Is Deep Learning on Tabular Data Enough? An Assessment

Sheikh Amir Fayaz[1]
Research Scholar
Department of Computer Sciences
University of Kashmir, J&K, India-190006

Sameer Kaul[3]
Department of Computer Sciences
University of Kashmir
Srinagar, J&K, India-190006

Majid Zaman[2]
Directorate of IT & SS
University of Kashmir
Srinagar, J&K, India-190006

Muheet Ahmed Butt[4]
Department of Computer Sciences
University of Kashmir
Srinagar, J&K, India-190006

***Abstract*—It is critical to select the model that best fits the situation while analyzing the data. Many scholars on classification and regression issues have offered ensemble techniques on tabular data, as well as other approaches to classification and regression problems (Like Boosting and Logistic Model tree ensembles). Furthermore, various deep learning algorithms have recently been implemented on tabular data, with the authors claiming that deep models outperform Boosting and Model tree approaches. On a range of datasets including historical geographical data, this study compares the new deep models (TabNet, NODE, and DNF-net) against the boosting model (XGBoost) to see if they should be regarded a preferred choice for tabular data. We look at how much tweaking and computation they require, as well as how well they perform based on the metrics evaluation and statistical significance test. According to our study, XGBoost outperforms these deep models across all datasets, including the datasets used in the journals that presented the deep models. We further show that, when compared to deep models, XGBoost requires considerably less tweaking. In addition, we can also confirm that a combination of deep models with XGBoost outperforms XGBoost alone on almost all datasets.**

Keywords*—*Deep learning; XGBoost; NODE; TabNet; DNF-net; statistical significance test; tabular geographical data

I. Introduction

Deep learning has gained popularity in a variety of fields in recent years, including medicine, engineering, and agriculture. The exponential growth of data is most likely to blame. Deep learning algorithms have shown to be effective in a variety of domains, including audio [1], images [2], and text data [3]. Many architectures exist in these domains that are capable of converting raw data into meaningful exemplifications. Because the most common type of data is in tabular format, which consists of rows and columns with a variety of parameters, These types of data are used in real-world applications in a variety of fields, including medicine, agriculture, academia, and geography. Traditional and ensemble machine learning approaches, such as Logistic model tree (LMT), Decision tree (DT), Random forest (RF), Gradient Boosted decision tree (GBDT), and others, are used to process these tabular datasets, and these models still outperform deep learning on tabular data. When using a deep learning model on tabular data, there are a number of issues to consider, including missing data, data integrity i.e., mixed data (nominal, numerical, and categorical), data imbalance, data overfitting, and a lack of specific knowledge about the dataset's structure. When tabular data is taken into account, boosting machine-learning algorithms like XGBoost perform better, according to the "no free lunch" (NFL) theorem [4] [5]. Since then, the authors [6] [7] have implemented deep learning on the tabular dataset in their research, and it has been demonstrated that the deep learning model outperforms GBDT. However, because each study was conducted on different datasets, one of the major flaws in their approach is that there was no benchmark dataset [8] [9]. So, based on these papers alone, it's difficult to claim that deep learning always outperforms traditional and ensemble algorithms like GBDT when dealing with tabular data [10].

Since the number of research studies using deep learning on tabular data is growing, there is no standard benchmark model in deep learning from which we can conclude that deep learning always outperforms traditional machine learning on tabular data. As a result, the main goal of this paper is to see if any deep learning model is a good fit for these types of tabular dataset problems. Furthermore, in this paper, we attempt to evaluate the proposed deep learning models on tabular datasets, as well as implement XGBoost on various algorithms, with a focus on a historical geographical dataset from India's Kashmir province [11].

This paper is structured as: Section 2 provides a basic background of deep learning and ensemble models on the tabular data. Next, Section 3 presents the experimental setup where dataset descriptions are presented and furthermore this section defines the implementation details with optimization parameters and statistical significance test. Section 4 defines the experimental results and model evaluation. Section 5 defines the overall working of the paper. Finally, the conclusion and future strategies have been suggested in Section 6.

II. Review of Literature

In this section, we present studies that used deep learning approaches and ensemble approaches to predict rainfall using a tabular geographical dataset. This section is divided into two subsections: Section 1 contains several studies that use deep learning models on tabular datasets, and Section 2 contains some model ensemble approaches that use the same tabular geographical dataset and record individual performances.

A. Deep Learning on Tabular Geographical Dataset

Salman et al. [12] (2015) use a variety of deep learning techniques, including recurrence neural networks (RNN), convolutional neural networks (CNN), and conditional restricted Boltzmann machines (CRBM), to look for hidden patterns in the dataset. These techniques were used in the Indonesian region, with data collected from the national weather service center for environmental forecasting (NOAA). This study used a dataset that spanned 35 years, from 1973 to 2009. Initially, RNN was applied to a dataset containing ESNO variables. RNN produces results with a higher level of accuracy, according to the findings.

Emiley et al [13] (2016) present a deep learning architecture-based accumulated daily rainfall prediction. This research employs auto encoders to reduce non-linear attribute relationships and a multi-layer perceptron (MLP) for prediction. This hybrid architecture was then compared to previously implemented techniques, and it was discovered that the model performs better for daily rainfall prediction when using root mean square error (RMSE) and mean squared error (MSE) statistical approaches. This research was carried out in the Colombian city of Manziles, where the data was grouped into a daily time series spanning the years 2002 to 2013.

Devi et al. [14] (2017) propose an artificial neural network (ANN) model for a reliable forecast mechanism. This method was used to analyze spatial and temporal data from the Nilgiris district in Tamil Nadu, India. Performance was measured using a variety of statistical parameters such as correlation coefficient, MSE, and so on. When compared to time delay neural network (NN) and other ANN models, the best model in this study is a wavelet Elman model. This research also develops a system for early landslide warnings based on the wavelet Elman model.

According to Geetha et al. [15] (2018), using deep learning techniques for meteorological purposes on a time series dataset will significantly improve accuracy precision. This research uses deep learning architectures such as LSTM and ConvNet to analyze time series data from 468 months in various locations. Later, it was discovered that increasing the number of hidden layers improves the model's performance for daily rainfall prediction when using RMSE and MAPE statistical approaches.

Yen et al. [16] (2019) proposed using Echo state network (ESN) and deep Echo state network (DeepESN) algorithms to apply deep learning models to rainfall prediction. This research uses hourly rainfall data from southern Taiwan from 2002 to 2014, spanning a period of 12 years. When the DeepESN algorithm's correlation coefficient was compared to ESN and commercial neuronal network algorithms like BPNN and SVR, the study concluded that it is a reliable algorithm. It was suggested that DeepESN could be used globally on larger sets of data to predict rainfall based on the results obtained.

Manoj et al. [17] (2020) proposed a hybrid deep learning model (BLSTM-GRU), for the monthly prediction of rainfall. The experiment was conducted using data obtained from Bhutan's National Center of Hydrology and Meteorology Department (NCHM). To test the data's predictive capability, various NN algorithms such as LSTM, CNN, BLSTM, and GRU were used. LSTM outperforms the other techniques with a MSE score of 0.0128, but the hybrid model BLSTM-GRU outperforms LSTM by approximately 41% with a MSE score of 0.0075.

Zeelan et al. [18] (2020) claimed that deep learning models can learn from nonlinear data with less error. The Multi-layer perceptron (MLP) and Auto-encoder NN are used in this study to predict the rainfall. The accuracy parameters used in this study were RMSE and MSE, and these implemented models were later compared with other machine learning models on the same set of data, with the study concluding that MLP and Auto-encoder NN perform significantly and can be used as a solution to all available approaches.

Ari Yari et al. [19] (2021) present a rainfall prediction comparative analysis study. The authors use deep learning (DL) models and simple rainfall estimation approaches based on traditional machine learning algorithms. The study was conducted in five major cities across the United Kingdom (UK), with data collected spanning roughly 20 years (2000-2020). The bidirectional LSTM network and stacked LSTM with two hidden layers performed best after the proposed model was evaluated. One of the study's major flaws was the model's inability to generalize the data. That is, the model over-fits the training data in most cases, which makes it difficult to record accurate, predicts in the testing and validation sets.

Razeef et al. [20][21] (2020,2022) proposed a neural network approach to predict the rainfall on the time series data of UT of J&K, India. Rainfall was predicted using a Grey Wolf-based neural network model. The data in this study spans 30 years, from 1990 to 2020, and includes variables such as maximum temperature, humidity, minimum temperature, wind, vapor pressure, and others. When using RMSE, PRD values, and MSE statistical approaches, it was discovered that the model performs better for daily rainfall prediction. This model was later compared to non-linear autoregressive models with exogenous inputs (NARX), and the study concluded that when both models are used together, non-linear time series data would perform better.

According to the literature reviewed in this study, many deep learning models have been utilized for various time-series prediction applications, but they have yet to become a standard algorithm in the artificial intelligence arena. We must analyze the performance of these models using varied threshold datasets, and these techniques must be re-evaluated as a result.

B. Model Ensembles on Tabular Geographical Dataset

Zaman et al. [22] (2019) use an ensemble distributed decision tree (DDT) approach to improve classification accuracy on a historical geographical dataset. The experiment was conducted on a tabular dataset containing approximately 6000 records with five different parameters. When the DDT approach was used, there was no performance improvisation, according to this study.

Patil et al. [23] (2020) use machine learning algorithms to forecast rainfall based on a variety of variables such as temperature, humidity, wind speed, and rainfall. These algorithms include linear regression and NN, and the type of data fed to it, according to the study, determines the accuracy of the algorithm. That is, when the dataset of different structures is used, we may get different accuracies and require some modifications. Furthermore, the accuracy of DT's was found to be superior to other techniques used on the same type of data in this study.

Sheikh et al [24] (2021) proposed a stepwise machine learning approach on the discrete data collected from the Indian Meteorological Department (IMD), Pune India. The implemented model, known as LMT, employs logistic regression functions at the DT's leaf nodes. The logistic functions on the leaf nodes combine the final output of the constructed DT into linear models, which were examined and revealed a significant improvement in accuracy performance. The accuracy of the constructed DT on the same set of data is 66 percent, but when the logistic functions are applied to the leaf nodes, the accuracy jumps to 87 percent. The dataset used in this study was from J&K's Kashmir province, and it covered the years 2012 to 2017, with around 6000 data rows.

Since there are other ensemble [25-29] and deep learning approaches such as NODE, TabNet, DNF-Net, and Boosting (XGBoost, CatBoost, GBDT) [30-33]. These models perform better on larger datasets, and we use the entire training dataset to train the model.

III. Experimental Setup

A. Dataset Description

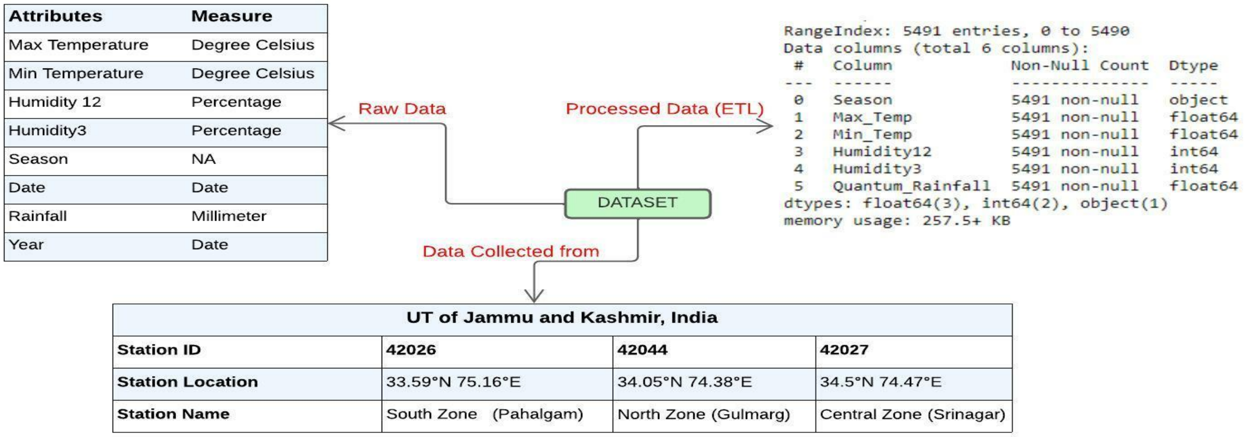
In this study, we employed a variety of tabular datasets from diverse fields which are used in various classification and regression problems. Some of these datasets have heterogeneous features, while others have just homogeneous features. There are approximately seven tabular datasets that have already been used by various academics in their publications, and we have used one additional dataset that has yet to be used by any researcher. In the experimental operations, roughly 80000 samples were taken, and the datasets range from 7 to 1600 parameters. The seven datasets are obtained from TabNet, NODE, and DNF-Net studies, and each dataset has been well-trained and preprocessed in its respective paper. These datasets include Blastchar [34] (Source: Kaggle), Higgs Boson [35] (Source: Kaggle), Microsoft MSLR [36](Source: MSLR-WEB10K), Forest Cover Type [37] (Source: Kaggle), Epsilon [38] (Source: PASCAL Challenge 2008), YearPrediction [39] (Source: Million Song Dataset) and Gas concentrations (Source: OpenML) [40]. These datasets have also been adjusted and relative values were calculated, resulting in a standardized data with a zero mean value and unit variance. As a result, we won't go into detail about these datasets in this study; instead, we'll just establish the historical geographical dataset that we will be implementing latter. The historical geographical dataset has been collected from three different locations in Jammu and Kashmir's UT. These three locations are in the province of Kashmir, but they are quite far apart. The data spans five years, from 2012 to 2017. At these locations, the average annual rainfall is around 1700 mm. The data consists of 5491 records with a total of 9 explanatory characteristics, including minimum temperature (°C), maximum temperature (°C), station ID, season, year, humidity at various intervals, and the target parameter rainfall, which shows the quantum of rainfall measured in millimeters [41-43].

In Fig. 1, the reader can find a brief description of the data. It has gone through an ETL (Extract, Transform, and Load) process to achieve data integrity, normalization, and standardization.

To normalize the dataset's range of parameters, we use the function (1), as given below:

$$X = \frac{X - \mu}{\sigma} \tag{1}$$

The data was scaled using the R tool's built-in function 'Scale.' We also use relative values of each attribute to normalize the training data. The tabular (Table I) and graphical (Fig. 2) representation of the dataset is shown.



Attributes	Measure
Max Temperature	Degree Celsius
Min Temperature	Degree Celsius
Humidity 12	Percentage
Humidity3	Percentage
Season	NA
Date	Date
Rainfall	Millimeter
Year	Date

```
RangeIndex: 5491 entries, 0 to 5490
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Season            5491 non-null   object
 1   Max_Temp          5491 non-null   float64
 2   Min_Temp          5491 non-null   float64
 3   Humidity12        5491 non-null   int64
 4   Humidity3         5491 non-null   int64
 5   Quantum_Rainfall  5491 non-null   float64
dtypes: float64(3), int64(2), object(1)
memory usage: 257.5+ KB
```

UT of Jammu and Kashmir, India			
Station ID	42026	42044	42027
Station Location	33.59°N 75.16°E	34.05°N 74.38°E	34.5°N 74.47°E
Station Name	South Zone (Pahalgam)	North Zone (Gulmarg)	Central Zone (Srinagar)

Fig. 1. Historical Geographical Dataset Description.

TABLE I. TABULAR REPRESENTATION OF GEOGRAPHICAL DATASET WITH RELATIVE VALUES

MAX	Relative Frequency (MAX)	MIN	Relative Frequency (MIN)	Hum 12	Relative Frequency (Hum12)	Hum 3	Relative Frequency (Hum3)	Rf	Relative Frequency (Rf)
20	0.010562739	8	0.00910581	60	0.024767802	93	0.032052449	0	0.685303223
19	0.008195229	1.6	0.008923693	50	0.024039337	92	0.032052449	0.4	0.014751411
27.5	0.008013112	11.2	0.008741577	53	0.023675105	88	0.029867055	0.6	0.012930249
29	0.00764888	9	0.008741577	51	0.023128756	91	0.029684939	0.2	0.012019669
14	0.007102531	0.2	0.008741577	55	0.022764524	89	0.029684939	1.2	0.011291204
22	0.006920415	0.4	0.008559461	54	0.022764524	87	0.029502823	0.8	0.009287926
16	0.006738299	10.2	0.008377345	59	0.022218175	85	0.027317429	1.4	0.008195229

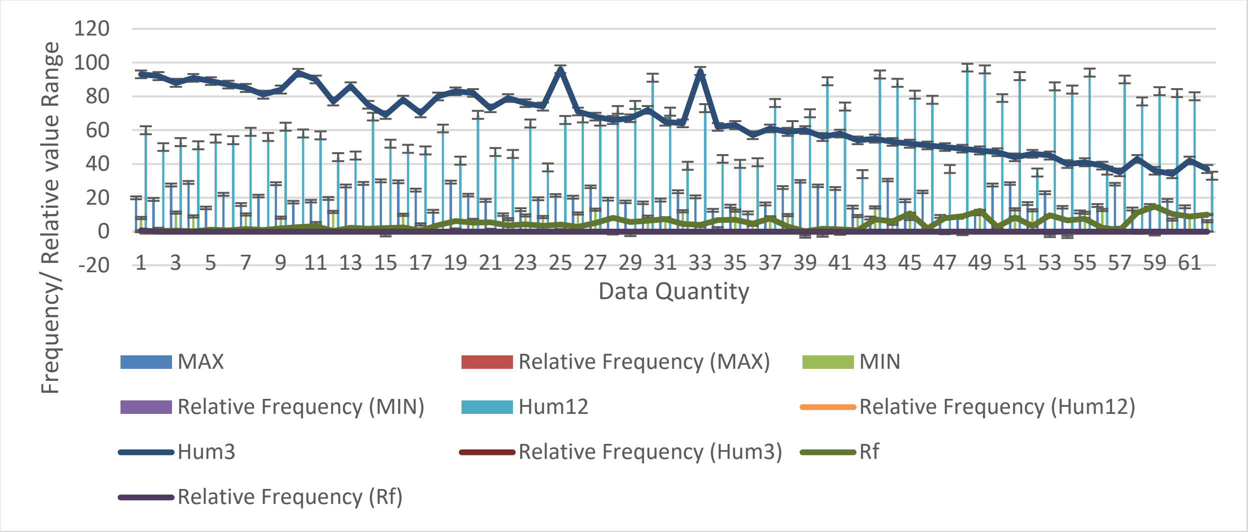


Fig. 2. Graphical Representation of Geographical Dataset with Relative Values.

A total of 70% of the data is used for training, while 15% is used for validation and testing, i.e. 3844 samples were randomly selected for training, 823 samples for validation, and the remaining 823 samples were selected for testing.

Thus, the overall description of the tabular datasets used in this paper is shown in Table II.

B. Implementation Details

1) Optimization process: To pick the model hyperparameters during the optimization phase, we used the HyperOpt parameter-tuning package. To optimize the results on the validation set, this technique first uses Bayesian optimization, followed by hyperparameter search on each dataset utilized in this study. There were around 7-9 main hyperparameters, which in the case of a deep learning model include the number of nodes, layers, and, most importantly, the learning rate.

To optimize the hyperparameters all the datasets used in this study were initially divided into three individual splits, which include training split, testing split and validation split. In partitioning process, we use stratified random sampling partitioning to split the data. The below tabular representation (Table III) shows the individual splits of the datasets in order to optimize the model.

Around 1000 steps of search were performed on each set of data in order to maximize the validation set's findings, and only the set of hyperparameters with the smallest loss for the final configuration were chosen.

2) Metrics evaluation and statistical significance test: In the case of classification issues using discretized data, we simply utilize cross-entropy loss metrics to evaluate the datasets. It calculates the impurity at each stage of the data and the total entropy loss in the end. Furthermore, when the data is continuous in nature, such as in regression situations, statistical parameters such as RMSE, mean signed difference (MSD), and MAE are used. We reported the performance of each dataset on their respective test sets based on these metrics. We also have Friedman's testing for statistical significance in addition to these cross entropy and RMSE, MAE measurements. Friedman's testing has the advantage of assuming that data is not evenly distributed. Using Friedman's hypothesis, we compare all of the classifiers to the baseline classifier. The null hypothesis is rejected at a certain level of confidence (90 percent in this study) if the p-value for any model pair is less than 0.05; otherwise, the hypothesis is not rejected.

TABLE II. TABULAR DATASETS DESCRIPTION

Dataset	Parameters	Records (Approx.)	Source	Research Paper	Hyperparameters and search spaces used for configuring each algorithms implemented	Ref
Blastchar	20	7000	Kaggle	DNF-Net	• Discrete uniform distribution for n. formulas and Number of layers. • Learning rate with log-distribution. • Batch size with Uniform choice.	[34]
Higgs Boson	30	80000	Kaggle	TabNet	• Log-Uniform distribution for Learning rate • Discrete uniform distribution for feature dimensions, n steps and output dimensions. • Uniform distribution for relaxation factor, Batch size and bn epsilon.	[35]
Microsoft(MSLR)	126	75000	MSLA-WEB10K	NODE	• Log-Uniform distribution for learning rate. • Discrete uniform distribution for Num Layers, tree output dimensions and tree depth. • Uniform choice for Batch size.	[36]
Forest Cover Type	50	55000	Kaggle	TabNet	• Log-Uniform distribution for Learning rate • Discrete uniform distribution for feature dimensions, n steps and output dimensions. • Uniform distribution for relaxation factor, Batch size and bn epsilon.	[37]
Epsilon	1700	50000	PASCAL Challenge 2008	NODE	• Log-Uniform distribution for learning rate. • Discrete uniform distribution for Num Layers, tree output dimensions and tree depth. • Uniform choice for Batch size.	[38]
Year Prediction	90	51500	Million Song Dataset	NODE	• Log-Uniform distribution for learning rate. • Discrete uniform distribution for Num Layers, tree output dimensions and tree depth. • Uniform choice for Batch size.	[39]
Gas Concentration	129	13900	OpenML	DNF-Net	• Discrete uniform distribution for n. formulas and Number of layers. • Learning rate with log-distribution. • Batch size with Uniform choice.	[40]
Historical geographical Dataset	9	5491	IMD	New Dataset	---	[41-43]

TABLE III. TABULAR REPRESENTATION WITH TRAINING TESTING AND VALIDATION SPLITS

Dataset	Records	Training	Testing	Validation
Blastchar	7000 (100%)	5600 (80%)	---	1400 (20%)
Higgs Boson	80000 (100%)	50000 (62%)	---	30000(38%)
Microsoft(MSLR)	75000 (100%)	60000 (80%)		15000 (20%)
Forest Cover Type	55000 (100%)	38500 (70%)	8250 (15%)	8250 (15%)
Epsilon	50000 (100%)	40000 (80%)	---	10000 (20%)
Year Prediction	51500 (100%)	41200 (80%)	---	10300 (20%)
Gas Concentration	13900 (100%)	9730 (70%)	2780 (20%)	1390 (10%)
Historical geographical Dataset	5491 (100%)	3844 (70%)	823 (15%)	823 (15%)

IV. EXPERIMENTAL RESULTS

A. *How Effectively can Deep Learning Models Generalize to other Datasets?*

The performance of deep learning models on the aforementioned datasets is proposed in this study, and the individual outcomes are compared to the XGBoost technique. The performance of each algorithm on each dataset is presented in the table below. The mean and standard error of each model's performance on the datasets are shown in the Table IV. The best performance of the dataset is presented for each model, and it was discovered that the model with the lowest value is considered to have the best performance. Friedman's testing was utilized to perform a statistical significance test between the models with a 90% confidence level.

TABLE IV. Results and Performance of each Tabular Dataset based on each Model used in this Study. For YearPrediction MSE is used and Cross Entropy is used for all other Datasets. The Values with Lower Value is better and these Values are the Averages of different Training runs with Standard Error of Mean (SEM)

Model	**Blastchar**	**Higgs Boson**	**Microsoft (MSLR)**	**Forest Cover Type**	**Epsilon**	**Year Prediction**	**Gas Concentration**	**Historical Geographical Dataset**
NODE	21.36 ± 0.23	21.21 ± 0.67	54.62 ± 3e-2	4.25 ± 0.17	**10.26 ± 1e-2**	76.88 ± 0.16	2.25 ± 0.22	14.76 ± 0.12
DNF-Net	27.91 ± 0.18	23.71 ± 0.88	55.78 ± 3e-2	4.01 ± 0.09	12.42 ± 4e-2	82.06 ± 0.15	**1.45 ± 0.08**	15.36 ± 0.18
TabNet	23.66 ± 0.16	**21.15 ± 0.22**	55.09 ± 2e-2	**3.02 ± 0.15**	11.96 ± 3e-2	82.89 ± 0.11	1.86 ± 0.10	14.62 ± 0.16
XGBoost	20.41 ± 0.23	21.83 ± 0.34	54.39 ± 2e-2	3.21 ± 0.11	11.23 ± 2e-2	75.68 ± 0.08	2.06 ± 0.32	13.45 ± 0.19
Simple Ensemble	21.22 ± 0.15	22.49 ± 0.41	54.44 ± 3e-2	4.15 ± 0.16	11.38 ± 4e-2	78.65 ± 0.16	2.41 ± 0.18	13.67 ± 0.15
Deep Ensemble & XGBoost	**20.13 ± 0.16**	22.36 ± 0.51	**54.21 ± 1e-2**	2.86 ± 0.05	11.35 ± 1e-2	**75.01 ± 0.22**	1.66 ± 0.06	**12.13 ± 0.15**
Deep Ensemble w/o XGBoost	24.36 ± 0.31	22.45 ± 0.55	55.53 ± 3e-2	3.57 ± 0.11	10.88 ± 1e-2	79.01 ± 0.17	1.91 ± 0.17	14.15 ± 0.14

There are some observations based on the results, as given in the table. To begin with, the models almost outperform unknown datasets on original datasets. On each dataset, the XGBoost model nearly outperformed all deep learning models such as NODE, DNF-Net, and TabNet. As we can see, the XGBoost model outperforms deep learning models in 5 of the 8 datasets, and these datasets had significant p-values (< 0.05), indicating that the results were significant. We can also see that the deep learning model has not consistently performed. The authors claimed in their study that deep learning models outperform other models, but this was only true for the datasets included in their study. As a result, when distinct datasets are involved, this conclusion is unjustifiable. We can also observe that the Deep ensemble and XGBoost model beats individual models in the majority of cases, i.e. it outperforms 5 individual models out of 8, and the p-value in these 5 cases was substantially less than 0.05, indicating that the null hypothesis is rejected.

Now, in order to evaluate these models and see which one is better for a given dataset, we compared the relative performance of each model (NODE [44], TabNet [45], DNF-Net [46][47], and so on) to the best model for that dataset. For example, assume we used the historical geographical dataset in table (Table IV) and compared the relative performance of the models to choose the model with the best performance (Deep Ensemble & XGBoost in this case). We discovered that Deep Ensemble & XGBoost had the best relative value gain of 2.46 percent, with XGBoost coming in second with 3.86 percent, TabNet with 8.67 percent, DNF-Net with 10.55 percent, and NODE with 13.23 percent. The tabular representation of average relative performance deterioration on unseen datasets is shown (Table V).

With these findings, we discovered that deep learning does not always outperform other methods. When compared to XGBoost, Deep Ensemble, and XGBoost, the deep learning model performs the worst when trained on datasets other than those used in the original studies. Only two choices exist for the lowest performance results. Either there is a selection bias or there is a difference in hyperparameter optimization. Furthermore, the results in the original papers reflect the results that we have reported, excluding the possibility that implementation errors were the cause of our observation.

B. Model Evaluation: Is it required to apply both the XGBoost and Deep Models in Combination?

In this section, we will see which model performs better in all scenarios when compared to other models. We employed four types of models in table (Table V), including deep models (TabNet, DNF-Net, and NODE), XGBoost, Deep ensemble with XGBoost, and Deep ensemble without XGBoost. When comparing the performance of deep learning models to XGBoost and combined Deep ensemble & XGBoost Models on various data types, we discovered that deep learning models perform poorly in most circumstances. The question now is whether we require a combined XGBoost and Deep model. To answer this, we can see that in 6 of the 8 examples, the combined ensemble and XGBoost show significant results. Simple ensemble did not produce any improvised results, although competing with deep learning model results. Furthermore, when we look at the Deep ensemble models without XGBoost, we can observe that it did not do well in any situation when compared to any other model. As a result of this analysis, we can conclude that for tabular datasets, we require both deep ensemble and XGBoost in combination.

TABLE V. Average Values of all the Models Implemented on each Dataset with Lower Value Treated as Best

Model	**Average Relative performance (%)**
Deep Ensemble w/o XGBoost	7.10 %
XGBoost	3.86%
Deep Ensemble & XGBoost	2.46%
TabNet	8.67%
DNF-Net	10.55%
NODE	13.23%
Simple Ensemble	4.23%

In real-world situations, time and resources are limited when it comes to training a model for a new dataset and optimizing its hyperparameters. As a result, it's fascinating to learn how difficult it is to do so for each model. Calculating the number of computations required by the model is one way to assess this. Floating point operations per second (FLOPS) is a common unit of measurement. However, because each parameter set has a different FLOPS number, comparing various models in this way when optimizing model parameters has become impossible [47].

V. Discussion

This study was based on deep models that had already been deployed by several academics on a tabular dataset [12-14]. Deep models were applied to tabular datasets (Forest CoverType, Higgs, Gas Concentration, Epsilon [30], MSLR [31], Year Prediction [32], Blastchar [33], and so on) by the authors in their publications, and they argued that deep models exhibit some promising outcomes. However, their research was limited to a single dataset. We used one more tabular dataset (Geographical dataset) in this research and attempted to construct all of the deep learning and ensemble models. On all of the datasets utilized in this study, we also investigated various possible tradeoffs that are required in real-time applications, such as hyperparameter tuning, metrics evaluation, and Statistical Significance test. Our results reveal that the performance is similar to what the authors have shown in their respected publications, but when we tried to compare the performance of different datasets on the models used by the authors in their study, the deep learning results were not as good as the original datasets. We next looked at XGBoost and ensembles of deep models with XGBoost and without XGBoost, and discovered that the XGBoost model outperforms deep models. However, as seen in the table, the ensemble of XGBoost models with Deep models outperforms the XGBoost model alone. Furthermore, optimizing a new dataset using deep models is a difficult procedure, whereas optimizing a new dataset using ensemble models with XGBoost is quite simple [48].

VI. Conclusion and Future Strategies

This research demonstrates that using various deep learning algorithms on tabular data does not improve performance. We also used XGBoost on these datasets, which produced some promising results when compared to deep models, and we used ensemble deep learning with and without XGBoost to see how it affected the performance of each dataset. On these tabular datasets, an ensemble of XGBoost models without deep learning never performed well, but when we looked at the overall performance using an ensemble of deep models with XGBoost, the results were astounding. This ensemble deep model with XGBoost beats all previous models, and our enhanced models pave the way for future study on tabular datasets in terms of comparing performance and assisting researchers in determining the best technique for optimizing hyperparameters. Our findings will also aid in the development of new models (such as CatBoost, where learning rates are uniformly distributed) that are simple to optimize and can compete with the performance of ensemble deep models such as XGBoost and many others.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: NAACL, 2019.

[2] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[3] van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, WaveNet: A generative model for raw audio, in: Proc. 9th ISCA Workshop on Speech Synthesis Workshop (SSW 9), 2016, p. 125.

[4] Adam, Stavros P., Stamatios-Aggelos N. Alexandropoulos, Panos M. Pardalos, and Michael N. Vrahatis. "No free lunch theorem: A review." Approximation and optimization (2019): 57-82.

[5] Popov, Sergei, Stanislav Morozov, and Artem Babenko. "Neural oblivious decision ensembles for deep learning on tabular data." arXiv preprint arXiv:1909.06312 (2019).

[6] Gorishniy, Yury, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. "Revisiting deep learning models for tabular data." Advances in Neural Information Processing Systems 34 (2021).

[7] Marais, Jan André. "Deep learning for tabular data: an exploratory study." PhD diss., Stellenbosch: Stellenbosch University, 2019.

[8] Shwartz-Ziv, Ravid, and Amitai Armon. "Tabular data: Deep learning is not all you need." Information Fusion 81 (2022): 84-90.

[9] Richman, Ronald, and Mario V. Wuthrich. "LocalGLMnet: interpretable deep learning for tabular data." Available at SSRN 3892015 (2021).

[10] Luo, Renqian, Xu Tan, Rui Wang, Tao Qin, Enhong Chen, and Tie-Yan Liu. "Neural architecture search with gbdt." (2020).

[11] Zaman, Majid, Sameer Kaul, and Muheet Ahmed. "Analytical comparison between the information gain and Gini index using historical geographical data." Int. J. Adv. Comput. Sci. Appl 11, no. 5 (2020): 429-440.

[12] Salman, Afan Galih, Bayu Kanigoro, and Yaya Heryadi. "Weather forecasting using deep learning techniques." In 2015 international conference on advanced computer science and information systems (ICACSIS), pp. 281-285. Ieee, 2015.

[13] Hernández, Emilcy, Victor Sanchez-Anguix, Vicente Julian, Javier Palanca, and Néstor Duque. "Rainfall prediction: A deep learning approach." In International Conference on Hybrid Artificial Intelligence Systems, pp. 151-162. Springer, Cham, 2016.

[14] Renuga Devi S., Arulmozhivarman P., Venkatesh C. (2017) ANN Based Rainfall Prediction—A Tool for Developing a Landslide Early Warning System. In: Mikoš M., Arbanas Ž., Yin Y., Sassa K. (eds) Advancing Culture of Living with Landslides. WLF 2017. Springer, Cham. https://doi.org/10.1007/978-3-319-53487-9_20.

[15] Aswin, S., P. Geetha, and R. Vinayakumar. "Deep learning models for the prediction of rainfall." In 2018 International Conference on Communication and Signal Processing (ICCSP), pp. 0657-0661. IEEE, 2018.

[16] Yen, Meng-Hua, Ding-Wei Liu, Yi-Chia Hsin, Chu-En Lin, and Chii-Chang Chen. "Application of the deep learning for the prediction of rainfall in Southern Taiwan." Scientific reports 9, no. 1 (2019): 1-9.

[17] Chhetri, Manoj, Sudhanshu Kumar, Partha Pratim Roy, and Byung-Gyu Kim. "Deep BLSTM-GRU model for monthly rainfall prediction: A case study of Simtokha, Bhutan." Remote sensing 12, no. 19 (2020): 3174.

[18] Basha, Cmak Zeelan, Nagulla Bhavana, Ponduru Bhavya, and V. Sowmya. "Rainfall prediction using machine learning & deep learning techniques." In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 92-97. IEEE, 2020.

[19] Barrera-Animas, Ari Yair, Lukumon O. Oyedele, Muhammad Bilal, Taofeek Dolapo Akinosho, Juan Manuel Davila Delgado, and Lukman Adewale Akanbi. "Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting." Machine Learning with Applications 7 (2022): 100204.

[20] Mohd, Razeef, Muheet Ahmed Butt, and Majid Zaman Baba. "Grey Wolf-Based Linear Regression Model for Rainfall Prediction."

International Journal of Information Technologies and Systems Approach (IJITSA) 15, no. 1 (2022): 1-18.

[21] Mohd, Razeef, Muheet Ahmed Butt, and Majid Zaman Baba. "GWLM–NARX: Grey Wolf Levenberg–Marquardt-based neural network for rainfall prediction." Data Technologies and Applications (2020).

[22] Fayaz, Sheikh Amir, Majid Zaman, and Muheet Ahmed Butt. "To ameliorate classification accuracy using ensemble distributed decision tree (DDT) vote approach: An empirical discourse of geographical data mining." Procedia Computer Science 184 (2021): 935-940.

[23] Patil, Deepali, Shakib Badarpura, Abhishek Jain, U. G. Student, and Aniket Gupta. "Rainfall Prediction using Linear approach & Neural Networks and Crop Recommendation based on Decision Tree."

[24] Fayaz, Sheikh Amir, Majid Zaman, and Muheet Ahmed Butt. "An application of logistic model tree (LMT) algorithm to ameliorate Prediction accuracy of meteorological data." International Journal of Advanced Technology and Engineering Exploration 8, no. 84 (2021): 1424.

[25] Altaf, Ifra, Muheet Ahmed Butt, and Majid Zaman. "A Pragmatic Comparison of Supervised Machine Learning Classifiers for Disease Diagnosis." In 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 1515-1520. IEEE, 2021.

[26] Fayaz, Sheikh Amir, Ifra Altaf, Aaqib Nazir Khan, and Zahid Hussain Wani. "A possible solution to grid security issue using authentication: an overview." J. Web Eng. Technol 5, no. 3 (2019): 10-14.

[27] Zaman, Majid, and Muheet Ahmed Butt. "Information translation: a practitioners approach." In World Congress on Engineering and Computer Science (WCECS). 2012.

[28] Ashraf, Mudasir, Syed Mudasir Ahmad, Nazir Ahmad Ganai, Riaz Ahmad Shah, Majid Zaman, Sameer Ahmad Khan, and Aftab Aalam Shah. "Prediction of cardiovascular disease through cutting-edge deep learning technologies: an empirical study based on TENSORFLOW, PYTORCH and KERAS." In International Conference on Innovative Computing and Communications, pp. 239-255. Springer, Singapore, 2021.

[29] Ashraf, Mudasir, Majid Zaman, and Muheet Ahmed. "Performance analysis and different subject combinations: an empirical and analytical discourse of educational data mining." In 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 287-292. IEEE, 2018.

[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: A highly efficient gradient boosting decision tree, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017, URL https://proceedings.neurips.cc/paper/2017/file/ 6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[31] L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush, A. Gulin, Catboost: Unbiased boosting with categorical features, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, in: NIPS'18, Curran Associates Inc., Red Hook, NY, USA, 2018, pp. 6639–6649.

[32] Y. Zhao, G. Chetty, D. Tran, Deep learning with XGBoost for real estate appraisal, in: 2019 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2019, pp. 1396–1401.

[33] S. Ramraj, N. Uzir, R. Sunil, S. Banerjee, Experimenting XGBoost algorithm for prediction and classification of different datasets, Int. J. Control Theory Appl. 9 (2016) 651–662.

[34] D. Dua, C. Graff, UCI machine learning repository, 2017, URL http://archive.ics.uci.edu/ml.

[35] Kaggle, Rossmann store sales, 2019, URL https://www.kaggle.com/c/ rossmannstore-sales.

[36] J. Vanschoren, J.N. Van Rijn, B. Bischl, L. Torgo, OpenML: networked science in machine learning, ACM SIGKDD Explor. Newsl. 15 (2) (2014) 49–60.

[37] T. Qin, T. Liu, Introducing LETOR 4.0 datasets, 2013, CoRR http://arxiv.org/abs/1306.2597.

[38] Pascal, Pascal large scale learning challenge, 2008, URL https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html.

[39] Kaggle, Churn modelling, 2019, URL https://www.kaggle.com/shrutimechlearn/churn-modelling.

[40] IBM, Telco customer churn, 2019, https://community.ibm.com/ community/user/businessanalytics/blogs/steven-macko/2019/07/11/telco-customer-churn-1113.

[41] Fayaz, Sheikh Amir, Majid Zaman, and Muheet Ahmed Butt. "Performance Evaluation of GINI Index and Information Gain Criteria on Geographical Data: An Empirical Study Based on JAVA and Python." In International Conference on Innovative Computing and Communications, pp. 249-265. Springer, Singapore, 2022.

[42] Fayaz, Sheikh Amir, Majid Zaman, and Muheet Ahmed Butt. "Knowledge Discovery in Geographical Sciences—A Systematic Survey of Various Machine Learning Algorithms for Rainfall Prediction." In International Conference on Innovative Computing and Communications, pp. 593-608. Springer, Singapore, 2022.

[43] Sidiq, S. Jahangeer, Majid Zaman, and Muheet Ahmed. "How Machine Learning is Redefining Geographical Science: A Review of Literature." (2019).

[44] S. Arik, T. Pfister, Tabnet: Attentive interpretable tabular learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, (8) 2021, pp. 6679–6687, URL https://ojs.aaai.org/index.php/AAAI /article/view/16826

[45] L. Katzir, G. Elidan, R. El-Yaniv, Net-DNF: Effective deep modeling of tabular data, in: 9th International Conference on Learning Representations, ICLR 2021,Virtual Event, Austria, May 3-7, 2021, OpenReview.net, 2021, URL https://openreview.net/forum?id=73WTGs96kho.

[46] S. Popov, S. Morozov, A. Babenko, Neural oblivious decision ensembles for deep learning on tabular data, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020, URL https://openreview.net/forum?id=r1eiu2VtwH.

[47] Kaul, S., Fayaz, S.A., Zaman, M., Butt, M.A. (2022). Is decision tree obsolete in its original form? A Burning debate. Revue d'Intelligence Artificielle, Vol. 36, No. 1, 105-113. https://doi.org/10.18280/ria. 360112.

[48] Fayaz, S. A., Zaman, M., & Butt, M. A. (2022). Numerical and Experimental Investigation of Meteorological Data Using Adaptive Linear M5 Model Tree for the Prediction of Rainfall. Review of Computer Engineering Research, 9(1), 1-12.