Poster: A Comparative Analysis of Machine Learning Models for SAT Runtime Prediction

Tomohisa Kawakami[1], Tomoyasu Shimada[2], Xiangbo Kong[3], Hiroyuki Tomiyama[2], Shigeru Yamashita[4]
[1]*Electrical and Computer Engineering, Duke University, USA*
[2]*Graduate School of Science and Engineering, Ritsumeikan University, Japan*
[3]*Department of Intelligent Robot Engineering, Toyama Prefectural University, Japan*
[4]*College of Information Science and Engineering, Ritsumeikan University, Japan*

***Abstract*—The satisfiability (SAT) problem is one of the most fundamental NP-complete problems, and competitions for SAT solvers are held annually to benchmark solver performance. Accurate real-time prediction of solver runtime has become more important for optimal solver selection and efficient resource allocation, especially as solvers and hardware advance. In this work, we evaluate the performance of major machine learning models on SAT runtime prediction tasks, using recent SAT competition datasets. Furthermore, we use not only conventional evaluation metrics e.g., Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), but also more detailed metrics e.g., classification-based confusion matrices and cumulative error distributions. Consequently, we observe that the performance of random forest models, often showing the highest performance in previous studies, is poor and biased toward the median runtime, while the performance of Multilayer Perceptron (MLP) models is more robust even when their RMSE and MAE are higher. These results offer new insights not only for developing more reliable runtime prediction models but also for improving algorithm selection and resource management methods in SAT solving.**

I. Introduction

The propositional satisfiability (SAT) problem, a fundamental NP-complete decision problem in computer science, has attracted a huge amount of research effort. In practice, various kinds of SAT solvers, e.g., CaDiCaL, IsaSAT, Kissat, etc., are developed in annual SAT competitions, introducing new challenging problem instances [1], [2]. Due to the development of solvers and computing processors, the ability to predict accurate runtime in real-time has become more important. Such predictions, for instance, can contribute to real-time optimal solver selection and efficient computing resource allocation in the real world.

Previous studies have explored SAT runtime prediction, linking instance features to runtime [3] and developing prediction models using feature extraction [4], machine learning [5], and graph neural networks [6]. However, these works share a critical limitation that they evaluate the performance using only Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). While RMSE and MAE are widely used in regression tasks, they primarily quantify the average magnitude of errors, i.e., they often fail to reveal the distribution of these errors such as a tendency to consistently under- or overestimate runtime.

In this work, we evaluate the performance of major machine learning models with not only basic evaluation metrics but also more advanced ones such as classification-based confusion matrix and cumulative error distributions.

II. Analysis Methods

For more detailed evaluation, we additionally use classification-based analysis and cumulative error distribution metrics.

In the classification-based analysis, we convert the runtime regression task into a classification problem by categorizing runtime (t) into three classes: Fast ($t < 10$s), Timeout ($t = 5000$s), and Medium (10s $\leq t < 5000$s). The setting of a 10-second threshold for a Fast category aims to distinguish rapidly solvable instances from those requiring more substantial computation. The 5000-second threshold corresponds to the standard timeout commonly used in SAT competitions. Using these categories, we evaluate the model ability to correctly classify instances by constructing a confusion matrix and calculating the F1 score, which allows us to evaluate how well the models classify instances into these practical difficulty categories.

In the cumulative error distribution metrics, we analyze the cumulative distribution function (CDF) of the absolute prediction errors, calculated by $|y_{\text{actual}} - y_{\text{predicted}}|$ for each performance. The CDF plots show the probability that the absolute error is less than or equal to a certain value, calculated by $P(\text{Absolute Error} \leq x)$.

By utilizing these additional metrics, we analyze the performance from multiple perspectives.

III. Experiments

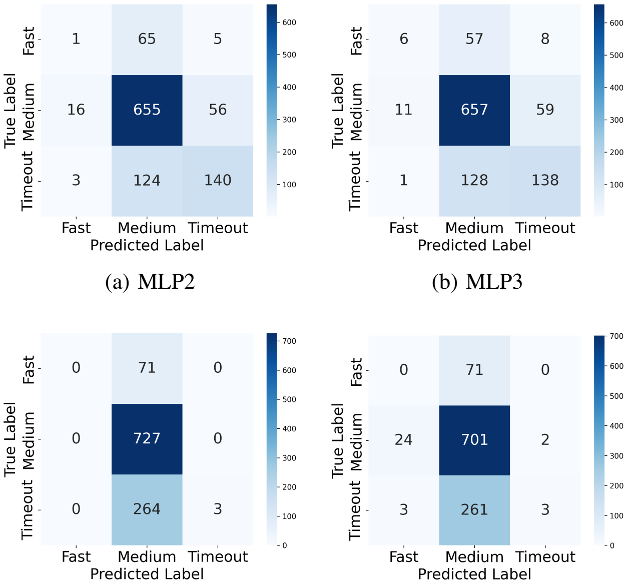
A. Experimental Settings

The datasets for our experiments were sourced from publicly available SAT competition benchmarks [7]. Downloaded dataset consists of 1,065 SAT problem instances; the number of variables per instance ranges from a minimum of 45 to a maximum of 48,505,464 averaging 1,171,465, and the number of clauses per instance varies from a minimum of 264 to a maximum of 214,309,011 averaging 5,665,251. As input data for machine learning models, we utilize 54 features, such as problem size features, graph-based features, balance features, and so on. As label data, we utilize actual runtime of two famous solvers, CaDiCaL_ESA and IsaSAT. we evaluate several models, e.g., MLP with one to three layers (MLP1,

2325-1301/25/$31.00 ©2025 IEEE
DOI 10.1109/RTCSA66114.2025.00035

TABLE I: Comparison of Performance

	CaDiCaL_ESA			IsaSAT		
	RMSE	**MAE**	**F1**	**RMSE**	**MAE**	**F1**
MLP1	1747.48	1204.25	0.430	1828.66	1276.31	0.407
MLP2	1756.68	1091.01	0.485	1891.72	1196.05	**0.451**
MLP3	1808.06	1122.90	**0.519**	1889.64	1194.66	0.435
RF	**1440.42**	**1014.13**	0.286	**1605.68**	**1134.34**	0.270
RR	1922.89	1637.92	0.207	1939.54	1688.76	0.210



(a) MLP2 (b) MLP3

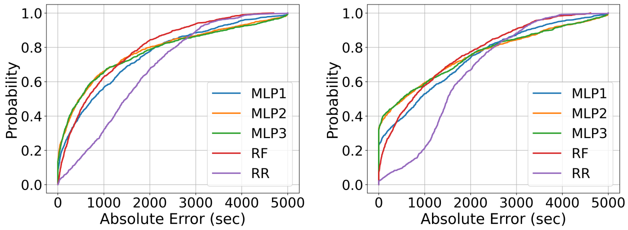
(c) Random Forest (d) Ridge Regression

Fig. 1: Confusion Matrices for CaDiCaL_ESA

MLP2, MLP3), Random Forest (RF), and Ridge Regression (RR) models, with five metrics, e.g., RMSE, MAE, F1 score, confusion matrices, and CDF plots of absolute errors. All models are trained and evaluated using a 10-fold cross-validation method, and hyper-parameters such as learning rate, number of nodes in each hidden layer, depth of trees, etc., are optimized by Optuna library in python, to ensure robust performance estimates and mitigate overfitting. For experiment runs, we use an Intel Core Ultra 7 CPU and 16 GB memory.

B. Experimental Results

We show the key performance metrics, such as RMSE, MAE, and F1 score, in Table I, the confusion matrices in Figure 1, and the CDF plots in Figure 2.



(a) CaDiCaL_ESA (b) IsaSAT

Fig. 2: CDF plots of Absolute Errors

As same as previous studies, RF models achieved the lowest MAE and RMSE for both solvers, shown in Table I. However, F1 scores in Table I show that MLP models, particularly those with more layers, achieved significantly higher F1 scores than RF models, and Figure 1 shows that prediction of RF models is mostly classified into the Medium class, meaning that prediction of RF models is mostly classified into the median runtime. Furthermore, as shown in the CDF plots in Figure 2, MLP models achieved a more favorable error distribution, with a higher probability of small errors and a lower probability of large errors. These results suggest that RF models, in contrast to MLP models, are biased towards predicting median runtime and perform less reliably, despite their low average error.

IV. Conclusion

In this work, we demonstrate that conventional metrics like RMSE and MAE are insufficient for properly evaluating SAT runtime prediction models. Our results show that while Random Forest achieves the lowest RMSE, its performance is misleading due to a strong bias toward predicting median runtime, failing to accurately classify very fast or timeout instances. In contrast, we found that MLP models, despite having a higher RMSE, are more robust and practically reliable, as they can more accurately classify instances into different runtime categories. This shows new insights for real-world applications like online algorithm selection and resource management.

Acknowledgment

This work is supported in part by a grant from JST CREST (JPMJCR24I4).

References

[1] D. Le Berre and L. Simon, "The essentials of the sat 2003 competition," in *International Conference on Theory and Applications of Satisfiability Testing*. Springer, 2003, pp. 452–467.

[2] A. Biere, T. Faller, K. Fazekas, M. Fleury, N. Froleyks, and F. Pollitt, "Cadical, gimsatul, isasat and kissat entering the sat competition 2024," *SAT Competition*, pp. 8–10, 2024.

[3] D. Mitchell, B. Selman, H. Levesque *et al.*, "Hard and easy distributions of sat problems," in *Aaai*, vol. 92, 1992, pp. 459–465.

[4] E. Nudelman, K. Leyton-Brown, H. H. Hoos, A. Devkar, and Y. Shoham, "Understanding random sat: Beyond the clauses-to-variables ratio," in *Principles and Practice of Constraint Programming–CP 2004: 10th International Conference, CP 2004, Toronto, Canada, September 27-October 1, 2004. Proceedings 10*. Springer, 2004, pp. 438–452.

[5] L. Xu, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Satzilla: Portfolio-based algorithm selection for sat," *Journal of artificial intelligence research*, vol. 32, pp. 565–606, 2008.

[6] J. Liu, W. Xiao, H. Cheng, and C. Shi, "Graph neural network based time estimator for sat solver," *International Journal of Machine Learning and Cybernetics*, vol. 16, no. 2, pp. 1145–1156, 2025.

[7] M. Iser and C. Jabs, "Global benchmark database," *arXiv preprint arXiv:2405.10045*, 2024.