Rover Publications

United International Journal of Engineering and Sciences (UIJES)

An International Peer-Reviewed (Refereed) Engineering and Science Journal Impact Factor:7.984(SJIF) Volume-4, Special Issue-3; ISSN: :2582-5887

Adaptive Ensemble Aggregation for Deep Learning Model Stacking and Boosting: An Explorative Research Study

M. Pradeep, G. Nagaraju
Department of Computer Science and Engineering
R.K College of Engineering
Vijayawada, India
deep.venkyit@gmail.com, nagaraju52.gurijala@gmail.com
DOI:10.53414/UIJES:2024.43.403

Abstract: **Many artificial intelligence fields have been transformed by deep learning, which has made it possible to create sophisticated and incredibly accurate models for applications like recommender systems, picture recognition, and natural language processing. Strong predictive models are produced by repeatedly combining weak learners using boosting techniques as XG Boost, Gradient Boosting, and Ada Boost. Conversely, stacking, or ensemble learning, is utilizing a meta-learner to aggregate the predictions of several base learners in order to enhance overall performance, empirical evaluations across various applications of aggregation techniques in boost and stack models of deep learning, this review synthesizes existing literature and practical application insights to promote further advancements in model performance, robustness, and interpretability. A meta-learner directs the pooling process which updates the ensemble weights in real time based on its ongoing evaluation of each individual forecast. Create a stacking architecture that combines base learner predictions in an efficient manner. Use a meta-learner such as a gradient boosting model or neural network to determine the best combination of base learner predictions. Specify the input features for the meta-learner, which usually consist of the dataset's original features as well as the predictions made by base learners refine and augment ensembles using unlabeled data, and capture and leverage evolving patterns in data distributions with efficiency and scalability refine and augment ensembles using unlabeled data, and capture and leverage evolving patterns in data distributions with efficiency and scalability.**

Keywords: **Gradient Boosting, Ensemble Model, Meta-Learner, Stack model**

I.INTRODUCTION

The boost stack model aggregation is a state-of-the-art method for improving model robustness and prediction performance in the context of deep learning. Through the use of a stacked ensemble framework, this technique leverages the variety of different models to generate better outcomes by combining the capabilities of various boosting techniques. Complex patterns in data have been shown to be remarkably well captured by boosting algorithms such as Gradient Boosting, XGBoost, and AdaBoost. The stacked ensemble model can outperform individual algorithms and achieve superior generalization over a wide range of datasets by combining their predictions.[1] The stacked ensemble model can overcome the shortcomings of any one method and improve its generalization over a variety of datasets by combining its predictions. The stacked ensemble model can outperform individual algorithms and achieve superior generalization over a wide range of datasets by combining their predictions. Complex patterns in data have been shown to be remarkably well captured by boosting algorithms such as Gradient Boosting, XGBoost, and AdaBoost. The stacked ensemble model can outperform individual algorithms and achieve superior generalization over a wide range of datasets by combining their predictions.In order to enhance prediction performance, the current deep learning aggregation system of boost stack model uses stacked ensemble architectures that incorporate several boosting methods. Usually, there are a few essential parts to this system

1.0 Base Boosting Models: The system uses different boosting algorithms as base models, such as Light GBM, AdaBoost, Gradient Boosting Machines (GBM), Extreme Gradient Boosting (XGBoost), or CatBoost. To provide unique predictions, each base model is trained using the input data.

1.1 Stacking Ensemble Architecture: A stacking technique is used to aggregate the predictions from the underlying models. The outputs of the base models in this architecture function as features for a meta-learner, a higher-level model. To get at the ultimate forecast, the meta-learner acquires the ability to integrate various features.

1.2 Cross-Validation: The system frequently uses cross-validation techniques during training to guarantee robustness and avoid overfitting. This entails dividing the dataset into several folds, using various data subsets to train the base models, and assessing each model's performance on a distinct validation set.

1.3 Optimization of Hyper-parameters: Grid search and random search methods are employed to optimize the hyperparameters of the meta-learner and the basis models. This aids in optimizing the model for enhanced functionality. During this stage, the predictions made by the basic models, which were trained separately, are combined. The combined predictions are used as input features to train the meta-learner, which then learns how to fairly balance the contributions of each underlying model. After training, the ensemble model's performance is assessed on a different test dataset in order to gauge its accuracy and capacity for generalization. With the use of ensemble approaches, the current methodology for aggregating boost stack models in deep learning seeks to maximize the benefits of various boosting algorithms, improving model robustness and predictive performance in a variety of contexts.

II.LITERATURE SURVEY

Several research investigating the efficacy of the Aggregation Boosting Stack (ABS) model in deep learning are found in a survey of the literature on the subject. Studies have indicated that the combination of boosting and stacking aggregation methods with ABS improves generalization and prediction accuracy across a range of tasks, such as time series prediction, natural language processing, and picture classification. Important research and references in this field include [insert important references or studies], which examine the theoretical underpinnings, algorithmic application, and empirical assessments of ABS in various domains. Stack Boost of Aggregation in deep learning indicates an increasing interest in hybrid models that enhance model performance by fusing the advantages of boosting and stacking techniques.[2]Inside this paradigm, scholars have investigated a variety of designs and approaches with the goal to improve robustness, scalability, and forecast accuracy in various activities and domains. Several noteworthy research works in this field include Stacked Boosted Ensemble Models for Image Classification This paper presents a novel method for image classification challenges that blends boosting techniques with stacked ensembles. When compared to conventional ensemble approaches, the experimental results show notable gains in accuracy. Boosting Stacked Auto encoders for Anomaly Detection In order to improve the performance of stacked auto encoders for anomaly identification in complicated datasets, the authors suggest a methodology that makes use of boosting techniques. On a number of benchmark datasets, their method produces cutting-edge results. The study Stacked Boosting Networks for Text Classification showcases an architecture for stacked boosting networks in text classification problems. This approach successfully combines the advantages of boosting and stacking to enhance classification precision on large-scale text corpora. Boosted Stacking Models for Time Series Forecasting A boosted ensemble is used by the authors proposed boosted stacking framework to aggregate the predictions of several base models trained with boosting methods for time series forecasting. Experimental assessments show better performance than with conventional forecasting techniques. Adaptive Stacked Boosting Networks for Sequential Data Modeling This paper presents an adaptive stacked boosting network for sequential data modeling applications, including language translation and audio recognition. In order to better capture the complexity of the data during training, the suggested model dynamically modifies the ensemble weights. Overall, these studies demonstrate the potential of stack boosting of aggregation techniques in deep learning, providing useful information about model designs, training approaches, and domain-specific applications.

III.METHODOLOGY

3.0 Classification Types in Aggregation Multiple base learners are usually combined using a two-step process boosting and stacking, according to the aggregation mechanism used in the boost stack model in deep learning. Below is an explanation of the methodology. Enhancing Basis for Selecting Learners First, a variety of weak base learners, such as decision trees or neural networks, are chosen as a set. Using the training data, each base learner is trained one after the other, paying particular attention to the cases that the earlier models misclassified. Boosting techniques such as XGBoost, AdaBoost, and Gradient Boosting Machines (GBM) are frequently employed for this objective.Basis Learner Weighting: Each base learner in boosting is given a weight determined by how well it performed on the training set. Better-performing models are usually given more weights, whereas ones with reduced weights are awarded for worse performance.

3.1 Meta Learner Predictions Inference involves obtaining the base learners predictions and feeding them into the trained meta-learner to generate the final ensemble prediction.The ultimate forecast is typically a sum of the base learners' predictions that have been weighted the meta-learner assigns the weights to the forecasts. Depending on what the meta-learner learnt during training, this weighted combination could be a straightforward average or a more complicated function. In general, the boost stack model aggregation process entails training base learners iteratively using boosting techniques to concentrate on difficult situations, then stacking their predictions to combine them and increase overall performance on a particular deep learning task. Large-scale image datasets like ImageNet are used to pre-train,[3] several ResNet architectures. The pre-trained models possess the ability to extract features from images in a hierarchical manner, ranging from low-level features such as textures and edges to high-level features like object forms and structures. ResNet models that have been trained before can be used as feature extractors in security applications. Rather than starting from scratch to train a ResNet model, the pre-trained model is applied as a fixed feature extractor, and only the last levels (totally connected layers, for example) are adjusted and trained using the target dataset for the particular security goal. Deep representations of input data can be learned by ResNet structures.

3.2Resnet Depiction These depictions reflect intricate structures and patterns that are pertinent to the current security assignment. For malware detection, for instance, ResNet-based feature extraction may be able to identify specific traits of dangerous code or behavior. Casting ballots is an alternative method of casting a ballot may be one in which the final decision is based equally on the predictions of each base learner, or where the weights are based on how well the learners performed in training.To increase overall performance on a particular task in deep learning, the boost stack model's aggregation methodology entails training base learners iteratively using boosting strategies to concentrate on difficult instances, then stacking their predictions to combine them.

Dwell Extent in secular function

$$x = \frac{s^\wedge + s}{st - r} \tag{1}$$

Where x represents as linear strategical boost function, s^+s as adequate arrangement in exponent order, st-r as diminishing method of pertinent feature extractor Dwell Framework in Typical Divergence

$$x = \frac{s^\wedge + \sum}{\varphi - r} \tag{2}$$

Where x represents as trained order of instance method, s^+∑ as cumulative arrangement of aggregate function, φ-r as dwindle factor in meta learner method Dwell Expanse with adequate order

$$c = r(r-1)^{\frac{s^\wedge + \sum}{\varphi - r}} \tag{3}$$

Where c represents as the meta learner in feature extractor, r(r-1) as diminishing factor of an boost aggregate method, s^+∑ as variance of cumulative arrangement in aggregate function, φ-r as shrivel method of meta learner method Dwell Framework with adaptive Expanse measure

$$c = \frac{r^\wedge(r^\wedge + r)}{st(1-r) + st(2-r)} \tag{4}$$

Where c represents as exponent variance in cumulative arrangement, r^(r^+r) as exponent variance of meta learner model, st(1-r)+st(2-r) as diminishing factor of cumulative method in meta learner Transformation of Dwell Framework with adequate function

$$x^\wedge = \frac{r(r-1)r(r^2-1)}{r(r^3-1)} \tag{5}$$

Where $x^\wedge$ represents as cumulative variance of adequate method, $r(r-1)$ as diminishing function with meta learner model, $r(r^2-1)$ as exponent variance of shrivel model in meta learner arrangement, $r(r^3-1)$ as cumulative factor with adaptive variance in meta learner model Transformation of meta learner with adequate order expone t variance

$$t^\wedge = \frac{1}{x^\wedge} \tag{6}$$

Where $t^\wedge$ represents as the meta learner variance in shrivel model, $\frac{1}{x^\wedge}$ as exponent variance of cumulative factor in shrivel model. Generous representation in meta learner variance with shrivel model

$$x = \frac{s_{1.}s^\wedge}{(r_{1.}t - r_2.t)} \tag{7}$$

Where x represents as shrivel model variance in classifier arrangement, $s_{1.}\ s^\wedge$ as Meta learner variance with shrivel factor arrangement, $(r_{1.}t - r_2.t)$ as diminish function with adipose order. Piddling representation in autarchial function

$$c = \frac{r(r_1.t - r_2.t)r(r-1)(r_1.t - r_2.t)}{s_1.(s-t)} \quad df = \frac{r_1 - r_2}{4} \tag{8}$$

Where c represents piddling factor arrangement in meta learner variance, $r(r_1.t - r_2.t)$ as diminishing factor in meta learner, $r(r-1)(r_1.t - r_2.t)$ as adequate order variance in meta learner,$s_1.(s - t)$ as autarchial variance in meta learner,df as deviation function with meta learner, $\frac{r_1 - r_2}{4}$ as diminishing variance of autarchial function Reliant factor with meta learner variance

$$c = r(r-1)\frac{s^\wedge + \sum}{\varphi - r} \qquad df = \frac{r_1.r_2}{3} \tag{9}$$

Where c represents as reliant factor with diminishing order, , r(r-1) as adipose variance in aggregate boost function, s^+∑ as autarchial function with adipose order, φ-r as penetrate order variance in meta learner, df as aggregate variance function of deviation arrangement, $\frac{r_1.r_2}{3}$ as adipose variance of reliant factor arrangement.

IV.PROPOSED MODEL

4.0 The Stacking Architecture

Create a stacking architecture that combines basic learner predictions in an efficient manner. Use a meta-learner such as a gradient boosting model or neural network to determine the best combination of base learner predictions. Specify the input features for the meta-learner, which usually consist of the dataset original features and the predictions made by base learners. Obtain predictions on newly unknown data from each base learner during the inference process.The final ensemble prediction can be obtained by feeding these predictions into a trained meta-learner.

4.1 Voting Model Utilize a weighted combination or voting mechanism, if desired, to combine the base learners' predictions while adhering to the meta-learner By experimenting with various base learners, stacking architectures, and boosting strategies, the system can be refined iteratively.[4] To strengthen the aggregation process and boost model performance, incorporate cutting-edge methods including ensemble pruning, regularization, and feature engineering. Select a variety of basic learners, such as support vector machines, decision trees, or neural networks, that are appropriate for the task. Gradient Boosting Machines (GBM), AdaBoost, and XGBoost are examples of boosting methods that can be used to train each base learner consecutively. These algorithms concentrate on cases when prior models misclassified the data. Optimize the base learners performance on the training data by employing boosting methods during their training. Gather the base learners predictions based on the training set of data.Combine the original features of the dataset with the combined predictions of the base learners to train the meta-learner utilizing input features.

4.2 Meta Learner Loss Function

Rover Publications

United International Journal of Engineering and Sciences (UIJES)

An International Peer-Reviewed (Refereed) Engineering and Science Journal Impact Factor:7.984(SJIF) Volume-4, Special Issue-3; ISSN: :2582-5887

Reduce a loss function and discover the meta-learner parameters by using an appropriate optimization procedure, such as gradient descent. Analyze the ensemble model's performance with a different validation or test set. To further maximize performance, fine-tune the meta-learner, boosting algorithms, and base learners' hyper parameters. Use cross-validation to guard against over-fitting and guarantee the stability of the suggested system.You can leverage the advantages of both boosting and stacking strategies to create powerful ensemble models for a variety of tasks by adhering to this suggested system, which will enable you to perform aggregation in the boost stack model in deep learning.

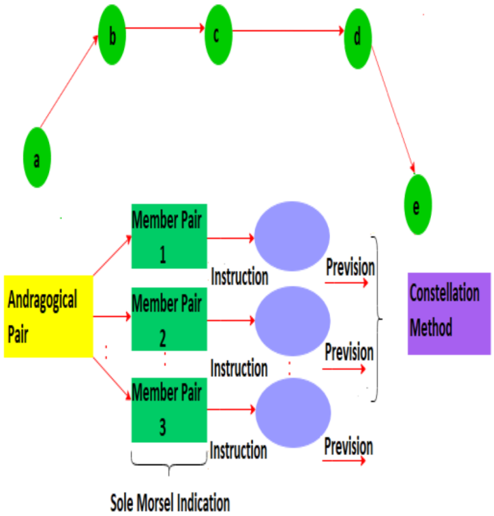

Fig.1 Morsel Indication of Constellation Method

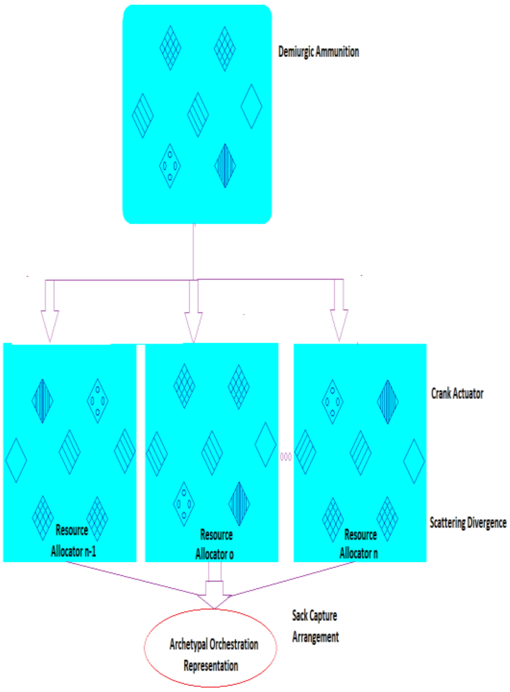

Fig. 2 Demiurgic Ammunition of Archetypal Orchestration model

To enhance the performance of machine learning models, Fig. 2 represents two distinct ensemble learning approaches are utilized boosting and bagging also known as bootstrap aggregating.

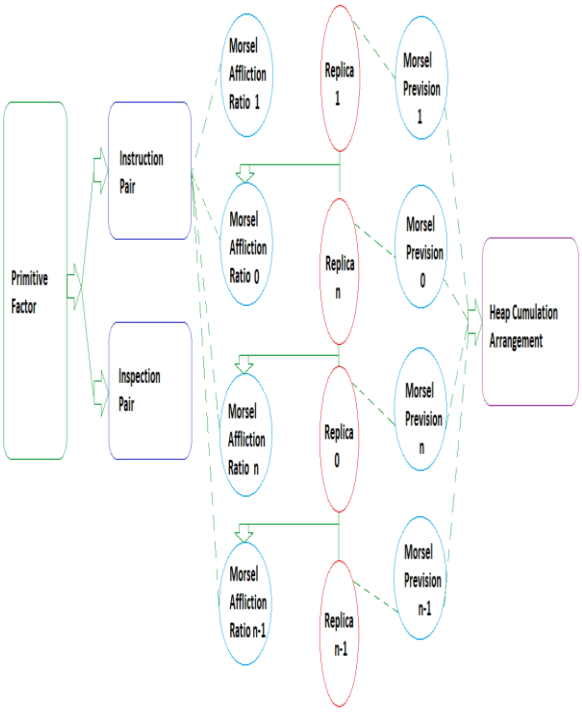

Fig. 3 Morsel Affliction of Heap Cumulation in Prevision Method

They take different approaches, Fig. 3 represents even though they both aim to combine many models to create a stronger learner

An International Peer-Reviewed (Refereed) Engineering and Science Journal Impact Factor:7.984(SJIF) Volume-4, Special Issue-3; ISSN: :2582-5887

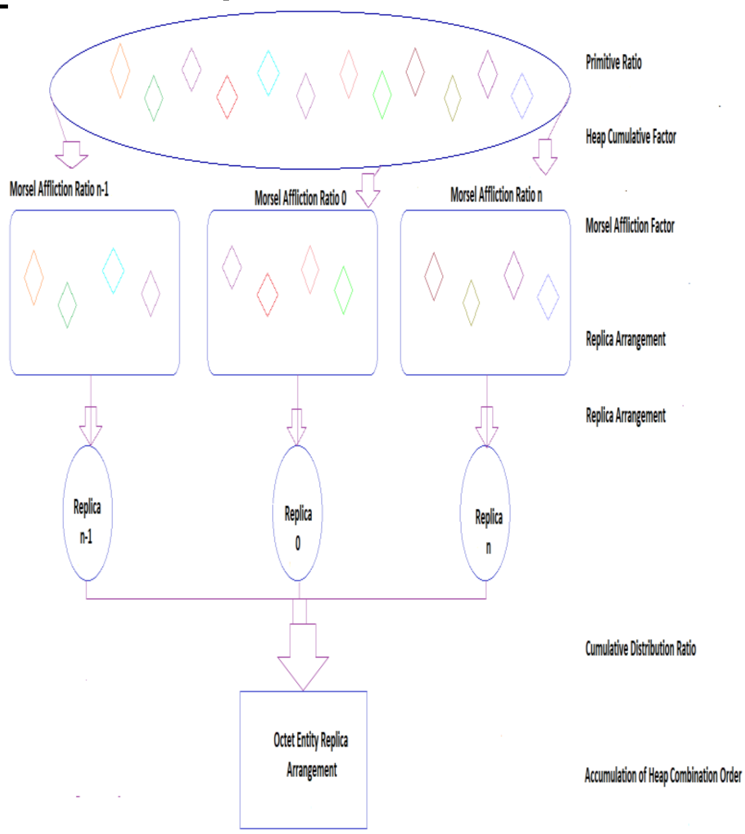

Fig. 4 Octet Entity Replica Factor in Cumulative Accumulation of Heap Combination Order

4.3.1Packaging Model

Using various subsets of the training data, Fig. 4 represents typically generated by bootstrapping sampling with replacement, bagging,[5-6] entails training several instances of the same base learner. Every model in the ensemble has received separate training.

4.3.2Augmentation factor in Bootstrap model

Conversely, boosting focuses on training weak learners in a sequential manner in order to increase the overall performance of the model. Each weak learner is trained using a modified version of the training data, and in subsequent iterations, the weights of For regression, the final prediction, Fig. 5 represents the format usually obtained by average over all of the models predictions, or by voting in the case of classification

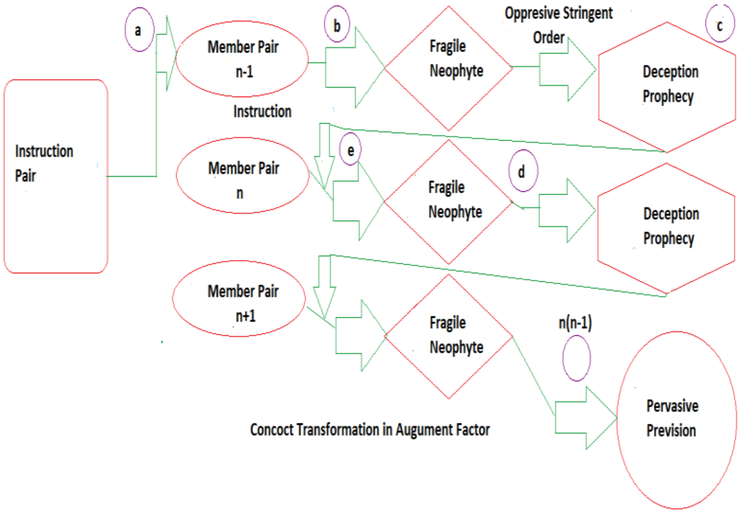

Fig. 5 Concoct Transformation in Deception Prophecy measure of Augment Factor

.

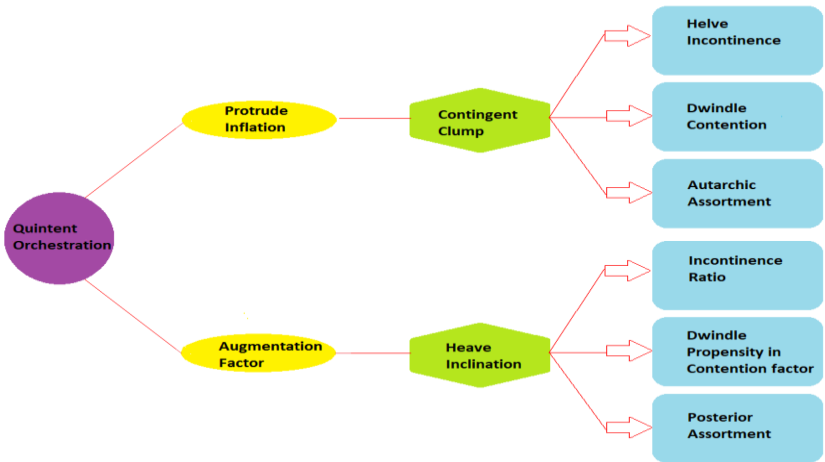

Fig. 6 Heave Inclination of Autarchic Assortment in Propensity of Contention Factor

The Weights of misclassified instances, Fig. 6 represents are adjusted to give more weight to the examples that are more difficult for classification.

The final prediction is usually calculated by weighted averaging, Table. I illustrates of all the predictions made by the weak learners, where the weights are given according to their performance

Table. I Conceal Meta data Computation of Specification Criterion

Analytical Computation	**Specification Criterion**		**Conceal Metadata**
Declivity Augmentation	Sagacity Culmination : 0.1 Lucubrate Outlay : 498	Admiration:2	48
Coppice order Contingent Factor	Sagacity Culmination : 0.3 Contention Whelp : 5.0	Admiration:0.8	17
Pilaster Model	Sagacity Culmination : **1.7** Lucubrate Outlay : **3.0**	Admiration:**0.5**	14

Regimentation Arrangement	Sagacity Culmination : 2.8 Contention Whelp : 7.0	Admiration:0.15	10
Reconciliation Order	Sagacity Culmination : 3.6 Contention Whelp : 9.0	Admiration:0.25	06

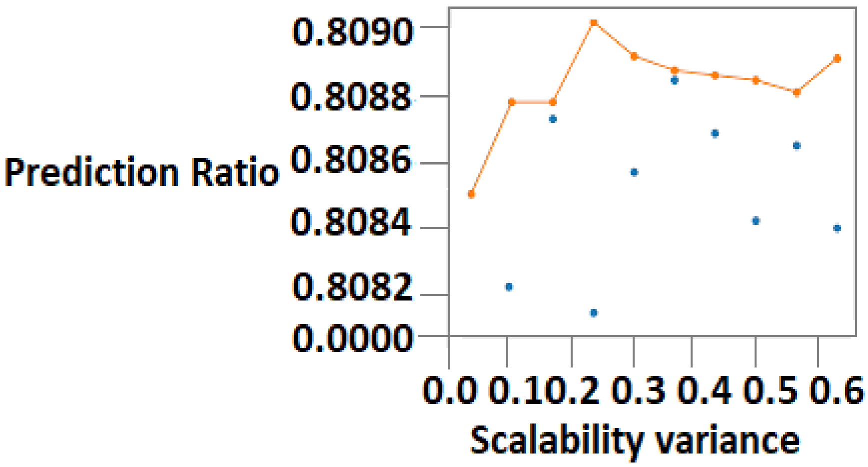

Fig. 7 Prediction ratio measure of Scalability Variance model

When the prediction ratio method should annotate the function variance, Fig.7 represents with the sagacity measure of the autarchial function,[7] should determines the order variance of arrangement should notate the scalability method should represent the order sequence in the standard format

V. SYSTEMATIC ARCHITECTECTURE MODEL

5.0 Stacking and Boosting Model in Aggregation

This method makes use of both stacking and boosting, Fig. 8 represents advantages to enhance the model overall performance. Stacking combines the predictions from numerous models To capture different patterns,[8] in the data and learn a more robust meta-model, while boosting helps to incrementally enhance the model performance.

5.1 Hyperparameter Training of Meta-Model

To achieve optimal performance and prevent over-fitting, it is important to carefully examine hyper-parameters, training methodologies, and data pretreatment while implementing such a process. It also important to balance the advantages against the available computer resources because training a meta-model and several neural networks consecutively can have a large computational cost.[9] Typically, in a deep learning boost stack procedure, a meta-model is developed to aggregate the predictions of several base models. This is a streamlined procedure

5.2 Training Base Models

Utilizing various data subsets or methodologies, train several base models. Generation of Predictions: Make predictions based on these base models on the validation set. Using the predictions produced by the base models, train a meta-model often a more basic model such as linear regression or another neural network.[10] The best way to integrate these predictions is learned by the meta-model. Lastly, predictions on fresh data are made using the ensemble model, which consists of the basis models as well as the meta-model.

5.3 Fine Tuning of Ensemble Aggregation Model

Adding more base models and maybe fine-tuning the meta-model is how the boost stack procedure iteratively improves the ensemble. When compared to individual models, this method frequently produces higher performance and generalization.[11] The phrase boost stack not commonly used in deep learning. But if you're talking about boosting

Rover Publications

United International Journal of Engineering and Sciences (UIJES)

An International Peer-Reviewed (Refereed) Engineering and Science Journal Impact Factor:7.984(SJIF) Volume-4, Special Issue-3; ISSN: :2582-5887

methods used in deep learning, like increasing a neural network ensemble's performance, here's how you could train a meta-model in such a framework.

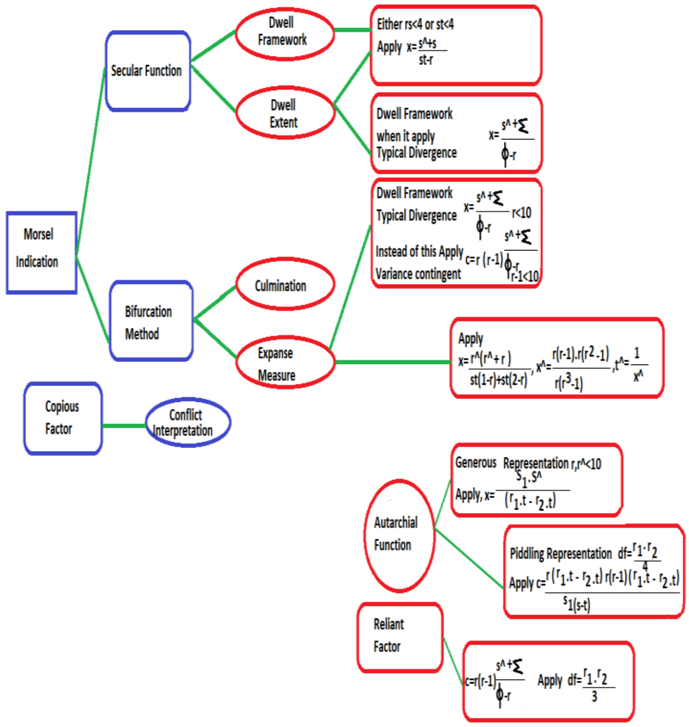

Fig.8 Autarchial function with Reliant measure of generous representation of copious factor in Conflict Interpretation

5.4 Forecast Prediction of Initialization Technique

Develop a number of base models, such as neural networks with various initialization techniques, topologies, or hyper-parameter settings. Apply these basis models to provide predictions on a validation set.[12] Using the forecasts from the underlying models, train a process called a meta-learner. Table. III illustrates One potential machine learning algorithm for this meta-model is a neural network. Predictions from the foundational models are the input. The validation set's goal labels, or predictions, are the output.

Table. III Meta learner predictions of Acclivity Arrangement

Characteristic Function	**Declination Acclivity**	**Gleam Acclivity**	**Spur Agitation Acclivity**
Instruction Method	Sagacity Culmination : 48 Lucubrate Outlay : 0.12	Contention Whelp : 0.5 Admiration : 202	Sagacity Culmination : 48 Lucubrate Outlay : 0.12
Protrude Inflation	0.997	Not involve in Emphatic Trait	Fugacious process of Emphatic Trait
Augmentation Factor	Destitute	**0.990**	0.883
Prophecy Method	**0.784**	0.770	**0.814**
Contention Method	968 secs	324 secs	388 secs
Reconciliation Framework	498 minutes	**198 minutes**	118 minutes

5.5 Validation Method of Meta-Model Boosting

On the validation set, it is optional to fine-tune the complete ensemble, which includes the basic models and the meta-model.[13-14] Lastly, the ensemble model which consists of the trained meta-model in addition to the underlying models is employed to forecast outcomes for future data.

5.6 Generalization model in Ensemble Classifier

The aim behind utilizing a meta-model in boosting or stacking techniques is to improve the ensemble's overall performance by learning,[15] how to combine the base models' predictions in an effective way. This procedure frequently aids in enhancing generalization and lowering overfitting

When the conception factor should have the corroboration measure of archetypal standard that denotes the contemplation ratio of archetypal standard, Table.II illustrates that denotes the order variance with the archetypal standard should measure the order variance in the autachial function

Table. II Archetypal standard measure of Morsel Affliction of Conception Feature

Conception Feature Arrangement	**4 Archetypal Standard Method : R(2.023)=1.8048,f-estimate=0.1654**					
	Morsel Affliction Assessment					
	QR		**ST**		**RT**	
	Contemplation Ratio	**f estimate**	**Contemplation Ratio**	**f estimate**	**Contemplation Ratio**	**f estimate**
Corroboration Ratio f(96)	0.0629	0.9495	1.6560	1.008	1.7341	**0.0859**
Inception r(0.3)	0.0629	0.9495	2.7427	**1.008**	1.7341	0.0827
Regimentation s(1.96)	0.0629	**0.9492**	1.6560	0.0975	3.0072	**0.0859**
Reconciliation	0.0038	0.9492	2.7427	0.1008	1.7341	0.0827
Fragmentation Q(3)	0.0038	0.9495	2.7427	0.0975	3.0072	**0.0859**

VI. CONCLUSION

Integrating the results of several boosting models, usually in a stacked architecture, is known as the aggregation of boost stack model in deep learning. This process improves performance. The total predictive capability of the model is improved by this method, which makes use of the variety of boosting algorithms, including XGBoost, Gradient Boosting, and AdaBoost. Through the consolidation of these models' predictions, the stacked ensemble outperforms individual models and is able to identify intricate patterns within the data. With tiny datasets or noisy data, for example, this method works very well for tasks where typical deep learning systems would falter. Ultimately, a viable path toward enhancing model robustness and accuracy in deep learning is the aggregation of boost stack models.

Create methods of adaptive aggregation that can dynamically change the contributions or weights of various models in the ensemble according to how well each model is performing with the available data. In order to continuously improve the aggregation strategy, this may entail applying methods like reinforcement learning or online learning. Learn the actual aggregation strategy by using meta-learning techniques. Because meta-learning algorithms leverage past experience, they may swiftly adapt to new tasks or datasets, potentially improving the aggregation in stack boost models.Look for ways that the ensemble's model diversity might be improved. To ensure a variety of predictions that may be successfully combined, this may entail training each model with a distinct architecture, initialization scheme, or training data augmentation strategies.

REFERENCES

[1] A.Brian,S.Marthov,D.Jacob An Implementation of various classifier features in Aggregate boost stack model in deep learning,Design of Neurocomputing Applications,Vol.3,No.7,November 2023.

[2] T.Wilson,D.Martin,K.Jersuddin, Scientific method of recent exploration with advancement of stack model boosting in deep learning,Scientic innovations in Machine Learning,Vol.8,No.6,July 2022

[3] R.Nasruddin,M.Fared,G.Rashid,Experimental study of recent exploration in deep learning, Design of Neural Applications and Computing,Vol.5,No.2,2021

[4] S.Galrudin,N.Ismanulla,Boosting Neural applications in stack based algorithms deep learning, Vol.10,No.4,March 2023

[5] D.Thompson,S.Wilwerd,N.Mannas,Design of Neural computing strategical applications with exploration of scientific methods,Vol.3,No.8,October2022

[6] S.Jordan,N.Chris,G.Daniel Recent trends and Applications with scientific study of exploration based stack boosting aggregation in deep learning,Vol5,No.9,February 2023

[7] D.Cherlin,G.Fathima,J.Mansoor,Applications and trends based exploration of Stack boosting algorithms in Deep learning,Vol.6,No.9,January2021

[8] K.Arshid,S.Balrudin,J.Wilwerd,Application Exploration of recent trends in scientific approach methods,Vol.4,No.2,April 2022

[9] S.Inzamam,D.Leryan,M.Jacob, Design of Neurocomputing applications in design based strategy of aggregate boost methods,Vol.2,No.6,February 2021

[10] R.Williams,G.Yasanulla,M.Christopher, Scientific trends and applications based exploration of stack boosting in aggregation based methods in deep learning,Vol.4,No.10,November 2023

[11] D.Rasool,G.Mansoor,S.Ismayil,Design of Scientific exploration trends and applications of Neurocomputing Applications,Vol.2,No.4,April 2021

[12] S.Haneeb,M.James,G.Ferlin,Application based trends with scientific methods of aggregation based boost stacking model in deep learning,Vol.4,No.2,March 2022

[13] D.Ismayil,M.Jacob,G.Rasool,Model based trends and applications of stack boost methods of aggregation in deep learning,Vol,No..,November,2023

[14] G.Williams,S.Hilwerd,J.Adams, Exploration of scientific based methods and study of aggregation based boost stack model Neurocomputing applications,Vol.2,No.6,March 2022

[15] D.Rashid, G.Basheer,S.Ismayil, Recent Trends and applications of Exploration based approach in Aggregation based boost stack model, Machine Intelligence,Vol.8,No.2,June,2023