International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Performance Evaluation of Proposed Deep Ensemble Method Algorithms in Distributed and Traditional Computing Environments for Structured Data Analysis

M. Bhargavi Krishna[1], Prof. S. Jyothi[2], Dr. P. Bhargavi[3]

Submitted:06/02/2024 **Revised**: 14/03/2024 **Accepted**: 21/03/2024

Abstract: Storing big data is a complex because of the large amount, many types, and high speed at which the data is generated. This data may be imbalanced distributed data because it contains structured and unstructured data. To classify these data, it requires multiple technologies and methodologies to ensure efficient processing and retrieval of the data like Map Reduce model which is introduced in big data analysis. MapReduce is a parallel processing technique used to process data in a distributed manner. Specifically, it simplifies concurrent processing by partitioning large amounts of data into smaller segments and executing them simultaneously on a big data platform. However, it lacks efficiency compared to other forms of parallel processing and may exhibit slowness when performing specific operations. Additionally, it is not optimised for real-time processing, also it is unsuitable for applications that demand minimal latency. In the present era, distributed computing has emerged as a viable and increasingly popular choice for numerous applications of complex data. This is mostly owing to the technological advancements in computers, networks, mobile devices, and wireless communication technologies, which are becoming widely utilised in day to day lives. In this paper, proposed a Deep Ensemble method such as Deep Learning without Tuning, Deep Ensemble with boosting and Performance Tuning are applied to classifies the structured data in both distributed computing and traditional computing environments.

Keywords: *Distributed Computing, Bigdata, Map Reduce, Deep Ensemble algorithm, Traditional Computing.*

1. Introduction

Big data analysis has become vital to modern business practices, allowing organizations to gain valuable insights
and make informed data-driven decisions. To handle the massive volume, diverse types, and high speed of big data, distributed computing has become a popular method, offering better scalability and performance than traditional computing systems.

However, distributed computing for big data analytics also presents several challenges, such as data security and
privacy, data heterogeneity and integration, scalability and performance, fault tolerance and availability, and resource management and allocation. These challenges must be addressed to fully realize the benefits of distributed computing for big data analytics. A Hadoop-based platform is an example of a distributed computing system that is well-suited to dealing with large amounts of data. Scalability and fault tolerance are two further benefits of this platform, which can easily manage all three types of data. As a result, Hadoop-based platforms based on distributed scale-out storage systems have become well-known for dealing with large amounts of data [1].

This research paper explores the challenges and opportunities of distributed computing for big data analytics.

The paper will provide The analysis of big data has become crucial for contemporary company operations, enabling organisations to get useful insights and make well-informed decisions based on data. In order to manage the immense quantity, varied formats, and rapid velocity of big data. Distributed Computing has gained popularity as it provides superior scalability and performance compared to Traditional Computing Systems. Nevertheless, the utilisation of Distributed Computing for Big Data Analytics entails certain obstacles including concerns over data security and privacy, integration of diverse data types, the ability to scale and performing of big data efficiently, the

1Research Scholar, Dept. of. CSE, SOET, Sri Padmavathi Mahila Visvavidyalayam, Tirupati
2 Professor, Dept. of. Computer Science, Sri Padmavathi Mahila Visvavidyalayam, Tirupati
3Assistant Professor, Dept. of. Computer Science, Sri Padmavathi Mahila Visvavidyalayam, Tirupati

need for fault tolerance and availability, and the administration and allocation of resources. In order to fully reap the advantages of distributed computing for big data analytics, it is imperative to tackle these issues. A Hadoop-based platform exemplifies a distributed computing system that is highly suitable for handling vast quantities of data. Additionally, this platform offers scalability and fault tolerance, allowing for efficient management of all three forms of structured data. Consequently, Hadoop-based platforms that rely on distributed scale-out storage systems have gained recognition for their ability to handle vast quantities of data [1].

This paper explores the Distributed Computing for processing of Structured Data with proposed Deep Ensemble without Tuning, Deep Ensemble with boosting and Performance Tuning algorithms in both Distributed and Traditional Environments. Because in deep learning Ensemble technique functions much like seeking advice from multiple sources before making a significant decision such as purchasing a car. Just as you wouldn't rely solely on one opinion, ensemble models combine predictions from multiple base models to enhance overall performance [2]. Then the proposed algorithms are compared with their time and accuracy in both Distributed Computing and Traditional Computing Environments for processing of larger datasets.

2. Methodology

In this paper the proposed algorithms like Deep Ensemble without Tuning, Deep Ensemble with boosting and Performance Tuning are applied on structured datasets to classifies the best environment by comparing run time of algorithm in distributed computing and traditional environments for larger data. The step by step process of work is shown in figure 1.

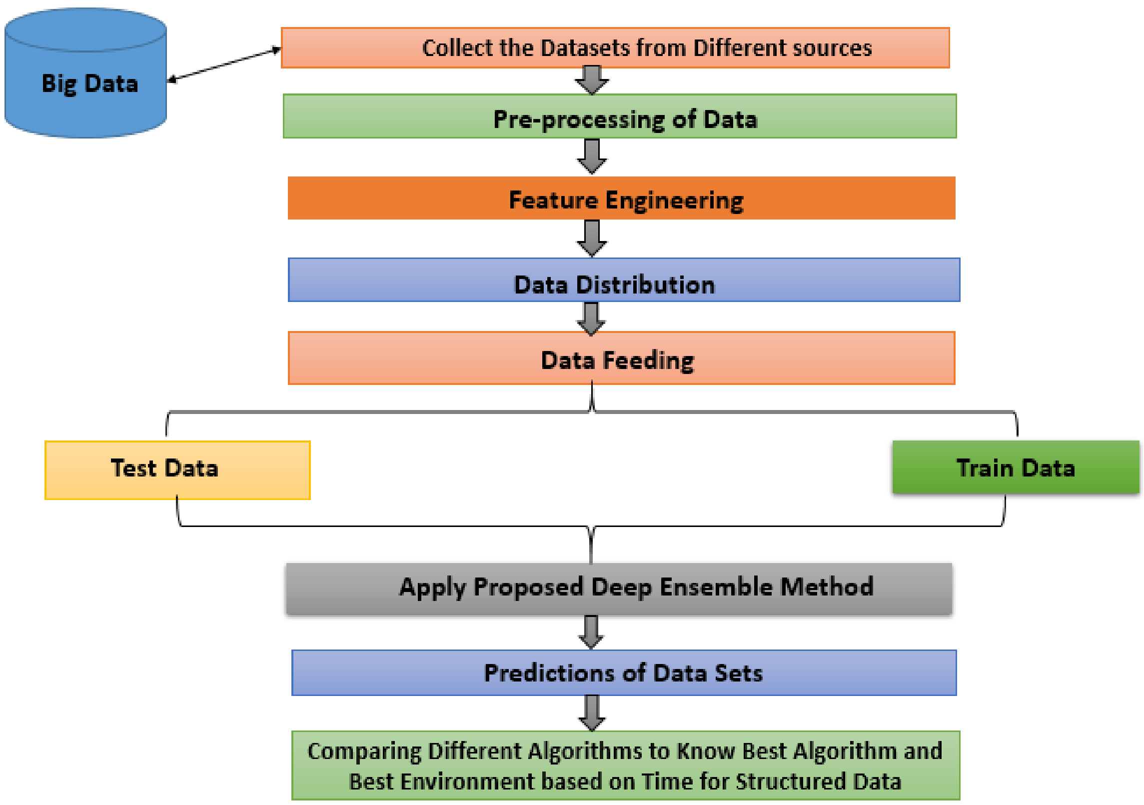

Figure 1: Workflow

2.1. About Datasets

2.1.1. Data Set 1: LOGIN DATA SET FOR RISK BASE AUTHENTICATION

A total of one Terabyte of data is extracted from a globally accessible web service, which has had more than 33 million login attempts and has a user base of 3.3 million individuals. Data were collected from February 2020 to February 2021. The objective of these data sets is to expedite research and development for Risk-Based Authentication systems. The data was obtained from the actual login patterns of more than 3.3 million people on a global scale using a large-scale single sign-on platform.

Data Set 2: INTERNET TRAFFIC MANAGEMENT SYSTEM

Systems gather up-to-the-minute traffic statistics and implement the required measures to alleviate internet traffic congestion. The Unified Traffic Management system captures the scores of protocols such as IP, UDP, TCP, HTTP, FTP, DNS, and TFTP. The highest score is awarded for effective maintenance.

2.1.2. Data set 3: MEDICAL RECOMMENDATION SYSTEM

A medical recommendation system utilises patient feedback to propose specialists specialising in a specific disease. In the rapidly advancing technology world, it is crucial since it has the potential to save numerous lives. Patients will rate doctors depending on their performance.

2.2. Deep Ensemble Method

Ensemble learning is a technique that combines the mapping functions acquired by multiple classifiers to create a consolidated mapping function. An ensemble learning system often involves the use of an aggregation function G to combine a group of baseline classifiers, c1; c2; ... ; ch, in order to make a single prediction. The output prediction based on this ensemble approach may be determined using the following equation for a given dataset of size n and features of dimension

$$m, D = \{(x_i, y_i)\}, 1 \leqslant i \leqslant n, x_i \in R^m:$$

$$y_i = \emptyset(x_i) = G(c1, c2, \dots, c_k)$$

Various approaches have been suggested over time, employing different techniques to calculate this combination. These approaches have developed diverse ensemble methodologies, leading to improved generalisation of the learning models. The ensemble strategies can be broadly classified into the categories like bagging, boosting, stacking, mixture of experts [3,4].

2.2.1. Bagging

The Bagging ensemble methodology, shortly "Bootstrap Aggregating," is one of the earliest proposed ensemble methods. In this approach, subsets are generated from a dataset by a process known as "bootstrap sampling." In essence, the process involves generating random subsets of a dataset through replacement, allowing for the possibility of a data point being included in multiple subsets.

These subsets are now considered as separate datasets, on which several Machine Learning models will be trained. The predictions made by all models trained on different subsets of the same data are taken into consideration during the test phase [5]. An aggregation process is employed to calculate the ultimate prediction. The primary objective of the bagging approach is to mitigate variance in the ensemble forecasts. The bagging ensemble mechanism is shown in the figure 2 where parallel processing takes place.

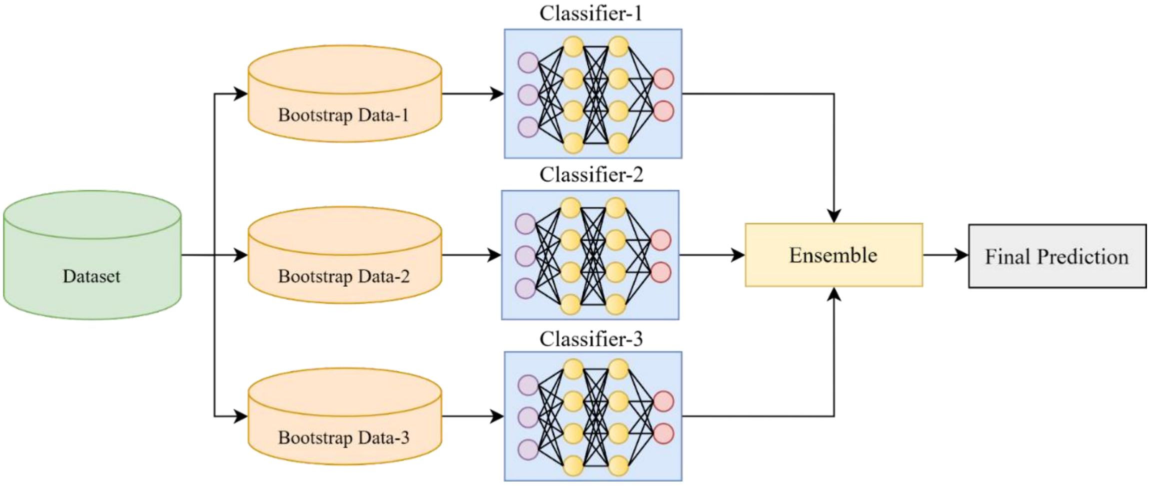

Figure 2: Exemplifies the Bagging Ensemble Mechanism

2.2.2. Boosting

The boosting ensemble mechanism operates in a significantly distinct manner compared to the bagging mechanism. Here, the dataset is processed sequentially instead of parallel processing. The

initial classifier is trained using the complete dataset, and the resulting predictions are examined.
where misclassifications made by Classifier-1 occur when samples are located close to the decision border of the feature space that are inputs to the second classifier. This is done to enable Classifier-2 to explicitly concentrate on the troublesome regions of the feature space and acquire a suitable decision boundary. Similarly, subsequent iterations of the same concept are utilised for classifier 3 and the collective set of all these preceding classifiers is calculated to generate the ultimate prediction for the test data as shown in the figure 3.

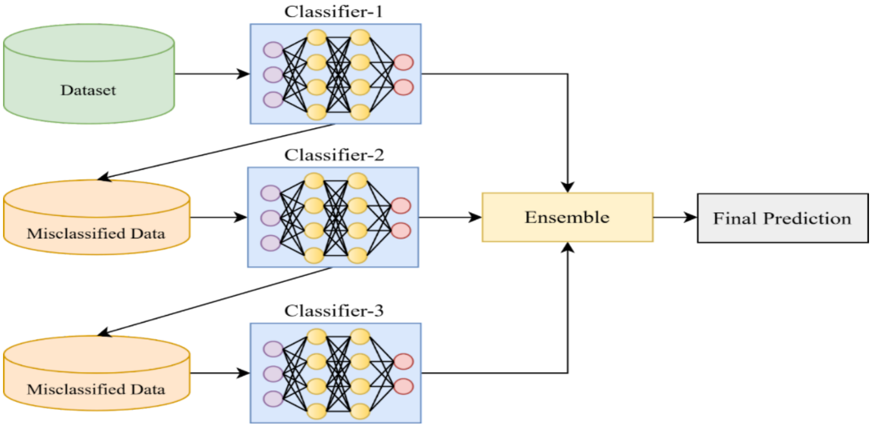

Figure 3: The Mechanism of Boosting Ensemble

The primary objective of the boosting method is to mitigate bias in the ensemble judgement [6]. Therefore, the classifiers selected for the ensemble typically require low variance and high bias, meaning they should be simpler models with fewer trainable parameters.

2.2.3. Stacking

The stacking ensemble method also entails the creation of bootstrapped data subsets, similar to the bagging ensemble mechanism used for training multiple models. However, in this case, the results of all these models are utilised as an input for another classifier, known as a meta-classifier, which ultimately makes predictions for the samples. The rationale for employing two layers of classifiers is to ascertain the adequacy of learning the training data. For instance, in the case of the cat/dog/wolf classifier if Classifier-1 is capable of differentiating between cats and dogs but struggles to distinguish between dogs and wolves, the meta-classifier in the second layer will be able to detect and learn this behaviour from Classifier-1[7,8]. Subsequently, the meta classifier can rectify this behaviour prior to generating the ultimate prediction. Figure 4 displays a visual depiction of the stacking mechanism.

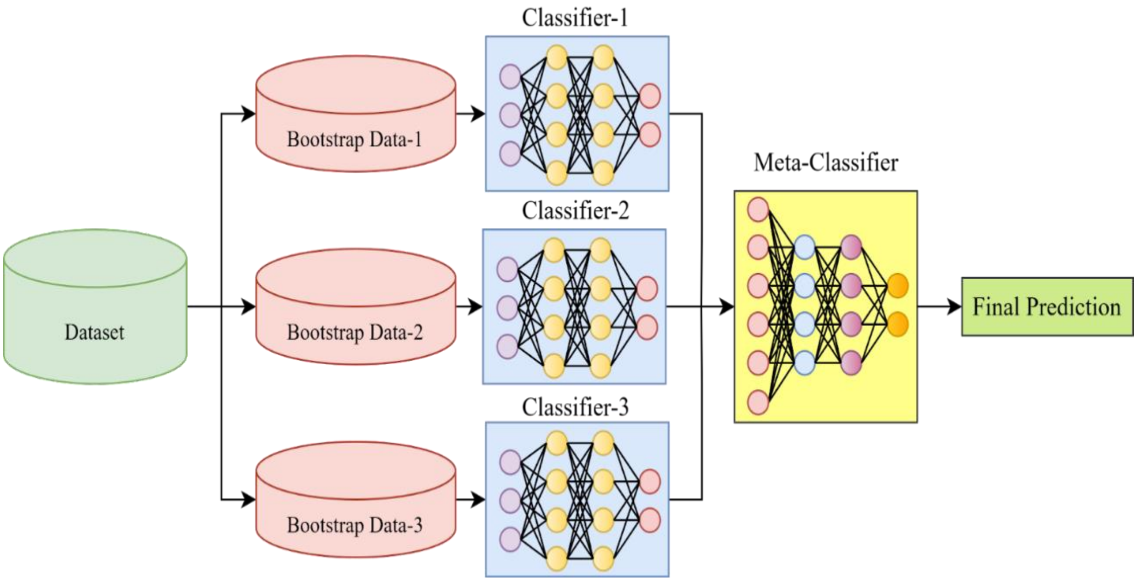

Figure 4: The Stacking Ensemble Mechanism

The figure 4 illustrates a single level of stacking. Furthermore, there exist multi-level stacking ensemble techniques that involve the incorporation of extra layers of classifiers in between bootstrap datasets [9]. Nevertheless, these methods incur significant computational costs for just a marginal improvement in performance.

2.2.3. Mixture of Experts

The "Mixture of Experts" ensemble genre involves training many classifiers and combining their outputs using a generalised linear algorithm. The weights assigned to these combinations are decided using a "Gating Network," which is a trainable model often implemented as a neural network, as depicted in figure 5.

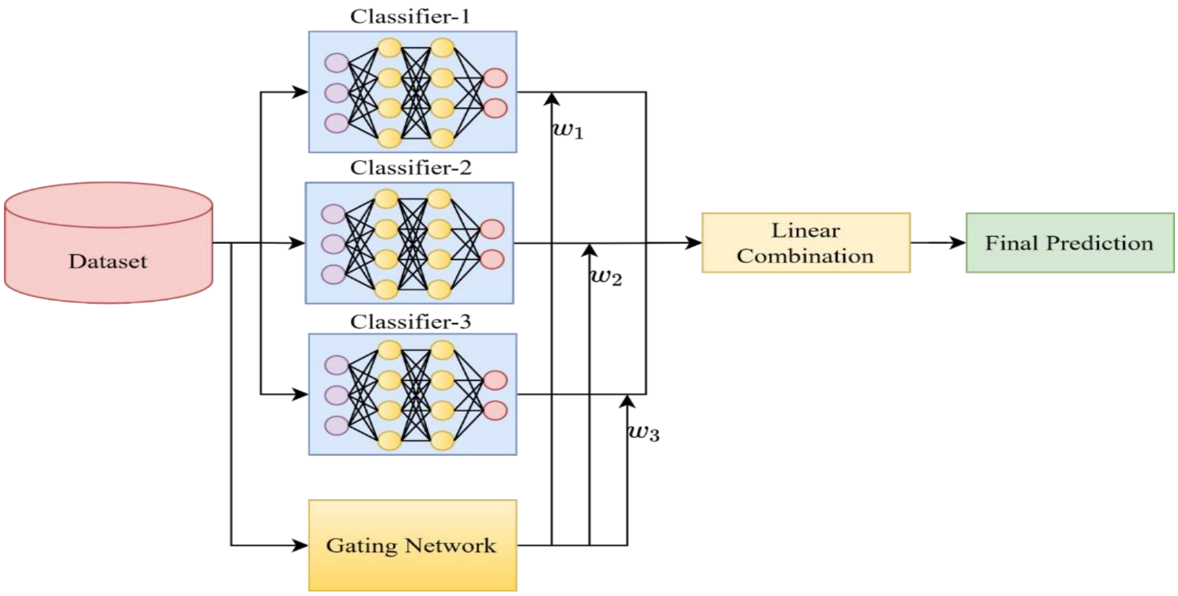

Figure 5: Mixture of Experts Ensemble Mechanism

An ensemble technique is commonly employed when multiple classifiers are trained on distinct portions of the feature space. Continuing from the previous example of the classification problem involving cats, dogs, and wolves, let's consider a scenario where one classifier is exclusively trained on data related to cats and dogs, while another classifier is trained on data related to dogs and wolves [10,11]. In our investigation, we implement the Deep Ensemble learning algorithm in two forms: Deep Ensemble without Tuning and Deep Ensemble with boosting and Performance Tuning.

2.3. Deep Ensemble without Tuning

Utilising the deep ensemble technique in a distributed setting, without any adjustments, can significantly enhance the resilience and effectiveness of a model. The concept involves training numerous deep learning models on various subsets of the data, or even the complete dataset with different initializations, and combining their predictions without adjusting hyper parameters [12,13]. This method employs a comprehensive strategy and execution approach for the distributed aspect. The phases involved are:

1. **Data Preparation:** Import and pre-process the data.
2. **Neural Network Architecture Definition**: Provide a precise definition of a neural network architecture.
3. **Distributed Training:** Employ the technique of training several instances of the model on distinct subsets of the data.
4. **Prediction Aggregation:** Consolidate the forecasts generated by all models.

2.4. Deep Ensemble with boosting and Performance Tuning

In a distributed setting, the deep ensemble without tuning can utilise Transfer Learning (TE) to enhance performance and efficiency by leveraging pre-trained models. Transfer learning is the process of utilising a pre-trained model that has been trained on a big dataset, and then adjusting it to perform well on a smaller, task-specific dataset [14]. This adjustment entails making a high-level strategy and implementing it in a distributed manner.

1. **Data Preparation:** Import and pre-process the data.
2. **Model Definition:** Utilise a pre-trained model and enhance its performance by fine-tuning.
3. **Distributed Training:** Employ the technique of training several instances of the model on distinct subsets of the data.
4. **Prediction Aggregation:** Consolidate the predictions generated by all models.

These The evaluation of deep ensemble methods is based on their predicted performance. The evaluation of classifiers has consistently relied on predictive performance measurements as the main criterion. Moreover, prediction performance indicators are regarded as objective and quantifiable, making them commonly employed to practically evaluate and compare machine learning algorithms [15,16]. To initiate the application of predictive performance, it is important to employ an appropriate dataset. The holdout technique is a common method used to evaluate prediction accuracy. It involves randomly splitting the dataset into two subsets: a training set and a test set. Alternative variations of the holdout method could be employed. Resampling data is a standard practice that involves partitioning it into distinct training and test sets. Two frequently used resampling techniques are random subsampling and n-fold cross-validation (Dai, 2013). There exist standard criteria for assessing an ensemble model. Accuracy is a widely used and straightforward metric, as defined in:

$$Accuracy = \frac{Number\ of\ true\ predictions}{Total\ number\ of\ prediction}$$

Accuracy alone may not be adequate and can be misleading when assessing an ensemble model that has uneven class distributions. In the second situation, different measures such as Recall, Precision, Specificity, and F-Measure might be employed [17,18]. Recall, often referred to as sensitivity, quantifies the capacity of the ensemble model to correctly detect positive samples, as per its definition.

$$Recall = \frac{True\ positive}{Positive}$$

The term "true positive" refers to the count of observations that are both positive and correctly identified as positive. Precision is another widely recognised performance metric. It measures the accuracy of identifying positive events. The accuracy equation can be defined formally as follows:

$$Precision = \frac{True\ positive}{True\ positive + False\ positive}$$

Similarly, specificity quantifies the model's ability to accurately detect negative samples. The equation is formally defined as

$$Specificity = \frac{True\ negative}{Negative}$$

The term "true negative" refers to the number of observations that are correctly identified as negative, while "negative" simply refers to the total number of negative observations [19,20]. There is typically a compromise between the accuracy and completeness measurements in data analysis. Attempting to improve one metric frequently leads to the decline of another. The F-Measure evaluates the trade-off between precision and recall by computing their harmonic mean [21].

$$F - Measure = \frac{2 \times Precision \times Recall}{Precison + Recall}$$

3. Experimental Analysis

In this work experimental analysis is done in two environments like distributed environment and traditional computing environment. For distributed environment Spark runtime environment is used whereas for traditional computing environment python programing environment is used. Here the analysis is done for three different types of structured datasets: Login Data Set for Risk Base Authentication, Internet Traffic Management System, Medical Recommendations System. The datasets have terabytes of data so; in distributed environment the data is divided into chunks of volumes for easy access. Whereas in traditional computing environment the Google Colab Python platform is used for easy access of data.

At initial step the pre-processing is done for three datasets individually with procedure.

- First, data cleaning with multiple sub categories like: renaming the columns, to find and remove the null columns in dataset, then checking the duplicate records, later changing the data frame from strings to numeric for easy training and testing dataset.
- Next feature Engineering is performed to find the input features because to know which data is suitable for each column and is passed to the different deep learning algorithms.
- After pre-processing, the three datasets are individually spited into 70% training and 30% testing to overcome the overfitting problems in both environments.
- Then the proposed deep ensemble algorithms like Deep Ensemble without Tuning, Deep Ensemble with boosting and Performance Tuning applied on this dataset in both the environments to know the

best environment for processing of huge structured datasets and the best proposed algorithm.

The results of three datasets are:

Data Set 1: LOGIN DATA SET FOR RISK BASE AUTHENTICATION

After applying proposed Deep Learning Algorithms on the login data set for risk base authentication in both Distributed Environment and traditional computing Environment the accuracy, time to train, predict in both distributed environment and traditional computing environment are shown in Table 1&2.

Table 1: Comparison of Algorithms in Distributed Computing Environment

Algorithms	Accuracy	Recall	Precision	F1-Score	Time to train in Sec	Time to prediction in sec	Total time in sec
Deep Ensemble Without Tuning	**10.00%**	**15.00%**	**12.00%**	**18.00%**	**2.0**	**0.5**	**0.5**
Ensemble With Boosting and Performance Tuning	**68.00%**	**63.00%**	**70.00%**	**66.00%**	**15.0**	**23.2**	**38.2**

By observing table 1, in distributed environment Ensemble with Boosting and Performance Tuning is the best algorithm in the terms of accuracy of 68% and deep ensemble without tuning is the best algorithm in terms of time 0.5.

Table 2: Comparison of Algorithms in Traditional Computing Environment

Algorithms	Accuracy	Recall	Precision	F1-Score	Time to train in sec	Time to prediction in sec	Total time in sec
Deep Ensemble Without Tuning	**60.00%**	**55.00%**	**65.00%**	**58.00%**	**25.0**	**2.5**	**27.5**
Ensemble With Boosting and Performance Tuning	**91.00%**	**91.00%**	**93.00%**	**92.00%**	**45.0**	**4.5**	**49.5**

By observing table 2, in traditional computing Ensemble with Boosting and Performance Tuning is the best algorithm with the accuracy of 91% and deep ensemble without tuning is the best algorithm in terms of time 27.5.

The prediction of dataset analysed in both environments is:

At first predicted likelihood of attacks for each timestamp is visualised in figure 6.

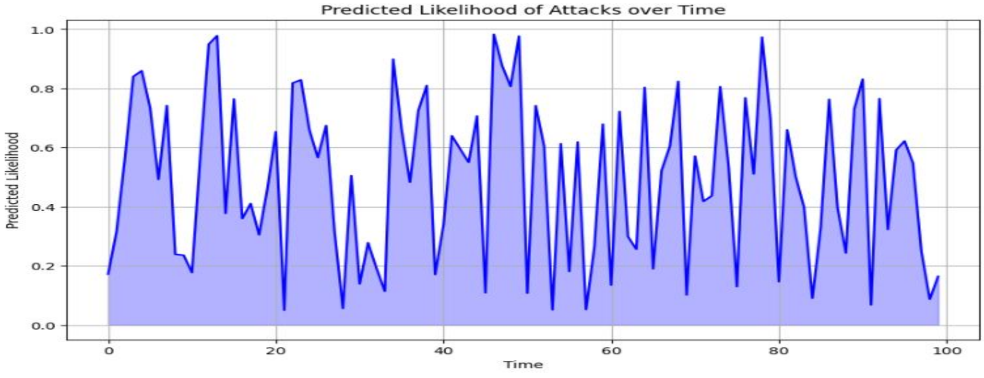

Figure 6: Predicted Likelihood of Attacks for Each Timestamp

In figure 6, X-axis is Time and Y-axis is Prediction likelihood by observing the figure across the globe at every 15seconds the internet attacks are happening.

Next predicted the distribution of attacks by OS name and OS version is visualised in figure 7.

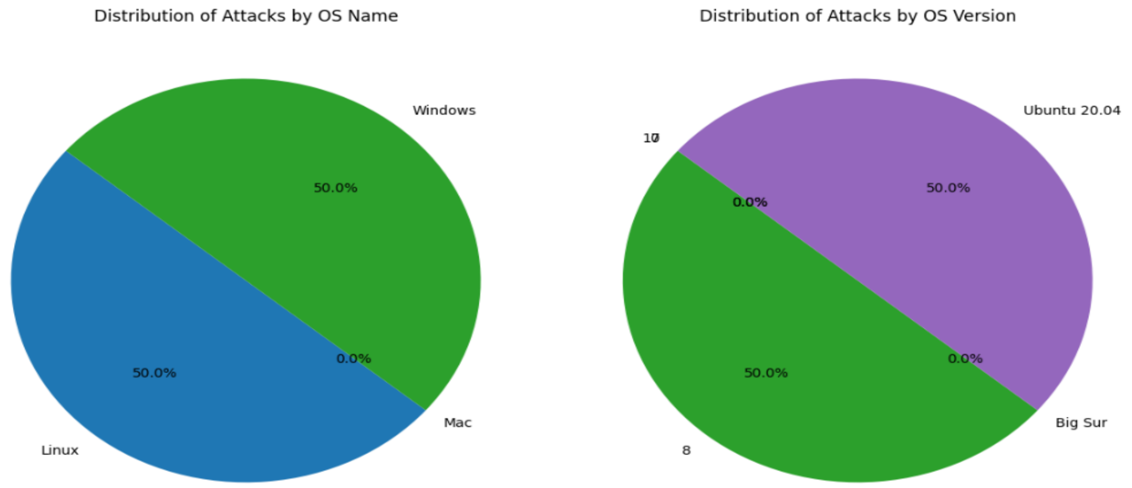

Figure 7: Distribution of Attacks by OS Name and OS Version

In figure 7, X-axis is name of the software, version of the software and Y-axis is attacks occurring at timestamp the two separate pie charts are shown the distribution of attacks by OS name and OS version. In distributed OS name Windows has high rate of attacks and in OS version windows 8 has the highest attacks.

3.1. DATA SET 2: INTERNET TRAFFIC MANAGEMENT SYSTEM

After applying proposed Deep Learning Algorithms on the internet traffic Management System in both Distributed Environment and traditional computing environment, the accuracy, time to train, predict in both distributed environment and traditional computing environment are shown in Table 3&4.

Table 3: Comparison of Algorithms in Distributed Computing Environment

Algorithms	Accuracy	Recall	Precision	F1-Score	Time to train in Sec	Time to prediction in sec	Total time in sec
Deep Ensemble Without Tuning	**85.00%**	**78.00%**	**92.00%**	**84.00%**	**10.3**	**3.1**	**23.4**
Ensemble With Boosting and Performance Tuning	**92.00%**	**89.00%**	**94.00%**	**91.00%**	**14.5**	**26.2**	**40.7**

By observing table 3, in distributed environment Ensemble with Boosting and Performance Tuning is the best algorithm with the accuracy of 92% and deep ensemble without tuning is the best algorithm in terms of time 23.4.

Table 4: Comparison of Algorithms in Traditional Computing Environment

Algorithms	Accuracy	Recall	Precision	F1-Score	Time to train in Sec	Time to prediction in sec	Total time in sec
Deep Ensemble Without Tuning	**15.00%**	**18.00%**	**12.00%**	**14.00%**	**0.3**	**3.1**	**3.4**
Ensemble With Boosting and Performance Tuning	**92.00%**	**89.00%**	**94.00%**	**91.00%**	**40.5**	**16.2**	**56.7**

By observing table 4, In traditional computing Ensemble with Boosting and Performance Tuning is the best algorithm with the accuracy of 92% and deep ensemble without tuning is the best algorithm in terms of time 3.4.

The prediction of dataset analysed in both environments is:

At first predicated the of average drops happening in service within an Internet Traffic Management System is visualised in figure 8.

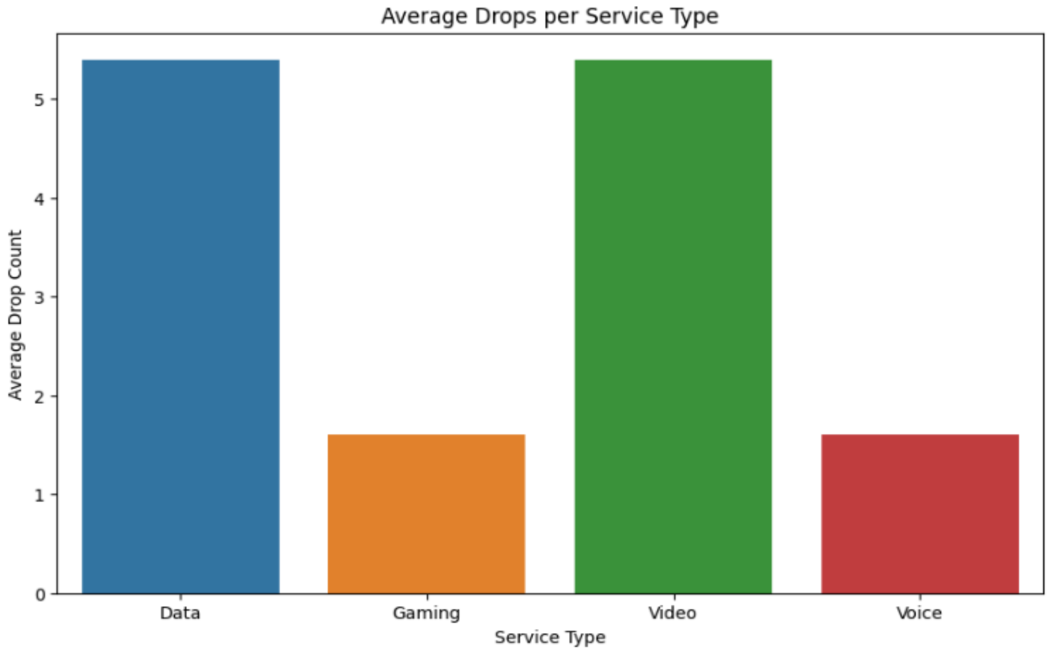

Figure 8: Average Drops Happening in Service Within an Internet Traffic Management System

In figure 8, X-axis is Service Time and Y-axis is Average Drop count by observing while data browsing and playing games internet drops is high.

Next predicted the distribution and density of the data, providing insights into the variability of user logins for each service is visualised in figure 9.

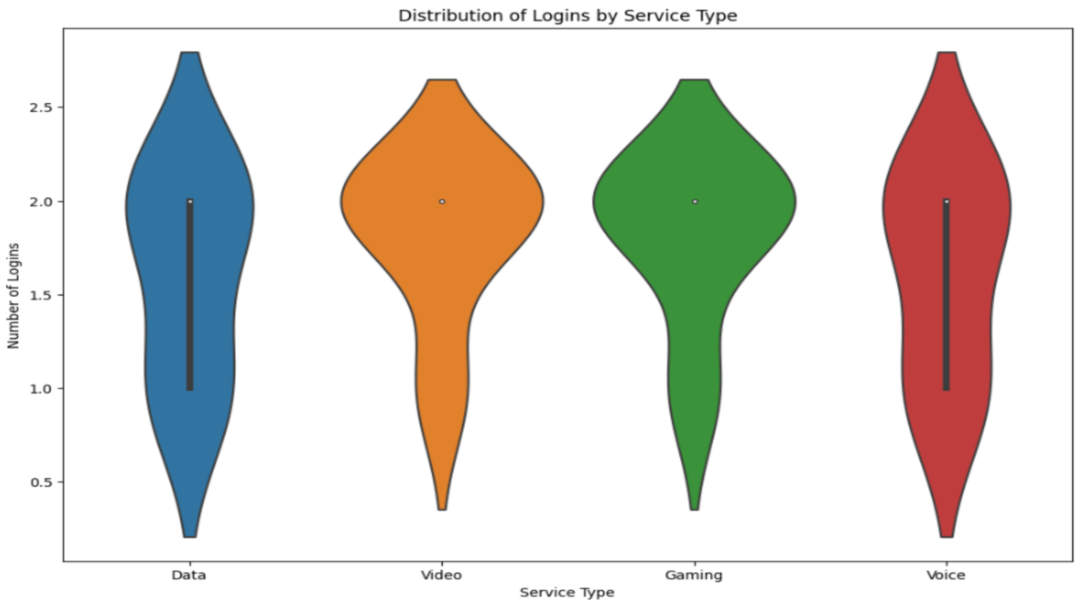

Figure 9: The Distribution and Density of the Data, Providing Insights into the Variability of User Logins for Each Service

In figure 9, X-axis is service type and Y-axis is number of logins visualizes that while data browsing and voice connection calls the distribution and density of user logins are high for each service.

3.2. DATA SET 3: MEDICAL RECOMMENDATIONS SYSTEM

After applying proposed Deep Learning Algorithms on the Medical Recommendations systems in both Distributed Environment and traditional computing Environment, the accuracy, time to train, predict in both distributed environment and traditional computing environment are shown in Table 5&6.

Table 5: Comparison of Algorithms in Distributed Computing Environment

Algorithms	Accuracy	Recall	Precision	F1-Score	Time to train in min	Time to prediction in sec	Total time in sec
Deep Ensemble Without Tuning	**85.00%**	**78.00%**	**92.00%**	**84.00%**	**20.3**	**3.1**	**23.4**
Ensemble With Boosting and Performance Tuning	**94.00%**	**91.00%**	**93.00%**	**92.00%**	**11.5**	**16.2**	**17.7**

By observing table 5, in distributed environment Deep Ensemble with Boosting and Performance Tuning is the best algorithm with the accuracy of 94% and deep ensemble without tuning is the best algorithm in terms of time 23.4.

Table 6: Comparison of Algorithms in Traditional Computing Environment

Algorithms	Accuracy	Recall	Precision	F1-Score	Time to train in min	Time to prediction in sec	Total time in sec
Deep Ensemble Without Tuning	**20.00%**	**25.00%**	**22.00%**	**28.00%**	**5.0**	**0.5**	**3.5**
Ensemble With Boosting and Performance Tuning	**62.00%**	**69.00%**	**64.00%**	**61.00%**	**20.5**	**18.2**	**38.7**

By observing table 6, in traditional computing Deep Ensemble with Boosting and Performance Tuning is the best algorithm with the accuracy of 62% and deep ensemble without tuning is the best algorithm in terms of time 3.5.

The prediction of dataset analysed in both environments is:

At first the medical service providers in India is predicated is visualised in figure 10.

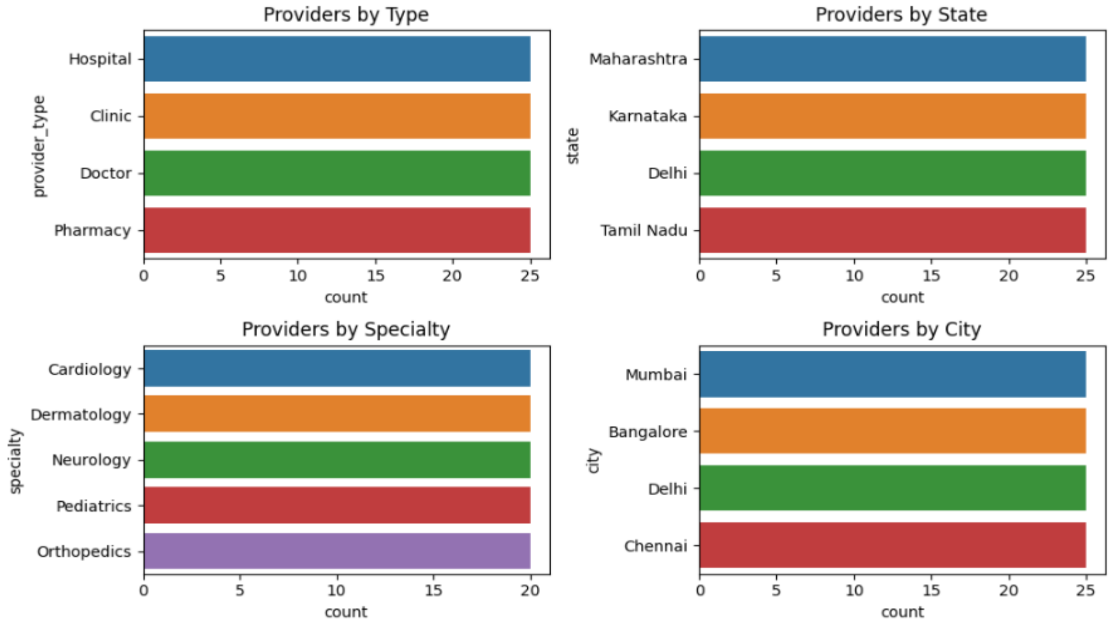

Figure 10: Medical Service Providers in India

In figure 10, X-axis is count and Y-axis is provider type, state, city, Speciality the hospitals, clinic, doctor, pharmacy's with speciality are considered by observing medical services are well established in the main states and city's so there is need of improving the medical services in the urban areas.

Next analysed area-wise clinics in India is visualised in figure 11.

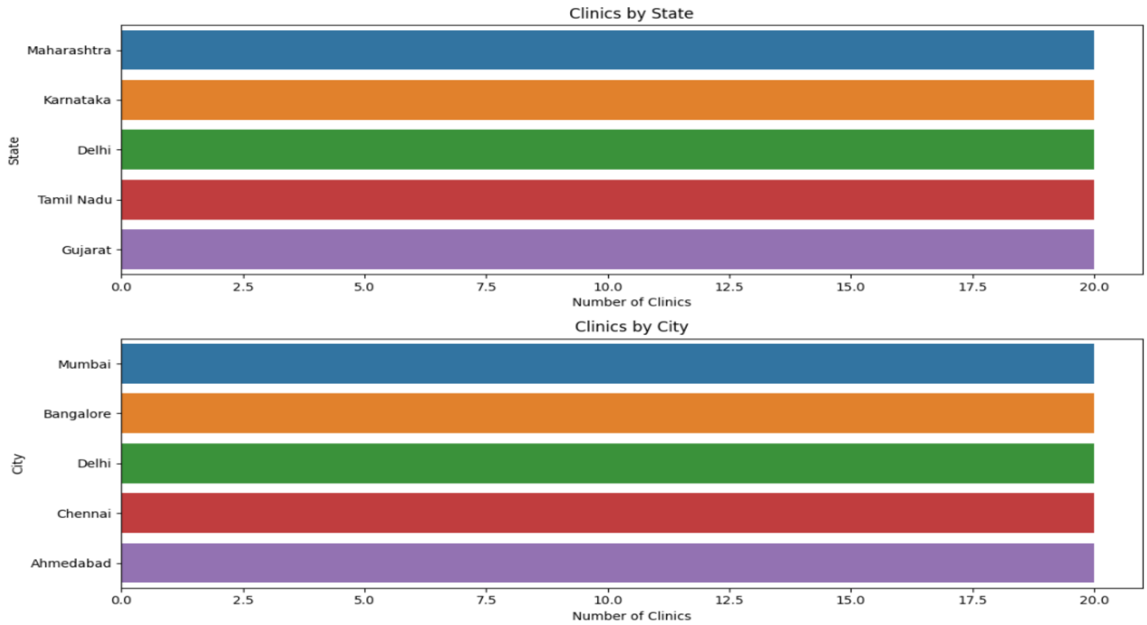

Figure 11: Area of Wise Clinics in India

In figure 11, X axis is number of clinics and Y-axis is state, City in every state observed that the main cities are well equipped with clinics and in urban areas medical services have to be improved a lot.

So by comparing both Distributed Environment and Traditional Computing Environments for analysis of different structured data by proposed deep ensemble methods: Deep Ensemble with Boosting and Performance Tuning, Deep Ensemble without Tuning. The deep ensemble with Boosting and Performance Tuning is the best algorithm in terms of accuracy but not in terms of time because model learns its training data too intimately with tuning to achieve high accuracy in structured data due to their ability to handle complex relationships whereas time is high because the sequential training process of dataset and deep ensemble without tuning is the best algorithm in terms of time but not in terms of accuracy because while training time the train models have inheritance nature of distributed drifts in parallel and have lack sequential dependencies whereas accuracy is low due to the absence of optimal hyper parameter settings in both environments.

4. Conclusion

Data has been generated at an exponential rate in recent years. Interpreting this data poses a challenge for the average individual. A distributed infrastructure is optimal for handling large datasets. In this analysis, three distinct structured datasets are examined in order to determine the optimal algorithm and environment. To that proposed a Deep Ensemble without Tuning and Deep Ensemble with boosting and Performance Tuning algorithms. These algorithms are applied on datasets in both distributed computing environment and traditional environments. Then compared the proposed algorithms with their time and accuracy in both distributed computing and traditional environments. By comparing algorithms, the Deep Ensemble with Boosting and Performance Tuning is the best algorithm based in terms of accuracy but not in time because model learns its training data too intimately with tuning and deep ensemble without tuning is the best algorithm in terms of time but not in accuracy because its inheritance nature of distributed drifts in both environments.

References

[1] K. N. Aye, (2013) "A platform for big data analytics on distributed scale-out storage system,".

[2] Zhao, Changming & Peng, Ruimin & Wu, Dongrui. (2023). "Bagging and Boosting Fine-Tuning for Ensemble Learning", IEEE Transactions on Artificial Intelligence. PP. 1-15. 10.1109/TAI.2023.3296685.

[3] Drucker, Harris & Cortes, Corinna & Jackel, Larry & Lecun, Yann & Vapnik, Vladimir. (1994). "Boosting and Other Ensemble Methods", Neural Computation. 6. 1289-1301.

[4] Eric Bauer and Ron Kohavi. (1999) "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants in Machine learning", https://link.springer.com/article/10.1023/A:1007515423169.

[5] Leo Breiman.(1996) "Bagging predictors. Machine learning", https://link.springer.com/article/10.1007/BF00058655.

[6] Odegua, Rising. (2023). "An Empirical Study of Ensemble Techniques (Bagging, Boosting and Stacking)", Int J Environ Res Public Health.; 20(6): 4977. doi: 10.3390/ijerph20064977.

[7] Zhang, W., Jiang, J., Shao, Y., Cui, B., (2020), "Snapshot boosting: a fast ensemble framework for deep neural networks", Science China Informat. Sci. 63 (1), 1–12.

[8] Sagi, O., Rokach, L., (2018), "Ensemble learning: A survey", Wiley Interdiscip. Rev.: Data Min. Knowledge Discov. 8 (4), e1249.

[9] Vilalta R. and Drissi Y. (2002), "A Perspective View and Survey of Meta-Learning", Journal of Artificial Intelligence Review, 18 (2), pp.77-95.

[10] Saso D., and Bernard Z (2004), "Is Combining Classifiers with Stacking Better than Selecting the Best One?", Machine Learning, 54, Kluwer Academic Publishers, Netherlands, pp.255–273.

[11] Domingos Pedro (1998), "Knowledge Discovery via Multiple Models", Intelligent Data Analysis, 2, pp.187-202.

[12] Ting, K. M., and Witten, I. H. (1999), "Issues in stacked generalization", Journal of Artificial Intelligence Research, 10, pp.271–289.

[13] Breiman L. (1996), "Bagging predictors", Machine Learning, vol. 24, pp.123–140.

[14] Oza N. C. and Tumer K. (2008), "Classifier ensembles: Select realworld applications," Information Fusion, vol. 9, no.1, pp. 4–20.

[15] Polikar R. (2006), "Ensemble-based systems in decision making," IEEE Circuits System Mag., vol. 6, no. 3, pp. 21–45.

[16] Rokach L. (2010), “Ensemble-based classifiers,” Artificial Intelligence Review, vol.33, pp.1-39.

[17] Islam R. and Abawajy J. (2013), “A multi-tier phishing detection and filtering approach”, Journal of Network and Computer Applications, vol. 36, pp.324–335.

[18] Kelarev A.V., Stranieri A., Yearwood J.L., Abawajy J., Jelinek H.F. (2012), “Improving Classifications for Cardiac Autonomic Neuropathy Using Multi-level Ensemble Classifiers and Feature Selection Based on Random Forest”, In Proceedings of the Tenth Australasian Data Mining Conference (AusDM 2012), Sydney, Australia, pp.93-101.

[19] Melville P. and Mooney R. J. (2005), “Creating diversity in ensembles using artificial data”, Information Fusion, vol.6, pp.99-111.

[20] Freund Y., Schapire R. (1996), “Experiments with a new boosting algorithm”, Proceedings of 13th International Conference of Machince Learning, pp. 148-56.

[21] D. Mustafa, (2022) "A Survey of Performance Tuning Techniques and Tools for Parallel Applications," in *IEEE Access*, vol. 10, pp. 15036-15055, doi: 10.1109/ACCESS.2022.3147846.