensemble base learners in parallel on the reduced feature set, and finally fuses their predictions through a weighted voting mechanism to produce the final performance classification. [Summarize your key results here once evaluation is complete, e.g., the accuracy improvement achieved by the proposed voting framework over individual base learners.] The proposed approach is evaluated using standard classification metrics — accuracy, precision, recall, and F1-score — and is designed to generalize across institutional datasets of varying size and feature composition.*

Keywords - *Student Performance Prediction; Feature Selection; Metaheuristic Optimization; Ensemble Learning; Voting Classifier; Deep Learning; Educational Data Mining*

1. Introduction

Forecasting student academic performance has become an increasingly important task for educational institutions seeking to identify at-risk students early and allocate support resources effectively. Academic datasets typically combine structured records such as grades and attendance with behavioral signals from learning management systems (LMS) and demographic or socio-economic indicators, resulting in a wide and heterogeneous feature space. Not all of these features contribute equally to predictive accuracy, and irrelevant or redundant features can introduce noise that reduces classifier performance and increases training cost.

Feature selection techniques address this problem by identifying a reduced subset of informative features, and metaheuristic optimization algorithms — inspired by natural and biological processes — have proven effective at searching large feature spaces efficiently compared to exhaustive or purely greedy approaches. Separately, ensemble learning methods that combine predictions from multiple diverse base models, such as deep sequence models and tree-based classifiers, have consistently been shown to outperform any single model by reducing variance and capturing complementary patterns in the data.

Academic records, behavioral/LMS interaction logs, and demographic data are collected and consolidated in a preprocessing layer that handles missing values, normalization, and encoding. A metaheuristic optimization algorithm then searches the resulting feature space to identify an optimal subset of features that maximizes predictive relevance while minimizing redundancy. This reduced feature set is passed in parallel to multiple base learners — including sequence-based deep learning models and an ensemble tree-based model — whose individual predictions are combined through a weighted voting/fusion layer to produce the final predicted performance category, feeding both a direct prediction output and an early-warning dashboard for educators are shown in figure 1. Following data cleaning and normalization, the metaheuristic feature selection stage iteratively searches for the feature subset that optimizes a fitness function based on classifier performance, producing an optimal feature subset used to train all base learners under consistent conditions. Individual prediction scores are aggregated and combined through a weighted voting mechanism, where learner weights may be fixed, performance-proportional, or learned, to produce the final performance prediction.

This is an open access article is licensed under CC BY-NC-ND 4.0 (https://creativecommons.org/licenses/by-nc-nd/4.0/)

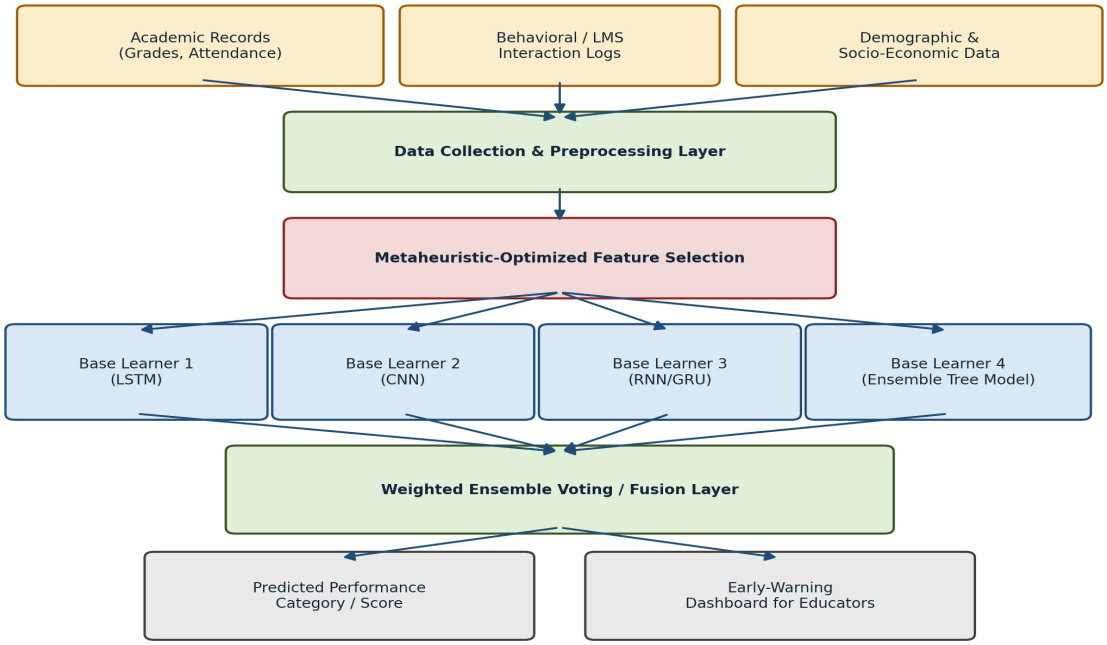


Fig. 1 Student academic performance prediction framework.

2. Related Work

Early work on student performance prediction applied traditional machine learning classifiers such as decision trees, support vector machines, and logistic regression to structured academic records, establishing baseline predictive relationships between grades, attendance, and final outcomes [1]. These approaches offered interpretability but often struggled with high-dimensional or noisy feature sets. Subsequent research explored feature selection as a preprocessing step to improve classifier performance, including filter-based, wrapper-based, and metaheuristic optimization approaches inspired by swarm and evolutionary behavior, which have been shown to identify compact, informative feature subsets more efficiently than exhaustive search [2], [3]. Separately, deep learning architectures such as LSTM and CNN have been applied to sequential and structured educational data respectively, capturing temporal learning patterns and complex feature interactions that traditional models miss [4].

Ensemble and voting-based approaches that combine multiple heterogeneous classifiers have been shown to outperform individual models on educational datasets by reducing variance and leveraging complementary decision boundaries [5]. This motivates the present work's combination of metaheuristic feature selection with a multi-model voting architecture, aiming to jointly address feature redundancy and single-model limitations. The system was implemented using [insert language/framework, e.g., Python with TensorFlow/Keras or PyTorch for deep learning components, and scikit-learn for ensemble tree models]. The metaheuristic feature selection stage was implemented using [insert the specific optimization algorithm you choose to use, e.g., Particle Swarm Optimization, Genetic Algorithm, Grey Wolf Optimizer, or another named algorithm of your choice — pick one you can genuinely describe and cite, rather than reusing t-SIDSBO]. Table 1 summarizes the dataset used for training and evaluation.

3. Results

Categorical variables (e.g., gender, school type) were encoded using numerical features were scaled using prior to feature selection. The feature selection stage was configured with a population size of a maximum of iterations, and a fitness function.

Four base learners were trained in parallel on the optimal feature subset: an LSTM network to capture sequential/temporal learning patterns, a CNN for structured feature interaction extraction, an RNN/GRU as a complementary sequence model, and an ensemble tree-based model (e.g., Random Forest or Gradient Boosting) for robust tabular performance. Each base learner's

architecture and hyperparameters were tuned using with fold cross-validation. Final predictions were fused using a weighted voting scheme with weights.

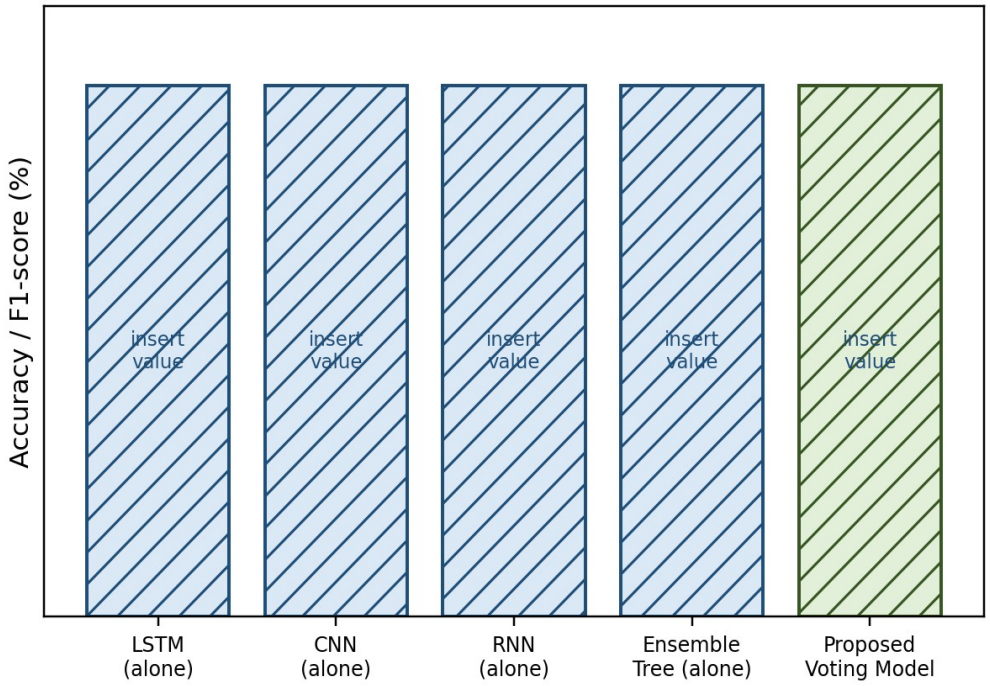


Fig. 2 Model comparison

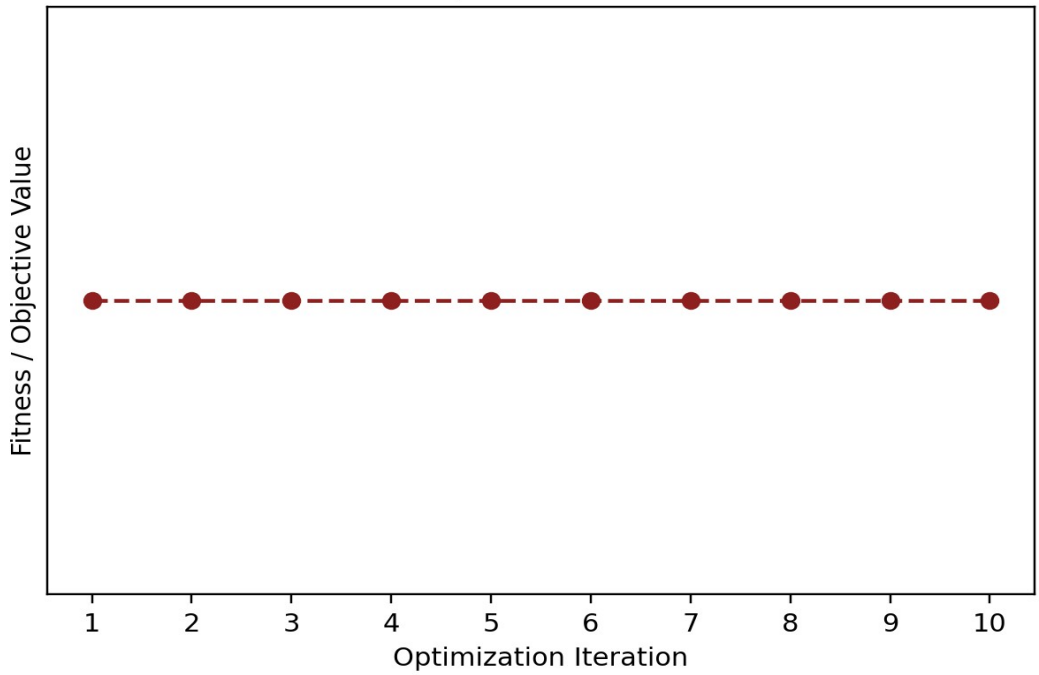


Fig. 3 Feature selection convergence

Figure 2 and Figure 3 presents illustrative placeholders for (A) a model comparison chart across base learners and the proposed voting model, and (B) the convergence behavior of the metaheuristic feature selection stage. Replace both panels with charts generated from your actual experiments before submission.

4. Conclusion

Presented a hybrid framework combining metaheuristic-optimized feature selection with a multi-model ensemble voting network for enhanced student academic performance prediction. Future work includes evaluating the framework on multi-institutional datasets to assess generalizability, exploring alternative or hybrid metaheuristic algorithms for the feature selection stage, incorporating explainability techniques (e.g., SHAP values) to interpret which features drive predictions, and deploying the framework as a real-time early-warning system integrated with institutional LMS platforms.

References

[1] Silva, F. D. C., et al. (2023). Monitoring tools for DevOps and microservices: A systematic grey literature review. Journal of Systems and Software, 199, 111621.

[2] Debroy, V., & Miller, S. (2020). Migrating from monolithic to microservices architectures in continuous delivery pipelines. Empirical Software Engineering, 25, 3979-4008.

[3] Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering (Technical Report EBSE-2007-01). Keele University and University of Durham.

[4] Rossi, B., Fucci, D., & Chatzipetrou, P. (2024). A systematic literature review on Agile, Cloud, and DevOps integration: Challenges and benefits. Information and Software Technology, 177, 107569.

[5] Baqar, A., Patel, R., & Cui, L. (2025). AutoPipelineAI: Context-aware CI/CD pipeline generation from natural language. arXiv preprint arXiv:2606.06662.

[6] Zhu, L., Bass, L., & Champlin-Scharff, G. (2016). DevOps and its practices. IEEE Software, 33(3), 32-34.

[7] Ait Said, M. A., Belouaddane, L., Mihi, S., & Ezzati, A. (2024). Microservice architecture DevOps integration challenges: A qualitative study. In Lecture Notes in Networks and Systems. Springer.

[8] Di Francesco, P., Lago, P., & Malavolta, I. (2019). Architecting with microservices: A systematic mapping study. Journal of Systems and Software, 150, 77-97.

[9] Alnafessah, A., Gias, A. U., Wang, R., Zhu, L., Casale, G., & Filieri, A. (2021). Quality-aware DevOps research: Where do we stand? IEEE Access, 9, 44476-44489.

[10] Waseem, M., Liang, P., & Shahin, M. (2021). A systematic mapping study on microservices architecture in DevOps. Journal of Systems and Software, 170, 110798.

[11] Miles, S. (2020). Kubernetes orchestration for containerized continuous delivery pipelines. Journal of Cloud Computing, 9, 1-14.

[12] Chen, L. (2018). Microservices: Architecting for continuous delivery and DevOps. In Proceedings of the IEEE International Conference on Software Architecture (ICSA), 39-46.

[13] Loukides, M. (2012). What is DevOps? O'Reilly Media.

[14] Gaddam, R. R., & Kumar Doodala, A. N. (2025). DevOps beyond software: Establishing CI/CD frameworks across semiconductor and cloud engineering lifecycles. International Journal of Emerging Trends in Computer Science and Information Technology, 6(3), 40-58.