SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I12P108

Volume 11 Issue 12, 83-91, December 2024

Original Article

Measuring and Analyzing the Time Complexity of a Prediction Model in Different Scenarios

Rajan Saluja[1], Munishwar Rai[2]

[1]*UIC, Chandigarh University, MMICTBM, MMDU, Ambala, India.*
[2]*MMICTBM, MMDU, Ambala, India*

[1]*Corresponding Author : rajansaluja.mca@gmail.com*

Received: 13 October 2024 Revised: 17 November 2024 Accepted: 07 December 2024 Published: 30 December 2024

Abstract *- These days, almost every industry uses machine learning techniques. These techniques improve the accuracy of predicting the target output by using a wide range and velocity of data. The goal of each method is to quickly and accurately predict the target value. In this research, the execution time, which is the total time taken to predict the student's grades, of the earlier proposed EMLRR model has been calculated. The model is based on an ensemble machine-learning technique: Stacking. Further, we have analyzed the time complexity of the model with other alternatives of stacking by choosing a variety of multiclassification models as meta-models. It has been observed that the proposed model has delivered an accuracy of up to 94% with an execution time of less than 3 seconds. This work uses various platforms, CPUs, and GPUs to analyze the execution time for two different datasets. Various student datasets have been tested to check the model's efficiency in different scenarios. In addition, a comparative study has been done with other possible combinations of base models by increasing and decreasing the number of base models. The proposed prediction model uses the Stacking of four multiclass models to predict student performance with the best accuracy of up to 94% and 89% for two different student datasets.*

Keywords *- Ensemble machine learning, Stacking, Multiclass models, Time complexity, Prediction.*

1. Introduction

In a world where a variety of data sources exist, one of the most significant subfields of Artificial Intelligence is Machine Learning (ML), which is being experimented with in almost every industry. There are a variety of ML techniques whose performance can be evaluated with various parameters like prediction accuracy, Recall, F1 ratio scalability, etc. Extensive research has found that two main categories of machine learning, i.e., ensemble machine learning techniques and deep learning models, are used in every industrial field [1-3]. According to literature, deep learning approaches can solve intricate problems at scale and provide an automated method for extracting features from unstructured data. A variety of successful deep learning techniques are there to solve complex problems. However, the issue with deep learning techniques is that the computational cost of training the model and tuning the parameters is very high. Overfitting is another issue likely to occur while training a deep neural network. Ensemble machine learning is an approach to building a larger, unitary, and powerful model by combining some baseline techniques. Ensemble machine learning models [3] can lower the chance of overfitting, and research proves that ensemble machine learning models are more effective than base models and can be implemented in various industries like disease prediction, fault prediction, and stock price prediction. An ensemble method in the area of education for predicting the student's grades has been used in this work. In this research work, the time complexity of the earlier proposed ensemble machine learning Rajan and Rai Model (EMLRR) [4] has been analyzed to predict a student's grades after first-semester results. The model's accuracy was 94%, and other parameters, such as recall, F1 ratio, and precision, had more than 90% values. The proposed ensemble model is based on stacking, often called a layered generalization, which is a particularly flexible way to combine different models and take advantage of each one's unique capabilities by using a meta-learner. The proposed model stacking of four multiclass classifiers as base models and one multiclass classifier as a meta-model has been experimented with. Base models, nonlinear One vs. Rest (OvR), k-nearest Neighbor classifier (KNN), Decision Tree (DT), Gaussian Naïve Bayes (GNB), and meta-model OvR have been stacked in the EMLRR model. Stacking has several benefits, such as improved prediction accuracy, recall, and F1 ratio. However, it also adds a lot of computing complexity, which may prevent large-scale or real-time applications from using it. To fully understand the parameters influencing the computational demands of stacking ensemble methods, a thorough analysis of their time complexity is the goal of our research work. This research work aims to find out the total execution time of the EMLRR model and the execution time of each model. Parameters that can affect the computational complexity of the stacking-based model have been elaborated. To achieve this goal, various aspects of ensemble machine learning models have been studied from the literature. The time complexity of stacking-based models has been hardly mentioned in the literature. This work aims to minimize the execution time of the EMLRR model for predicting the student's grades. The work consists of six sections. Section two covers the literature analysis of the various approaches of ensemble techniques and their usage. Part three's overview and structure of the EMLRR model have been briefed. The fourth section describes the various factors that can affect the time complexity of stacking-based models. The overall analysis of the time complexity of the model and comparison with other models have been described in the fifth section. In the last section, conclusions have been briefed by answering the research questions set for the work. The research questions that have been set for the work are as follows:

- What is the time complexity in the proposed model?
- What are various factors influencing the execution time of a prediction model?
- What is the effect on execution time by increasing or decreasing the count of base models?
- What is the impact of changing the meta-model on the execution time of a stacking-based model?
- How can the number of output classes affect the execution time in stacking-based models?

2. Literature Survey

In many research works related to predicting students' grades or scores, various ML techniques have been used. Ensemble techniques in the concerned area have also been noted in some literature. Various ensemble models like stacking, boosting, and bagging [2] have been used to predict healthcare, finance, and agriculture. Boosting [2] is a strategy that can weaken the bias and transform a weak model into a strong one. Gaikwad [5] used the bagging technique to combine the predictions from the randomly generated training set and improve the prediction performance of a model. The authors contended that bagging can increase the accuracy because varying the learning set may result in appreciable changes to the predictor that has been produced. Stacking is an approach to train a model to aggregate the prediction of two or more ensemble members. Stacking [6, 7] is a meta-learning technique that discovers the optimal approach to combine each base estimator's predictions to reduce the generalization error in ML models. The idea behind stacked models is straightforward: considering a variety of semantic similarity metrics can lower the possibility that one subpar semantic similarity metric will be unintentionally chosen and implemented.

Ensemble machine learning techniques have demonstrated superior performance in a wide range of applications like fraud detection [8, 10], medical diagnosis [11-13], spam detection [14-16], and sentiment analysis [17-19]. Although the implementation of any ensemble model in the education field is not commendable in literature, we have still found some usage of ensemble models in literature. Satrio Adi Priyambada et al. [20] proposed a model that uses two-way ensemble techniques with three base models to predict student performance with the highest accuracy of 86%. The limitation of the study was that the different kinds of 3 datasets used in that work comprised very few data. There was no idea about the total running time of the model for predicting the student's grades. Martin Stapel et al. [21] tried to classify the students according to their performance with the ensemble technique AdaBoost with a low accuracy of 73.5%. Pooja Kumari et al. [22] invented a model in which voting, bagging, and boosting ensemble techniques were used to identify the weak learners with the highest accuracy of 89%. However, she did not mention the time taken for prediction. Singh and co-authors stated [23] that bagging and boosting ensemble techniques improved the student results, and their model predicted the grades with 89% and 91% accuracy. The limitation of their study was that the dataset was of one class only, and no discussion about the time complexity of their model was available in the work. Meimei Han [24] proposed a model for predicting student learning quality with 91% accuracy with the AdaBoost algorithm. Again, the dataset was very small and constituted only 120 students' entries. A model was proposed by Kingsley Okoye [25] to predict the retention of a student (with an accuracy of 90%) with the help of the bagging method, but they stated that due to multiple k-fold and feature selection, the model had complexity so it could take more time to predict the retention, though there was no focus on the time taken for prediction in the work.

The researchers' primary goal in the literature review was to assess the highest accuracy, but no researcher described the time complexity or satisfied accuracy. The time complexity of basic ML models has been analyzed in the literature. However, the researchers aimed to achieve the highest accuracy and other related parameters using ensemble models. An ensemble model's time complexity is a crucial component compared to other models. The Bayesian ensemble learning model was proposed by Elisabetta Fersini et al. [26] for sentiment analysis, and the authors stated the time complexity of the model as O(n), where n is the number of data samples in the dataset. Afrifa, S [27] observed that the stacking process they used in their proposed model took approximately 3 minutes to predict with 70% accuracy, which was not good. So, we aim to get the best prediction at the fastest time using the EMLRR Model in this work.

3. Proposed Ensemble Machine Learning Model (EMLRR Model)

This section outlines the study structure and describes the Ensemble Machine Learning Rajan & Rai (EMLRR) model proposed earlier in [4]. The model (Figure 1) comprises multiple pivotal steps to enhance performance to predict a student's academic success as soon as possible. To guarantee that the dataset is ready for analysis, the first part of the model has been omitted from the diagram as it involved data preparation, which involves data preprocessing, normalization, transformation, testing at basic model levels, and data imbalance handling process. In the proposed model, multiclassification machine learning models KNN, GNB, DT, and OvR have been stacked, and final testing with the meta-model OvR has been performed.

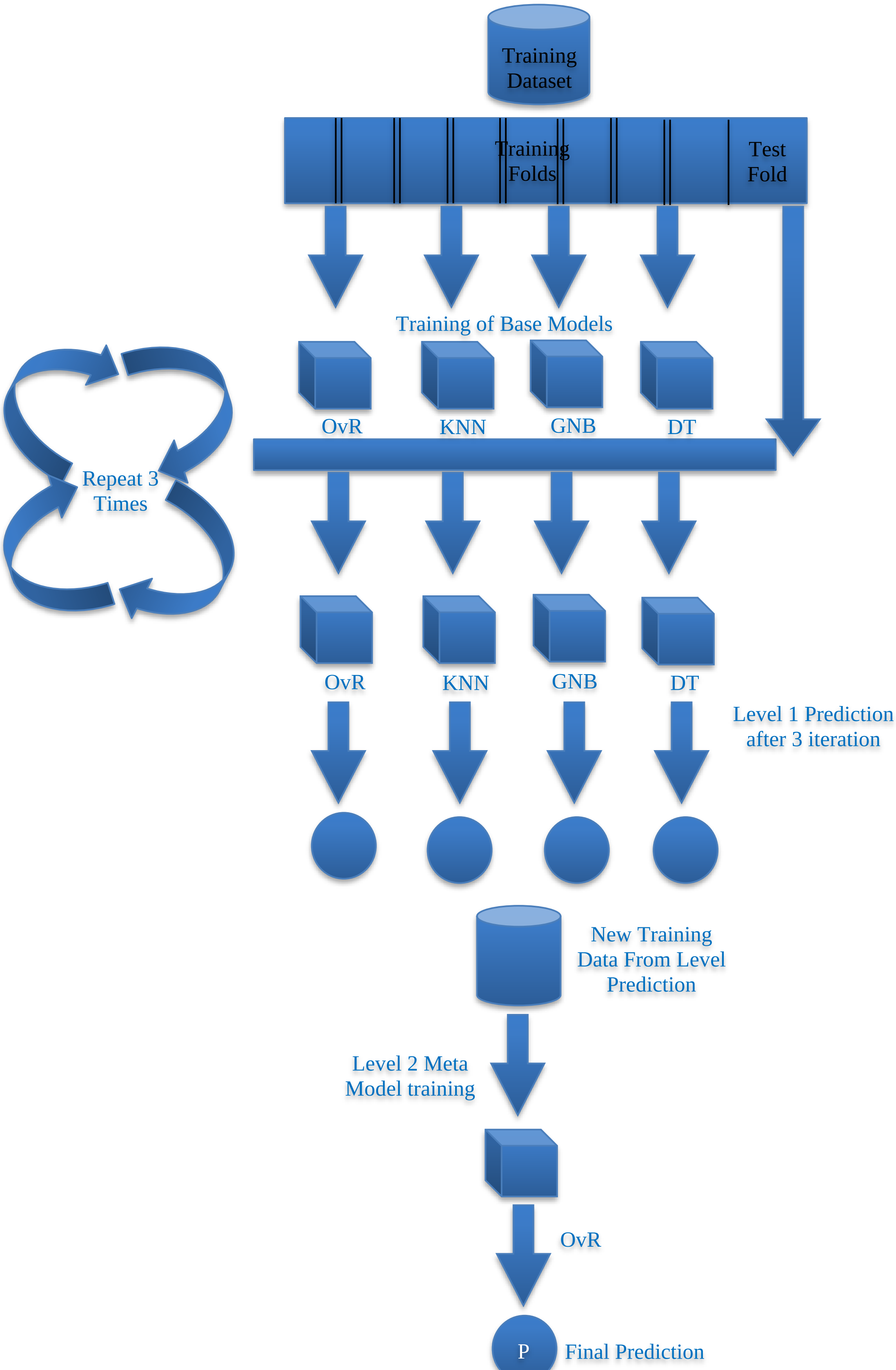

Fig. 1 The proposed prediction model

The structure of EMLRR has been described in Figure 1. The training dataset has been portioned into training and test folds for k-fold cross validation, and individual models have been tested for the first level of predictions in the n-repeat (n=3) process. The predictions collected from base models act as a training dataset for the meta-model. Meta-model is then tested with test data to give final predictions. This work focuses mainly on execution time and analyzing the time complexity of the proposed model. For this, the model has been tested in two different scenarios. The dataset used for the first scenario in the model was taken from an educational institution where the model was implemented to improve the academic grades of engineering students and other graduation courses. The dataset has more than 1600 students' data with multiclass target output attribute Grades with 3 different class grades of multiple courses of an institute (as shown in Figure 2).

Sr. No	Roll No	Batch	Branch	Zodiac	Gender	Cat	City	LOC	Phone	Host	10th	12th%	PQT	1st%	2nd%	3rd%	4th%	5th%	6th%	7th%	8th%	Final%	Avg%	Grade
1	2308259	2009	ECE	pisces	M	GEN	PANIPAT	U	M	N	73	67.6	62	60	64.8	0	0	0	0	64.2	71	0	32.4	0
2	2807001	2007	CSE	sagit	F	GEN	DELHI	U	B	N	82	80	66	57	63.1	63.6	66.8	68.7	70.6	72.5	69.3	67.76	66.5	1
3	2807002	2007	CSE	libra	F	GEN	DELHI	U	B	N	77	62	60	61	61.6	59.5	59.4	62.4	73.1	72.4	71.5	66.55	65.1	1
4	2807003	2007	CSE	capri	M	GEN	SONEPAT	U	B	N	63	69	64	57	63.4	61.4	62.5	69	68.2	72.1	71	67.07	65.6	1
5	2807004	2007	CSE	virgo	M	GEN	PANIPAT	U	M	N	68	71	64	60	67.9	59.4	65.4	69.5	70.6	70.6	69.1	67.55	66.6	1
6	2807005	2007	CSE	aqua	M	GEN	JIND	U	B	N	83	71	60	55	61.4	59.2	61.5	61.7	68.6	66.3	66.6	63.62	62.6	1
7	2807006	2007	CSE	sagit	M	GEN	FARIDABAD	U	B	Y	70	49	62	56	60.9	57	65	65.6	69.9	69.6	68.3	65.44	64.1	1
8	2807007	2007	CSE	cancer	F	GEN	PANIPAT	U	B	N	67	77	65	61	67.1	62.5	65.7	68.6	74.1	73.8	70.2	69.04	67.9	1
9	2807008	2007	CSE	capri	M	GEN	UP	R	M2	Y	58	53	36	57	58.4	52.2	0	57.9	60	57.5	57.5	0	50	0
10	2807009	2007	CSE	tauras	M	GEN	SONEPAT	U	B	N	59	51	58	50	57.1	52.4	61	63.4	66.4	67.8	69.1	62.81	60.9	1
11	2807010	2007	CSE	libra	M	GEN	PANIPAT	R	M	N	61	58	54	49	54.5	50.8	56.8	57.3	57.2	58.4	58.8	56.13	55.3	1
12	2807011	2007	CSE	pisces	F	GEN	KARNAL	U	B	N	62	62	66	66	70.6	67.4	67	67	69.4	70.8	72.1	69.13	68.8	1
13	2807012	2007	CSE	gemini	M	GEN	PANIPAT	U	B	N	67	68	69	65	70.7	67.7	71.1	71.4	74.2	75.2	79.4	72.97	71.9	2
14	2807013	2007	CSE	gemini	M	GEN	KARNAL	R	L	N	57	50	67	58	63.9	63	68.2	71.1	75.8	75.7	79.1	71.35	69.3	1
15	2807014	2007	CSE	cancer	M	GEN	SONEPAT	U	B	N	71	67	60	60	57.9	54.4	63.3	63.7	62	68.8	71.7	64.09	62.7	1
16	2807015	2007	CSE	scorpio	F	GEN	SONEPAT	U	B	N	57	59	59	61	54.3	59.5	61.7	56.8	67.2	66.3	65.5	62.38	61.5	1
17	2807016	2007	CSE	scorpio	F	GEN	DELHI	R	B	N	68	56	74	69	73.1	69.9	76.4	76.5	79.3	74	77.2	75.06	74.4	2
18	2807017	2007	CSE	sagit	F	GEN	SONEPAT	U	B	N	79	71	68	63	68.4	67.8	68.7	68.9	72.5	73.7	76.7	71.01	70	1
19	2807018	2007	CSE	cancer	M	GEN	FATEHABAD	U	B	Y	52	54	60	54	56.8	59.3	59.9	63.4	66.5	69.6	70.3	64.23	62.5	1
20	2807019	2007	CSE	virgo	M	GEN	KARNAL	U	B	N	80	67	76	71	76.4	72	79	79.5	82.3	80.4	84.2	79.18	78.1	2

Fig. 2 Dataset used for Analysis (Scenario 1)

For the second scenario, we have taken the student dataset of 2393 students with 5 class grades from the Kaggle (as shown in Figure 3). All possible combinations of base models and meta-models were evaluated for both scenarios. A variety of Synthetic Oversampling Minority Techniques (SMOTE) [29] options have been applied to handle imbalanced issues in datasets. The model was implemented in Python, and the experiments were run on a Jupyter notebook.

StudentID	Age	Gender	Ethnicity	ParentalEd	StudyTime	Absence	Tutoring	ParentalSu	Extracurric	Sports	Music	Volunteer	GPA	GradeClass
1001	17	1	0	2	19.83372	7	1	2	0	0	1	0	2.9292	2
1002	18	0	0	1	15.40876	0	0	1	0	0	0	0	3.0429	1
1003	15	0	2	3	4.21057	26	0	2	0	0	0	0	0.1126	4
1004	17	1	0	3	10.02883	14	0	3	1	0	0	0	2.0542	3
1005	17	1	0	2	4.672495	17	1	3	0	0	0	0	1.2881	4
1006	18	0	0	1	8.191219	0	0	1	1	0	0	0	3.0842	1
1007	15	0	1	1	15.60168	10	0	3	0	1	0	0	2.7482	2
1008	15	1	1	4	15.4245	22	1	1	1	0	0	0	1.3601	4
1009	17	0	0	0	4.562008	1	0	2	0	1	0	1	2.8968	2
1010	16	1	0	1	18.44447	0	0	3	1	0	0	0	3.5735	0
1011	17	0	0	1	11.85136	11	0	1	0	0	0	0	2.1472	3
1012	17	0	0	1	7.598486	15	0	2	0	0	0	1	1.5596	4
1013	17	0	1	1	10.03871	21	0	3	1	0	0	0	1.5201	4
1014	17	0	1	2	12.10143	21	0	4	0	1	0	0	1.7516	4
1015	18	1	0	1	11.19781	9	1	2	0	0	0	0	2.3968	3
1016	15	0	0	2	9.728101	17	1	0	0	1	0	0	1.3415	4
1017	18	0	3	1	10.09866	14	0	2	1	1	0	0	2.2322	3
1018	18	1	0	0	3.528238	16	1	2	0	0	0	0	1.3844	4
1019	18	0	1	3	16.25466	29	0	2	1	0	0	1	0.4696	4

Fig. 3 Dataset used for analysis (Scenario 2)

The parameters of highest accuracy, average accuracy, precision, Recall, F1, fastest execution time, and average execution times were used to evaluate the performance of our model.

4. Factors Affecting the Time Complexity of a Stacking-Based Model

It was important for this research work to know the components that can impact the time complexity of a stacking model before doing any analysis. The computational complexity of any stacking model can be altered by various components, including factors related to the base models and the meta model. The key factors that can affect the execution time of a stacking model are base model structure, number of base models, meta model complexity, number of entries in the dataset, number of attributes, number of classes in target variable, k value (k-fold validation), K value (the number of neighbors in KNN), and the number of depth level of DT. The number of base models directly impacts the time complexity of the overall model. If the number of base models increases, more predictions will be generated during testing and training, which increases over execution time. The total computational complexity of a stacking model is the combination of the time complexities of all base models and the time taken for other processes. The computational complexity is directly affected by the time complexity of each base model. Complex models like SVM OvR [30] and Random Forest, whose structure is more complex than simpler models like linear regressions or decision trees, take much longer while training, which may increase the overall execution time of a stacking model. Stacking main requirement is that each base model should be trained with a full dataset, which is time-consuming. Furthermore, the training of the meta-model depends on the dataset prepared by the predictions made by base models, which adds extra computational cost, so overall execution time increases. The meta-model's structure also affects a stacking model's overall time complexity. Even if we select a simpler meta-model compared to the structure of base models, its complexity still adds up to the total computation cost. The quantity of test and train data for each base model can also affect the overall execution time of a stacking model. When a dataset is large, and there are many attributes in the dataset, the time complexity of the overall model will increase, especially in the case of OvR and some neural networks where higher computational tasks are required. The model will predict the output quickly if more effort is invested in data pretreatment chores, including cleaning, filling missing entries, feature engineering, and solving imbalance problems in multi-classification output. Once an appropriate dataset is prepared, further processes will take less time. K value, number of test folds, depth of DT, the number of repeats for first-level predictions, and other hyperparameters are tuned for best performance.

5. Results and Discussion

The time complexity of our proposed model on GPUs and a variety of machines available for the work, but for the comparison and analysis, a general machine IdeaPad S340 with Intel(R) Core (TM) i5-8265U CPU @ 1.60GHz with installed RAM 8.00 GB 64-bit operating system, x64-based processor has been used. Dataset 1 comprises more than 1600 students' personal and academic records entries, with the output variable grade for three different grade classes. The second dataset taken from Kaggle contains 2393 students' entries; in the final output grade class, there are 5 different classes. The model's performance was evaluated on various machines, but the final analysis was performed on the prescribed machine. SVM OvR has been used as a meta-model, and 4 base models have been used in the proposed EMLRR model. To calculate the proposed model's time complexity, the base models' time complexity was formulated for both training and prediction and then the meta model's complexity was evaluated. Let the time complexity of individual base models OvR, KNN, DT, and GNB be T1, T2, T3, and T4 for training and P1, P2, P3, and P4 for prediction. Let T5 be the training time complexity of the meta-model OvR and P5 be the time complexity of the meta-model for prediction. Let n be the number of samples, m be the number of classes, K is the number of neighbors taken in KNN, d is the number of attributes in the dataset, and k is the number of folds taken for stacking purposes. Then, different training and prediction complexities can be written as follows:

$$T1 = O(n^3 \cdot m) \quad [1]$$

$$P1 = O(n \cdot m) \quad [2]$$

$$T2 = O(1) \quad [3]$$

$$P2 = O(n.d.K) \quad [4]$$

$$T3 = O(n.\ d.\ \log_n) \quad [5]$$

$$P3 = O(\log_n) \quad [6]$$

$$T4 = O(n.d) \quad [7]$$

$$P4 = O(n.\ d) \quad [8]$$

$$T5 = O(n^3 \cdot m) \quad [9]$$

$$P5 = O(n \cdot m) \quad [10]$$

While doing cross-validation with k-fold methods for stacking purposes, the time complexity of all base learners increases with a factor of k. So, the total time complexity for base learners for training is:

T_{base} = k. (O ($n^3 \cdot m$) + O (1) + O (n. d. $\log_n$) + O ($n.\ d$)) {by adding 1,3,5,7}

$$T_{base} = O(k.(n^3 \cdot m + n.\ d.\ \log_n + n.d)) \quad [11]$$

$$T_{meta} = O(n^3 \cdot m) \quad [12]$$

$$P_{base} = O(n \cdot m + n.d.K + \log_n + n.\ d) \quad [13]$$

$$P_{meta} = O(n \cdot m) \quad [14]$$

T_{total} = O (k. ($n^3 \cdot m$ + n. d. $\log_n$ + $n.\ d$)) + O ($n^3 \cdot m$) {Adding 11 and 12}

P_{total} = O ($n \cdot m + n.d.K + \log_n + n.\ d + n \cdot m$) {Adding 13 and 14}

So, the total time complexity of the proposed model is the sum of the complexities of individual models and stacking process complexity. For analysis of time complexity execution time, i.e., the total runtime of the model (training and prediction) of the proposed model has been calculated on various machines and has been compared with the other three options available for choosing KNN, DT, and GNB as meta-models for the described machine. Various hyperparameter values like the value of k fold for the stacking model, the number of repeats for stacking looping, the random value for the sampling fold, the number of neighbors for the KNN model, the maximum depth of DT model, and various types of the train-test split were tried for the evaluation to get best results. The analysis has been briefed in Table 1.

Table 1. Analysis for dataset 1 (Output grades with 3 different grade classes)

Model	Base Models	Execution Time (s)	Meta Model	Highest Accuracy	Avg Accuracy	Precision	Recall	F1	ET (s)	Avg ET(s)
EMLRR	GNB	0.010	OvR	94%	90%	93%	93%	93%	2.97	3.11
	KNN	0.015								
	SVM	0.083								
	DT	0.007								
M2	GNB	0.010	KNN	91%	88%	90%	89%	91%	3.1	3.15
	KNN	0.016								
	SVM	0.083								
	DT	0.007								
M3	GNB	0.010	GNB	89%	86%	88%	89%	89%	2.8	3.12
	KNN	0.016								
	SVM	0.083								
	DT	0.007								
M4	GNB	0.010	DT	90%	87%	89%	90%	89%	2.87	3.1
	KNN	0.015								
	SVM	0.083								
	DT	0.007								

In dataset I, the highest accuracy for grade class prediction was 94%, and the average accuracy was more than 90%, with the fastest execution time of 2.97s. The Execution Time (ET) of the proposed model was a little bit higher than the other two options when we opted for the meta-model in stacking as GNB or DT. As we have already seen in the SVM OvR training period is higher in comparison with DT and GNB. But collectively, with the accuracy parameter and other parameters (Recall, Precision, and F1 ratio), the proposed model's performance was best with all other alternatives available with stacking of 4 base models, 3 base models, 2 base models, and all other meta model choices. When stacking was done with 3 base models and 2 base models execution time was very low, but accuracy, precision, Recall, and F1 values were not up to the mark.

Table 2. Analysis for dataset 2 (Output grade with 5 different grade classes)

Model	Base Models	Execution Time (s)	Meta Model	Highest Accuracy	Avg Accuracy	Fastest ET (s)	Avg Precision	Avg Recall	Avg F1 Ratio	Avg ET(s)
EMLRR	GNB	0.003	OvR	89%	85%	3.20	89%	88%	89%	3.3
	KNN	0.03								
	SVM	0.05								
	DT	0.003								
M2	GNB	0.003	KNN	87%	83%	3.17	87%	86%	87%	3.2
	KNN	0.03								
	SVM	0.05								
	DT	0.003								
M3	GNB	0.003	GNB	84%	81%	2.84	84%	84%	85%	3.1
	KNN	0.03								
	SVM	0.05								
	DT	0.003								
M4	GNB	0.003	DT	84%	80%	2.92	83%	84%	83%	3.1
	KNN	0.03								
	SVM	0.05								
	DT	0.003								

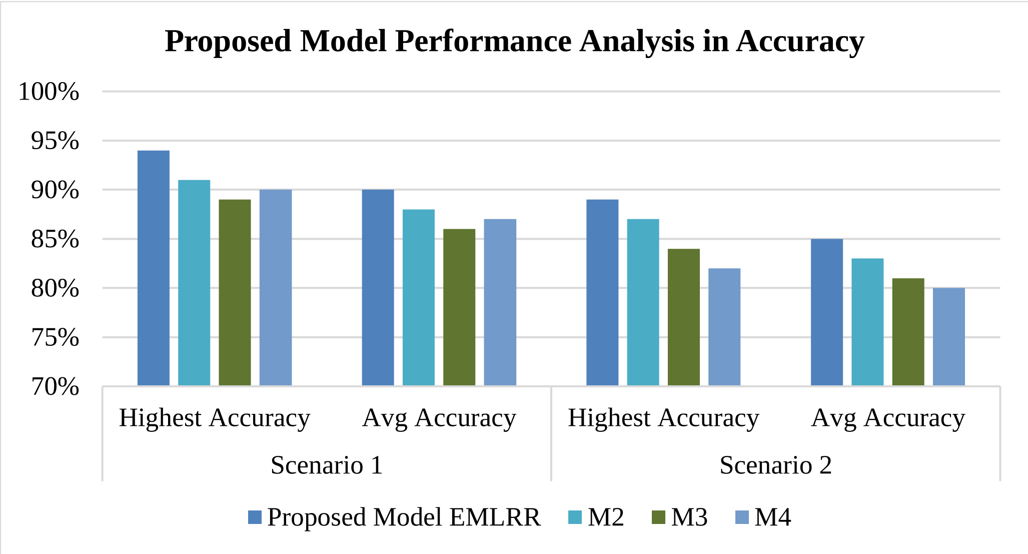

Fig. 4 Performance analysis of the EMLRR Model in terms of accuracy

In the case of scenario 2, when the dataset had 5 different classes in the grade attribute, the highest accuracy for grade class prediction was 89%, and the average accuracy was more than 85%, with the fastest execution time being 3.12s. The time complexity analysis has been described in Table 2 for scenario 2. The results were almost the same in scenario 2; in this case, the number of student entries was greater than in dataset II, so a slight fall in accuracy and execution time was observed. The time complexity of the proposed model is directly affected by the number of training and test samples in the dataset and the number of classes in the target output. So, a decline in the ET from 5-10% was observed in the case of scenario 2, as the amount of data in the second dataset was much higher than in dataset I. Figures 4 and 5 describe a complete analysis of accuracy and time complexity in terms of all four evaluation parameters.

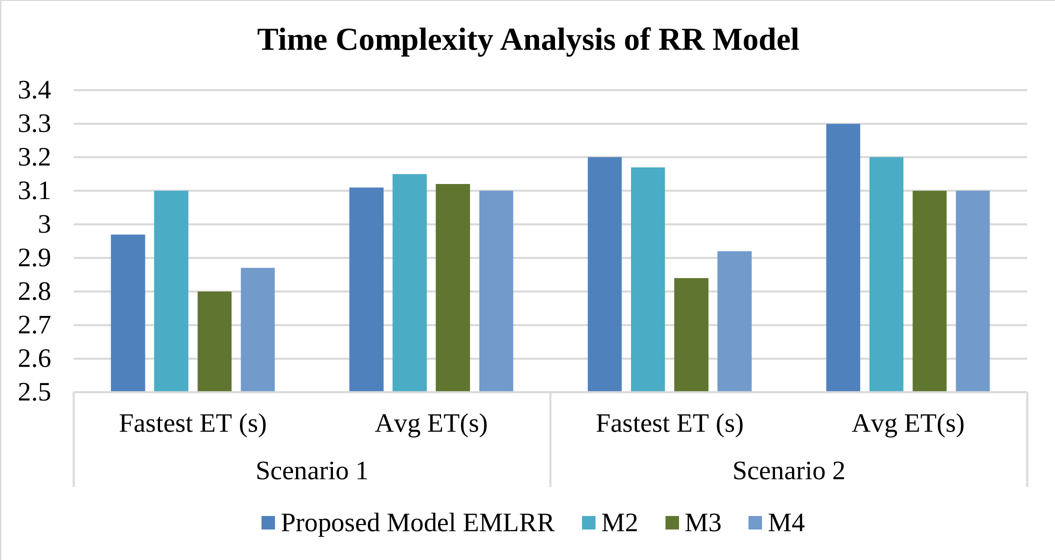

Fig. 5 Time complexity analysis of the EMLRR model

From Figure 4, it can be observed that the proposed model predicts the grade in both scenarios with the highest accuracy of 94% and 89% in comparison with M2 (where KNN is used as meta-model), M3 (where GNB is used as meta-model), and M4 (where DT is used as meta-model). From the time complexity analysis graph (Figure 5), it can be observed that besides the high training complexity of the stacking meta-model, our model predicts the final grade within 3 seconds compared with other stacking models available in the literature.

6. Conclusion and Future Work

In this study, the time complexity of the proposed ensemble machine learning model based on stacking has been analyzed in different scenarios. Finally, a comparison with other possibilities of various combinations of base models and meta-models has been made. Two different datasets in which academic and personal data of students have been collected have been used for the analysis. Experimentally, it has been observed that the proposed EMLRR model predicted a student's academic grade with the highest 94% accuracy and average accuracy above 90% within approximately 3s for the first dataset and 89% highest accuracy in 3.2s for the dataset II. It can be concluded that the EMLRR model predicts a student's grades in the best way with all parameters like accuracy, precision, Recall, F1 values, and total runtime. The model performance has been compared with all other options of increasing and decreasing the number of base models by changing the meta-model. However, the EMLRR model was predicted with the highest accuracy in a faster time. The training period of DT and GNB is less than OvR, so execution time was less when DT and GNB were opted for as meta models. However, with so much gap in accuracy, we can conclude that our model predicts the grade in an average of 3 s execution time. We conclude that the number of base models in stacking has a direct impact on execution time, as when we tried two and three base models stacking, the execution time was less than four base models stacking as the time complexity of a stacking model is the order of the sum of time complexities of base models. The complexity of a stacking model depends on the computational complexity of base models individually and the computational complexity of the meta-model. Even though the meta-model's complexity is simpler, it still adds to the total computational time as we can analyze from Tables 1 and 2 that the total ET is much greater than the sum of ETs of individual base models. The number of train data, test data, and attributes affects the time complexity of an ensemble model. When we used dataset II, the number of data entries was larger than in dataset I, and execution time was slower than in the case of dataset I. Finally, we conclude that the time complexity of our model is a combination of the time complexities of all base models and meta-models and all the time spent in other processes like validation, training, or testing processes.

For future work, researchers can evaluate the model for a variety of student datasets from different institutions and with larger numbers to check the model's scalability and improve the prediction accuracy and execution time of the proposed model. We will evaluate the EMLRR model's performance for predicting student grades with secondary and senior secondary school datasets. We will validate the model for other prediction problems in related areas.

Acknowledgments

I express my sincere thanks to all those who have helped me complete this research work. I sincerely thank the staff and faculty members of MMDU, Mullana, Haryana, India, and Chandigarh University, Punjab, India, who helped me with all the experiments and data collection.

Conflict of interest

I declare that neither I nor any co-author nor any of our relatives or business organizations associated with us are interested in gaining financial benefits from this research work. This research work contributes to the welfare of society and educational institutes.

References

[1] Ammar Mohammed, and Rania Kora, "A Comprehensive Review on Ensemble Deep Learning: Opportunities and Challenges," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 2, pp. 757-774, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Yuzhen Zhang, Jingjing Liu, and Wenjuan Shen, "A Review of Ensemble Learning Algorithms Used in Remote Sensing Applications," *Applied Sciences*, vol. 12, no. 17, pp. 1-20, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] M.A. Ganaie et al., "Ensemble Deep Learning: A Review," *Engineering Applications of Artificial Intelligence*, vol. 115, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] Rajan Saluja, Munishwar Rai, and Rashmi Saluja, "Designing New Student Performance Prediction Model Using Ensemble Machine Learning," *Journal of Autonomous Intelligence*, vol. 6, no. 1, pp. 1-12, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] D.P. Gaikwad, and Ravindra C. Thoo, "Intrusion Detection System Using Bagging Ensemble Method of Machine Learning," *2015 International Conference on Computing Communication Control and Automation*, Pune, India, pp. 291-295, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[6] Jorge Martinez-Gil, "A Comprehensive Review of Stacking Methods for Semantic Similarity Measurement," *Machine Learning with Applications*, vol. 10, pp. 1-12, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Mingshen Lu et al., "A Stacking Ensemble Model of Various Machine Learning Models for Daily Runoff Forecasting," *Water*, vol. 15, no. 7, pp. 1-19, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Gajula Lakshmi Sahithi et al., "Credit Card Fraud Detection Using Ensemble Methods in Machine Learning," *2022 6th International Conference on Trends in Electronics and Informatics*, Tirunelveli, India, pp. 1237-1241, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Abdul Rehman Khalid et al., "Enhancing Credit Card Fraud Detection: An Ensemble Machine Learning Approach," *Big Data and Cognitive Computing*, vol. 8, no. 1, pp. 1-27, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Jinyue Wang, and Chao Yang, "Financial Fraud Detection Based on Ensemble Machine Learning," *IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress*, Falerna, Italy, pp. 1-6, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Shahid Mohammad Ganie, and Majid Bashir Malik, "An Ensemble Machine Learning Approach for Predicting Type-II Diabetes Mellitus Based on Lifestyle Indicators," *Healthcare Analytics*, vol. 2, pp. 1-14, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] Abebech Jenber Belay, Yelkal Mulualem Walle, and Melaku Bitew Haile, "Deep Ensemble Learning and Quantum Machine Learning Approach for Alzheimer's Disease Detection," *Scientific Reports*, vol. 14, pp. 1-10, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Mohammed S. Atoum et al., "A Fog-Enabled Framework for Ensemble Machine Learning-Based Real-Time Heart Patient Diagnosis," *International Journal of Engineering Trends and Technology*, vol. 71, no. 8, pp. 39-47, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Tianfu Li, Jiaxing Shao, and Haoli Wang, "A Research of Spam Message Detection Based on Ensemble Learning," *Proceedings of the 4th International Conference on Artificial Intelligence and Computer Engineering*, Dalian, China, pp. 507-512, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Abdallah Ghourabi, and Manar Alohaly, "Enhancing Spam Message Classification and Detection Using Transformer-Based Embedding and Ensemble Learning," *Sensors*, vol. 23, no. 8, pp. 1-17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Chensu Zhao et al., "A Heterogeneous Ensemble Learning Framework for Spam Detection in Social Networks with Imbalanced Data," *Applied Sciences*, vol. 10, no. 3, pp. 1-18, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Kian Long Tan, Chin Poo Lee, and Kian Ming Lim, "Sentiment Analysis with Ensemble Hybrid Deep Learning Model," *IEEE Access*, vol. 10, pp. 103694-103704, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Rania Kora, and Ammar Mohammed, "An Enhanced Approach for Sentiment Analysis Based on Meta-Ensemble Deep Learning," *Social Network Analysis and Mining*, vol. 13, pp. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] Oscar Araque et al., "Enhancing Deep Learning Sentiment Analysis with Ensemble Techniques in Social Applications," *Expert Systems with Applications*, vol. 77, pp. 236-246, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[20] Satrio Adi Priyambada, Tsuyoshi Usagawa, and E.R. Mahendrawathi, "Two-Layer Ensemble Prediction of Students' Performance Using Learning Behavior and Domain Knowledge," *Computers and Education: Artificial Intelligence*, vol. 5, pp. 1-12, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] Martin Stapel, Zhilin Zheng, and Niels Pinkwart, "An Ensemble Method to Predict Student Performance in an Online Math Learning Environment," *Proceedings of the 9th International Conference on Educational Data Mining*, Raleigh, pp. 231-238, 2016. [Google Scholar] [Publisher Link]

[22] Pooja Kumari, Praphula Kumar Jain, and Rajendra Pamula, "An Efficient Use of Ensemble Methods to Predict Students Academic Performance," *2018 4th International Conference on Recent Advances in Information Technology*, Dhanbad, India, pp. 1-6, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[23] Randhir Singh, and Saurabh Pal, "Machine Learning Algorithms and Ensemble Technique to Improve Prediction of Students Performance," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, pp. 3970-3976, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[24] Meimei Han et al., "Application of Ensemble Algorithm in Students' Performance Prediction," *2017 6th IIAI International Congress on Advanced Applied Informatics*, Hamamatsu, Japan, pp. 735-740, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[25] Kingsley Okoye et al., "Machine Learning Model (RG-DMML) and Ensemble Algorithm for Prediction of Students' Retention and Graduation in Education," *Computers and Education: Artificial Intelligence*, vol. 6, pp. 1-13, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[26] E. Fersini, E. Messina, and F.A. Pozzi, "Sentiment Analysis: Bayesian Ensemble Learning," *Decision Support Systems*, vol. 68, pp. 26-28, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[27] Stephen Afrifa et al., "Ensemble Machine Learning Techniques for Accurate and Efficient Detection of Botnet Attacks in Connected Computers," *Engineer*, vol. 4, no. 1, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[28] Muhammad Fazal Ijaz et al., "Hybrid Prediction Model for Type 2 Diabetes and Hypertension Using DBSCAN-Based Outlier Detection, Synthetic Minority Over Sampling Technique (SMOTE), and Random Forest," *Applied Sciences*, vol. 8, no. 8, pp. 1-22, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[29] Abdiansah Abdiansah, and Retantyo Wardoyo, "Time Complexity Analysis of Support Vector Machines (SVM) in LibSVM," *International Journal of Computer Applications*, vol. 128, no. 3, pp. 28-34, 2015. [CrossRef] [Google Scholar] [Publisher Link]