An approach to understand what type of lightly trained models can perform equivalent to heavily trained model

Dhruvil Anil Trivedi
Department of Data Science, Viterbi School of Engineering,
University of Southern California, Los Angeles, California 90089, USA

(Dated: May 1, 2021)

There is abundant amount of data available today but the actual meaningful and useful data is very less. This is what creates the need for lightly trained models to be more efficient and accurate. These models are to be such that they perform equally effectively as heavily trained models. Here, the article proposes an experiment with a basic architecture inspired by the characteristics of various machine learning domains such as Federate Machine Learning, Feature Importance and Ensemble Learning. This approach is tested with a baseline model to determine whether the proposed approach is better than the current approach or not. In the proposed experiment, the new architecture proved to be significantly close for competing with the traditional approach of using neural network for image classification.

Keywords: Machine Learning, Federated Learning, Ensemble Learning, Feature Importance

I. INTRODUCTION

In this article, a new approach is being tested regarding the work currently done in the field of federated learning. In recent times, the federated learning is mainly developed to cope with the problems of personal data security [1, 3, 4]. The custom algorithm is not directly related with the concept of federated machine learning, rather its specific characteristics. These are the characteristics such as in federated learning, there are multiple instances of the model created which are then sent to the individual nodes for training and then the knowledge is used further to get a federated model [1, 2, 5, 6]. The approach presented here is the way to test that instead of using heavily trained models, using lightly trained models efficiently can provide sufficiently good results. Hence, the qualities of a federated learning approach of creating instances of a single model can be used for training separately is going to be used. In addition to that, there are other domains that are also being included in the experiment as they provide certain other features which help the architecture to perform flow control in efficient manner. Firstly, the feature importance will be an important aspect as it will provide the proposed architecture a means of determining the further decisions in the flow structure. Lastly, the characteristics of ensemble learning are also depicted and used to test and attain a more generalised results from the proposed architecture. Ensemble learning is a very deep concept, however, only the motivation behind the concept is going to be considered for the given experiment. All these domains are explained in detain in the II. Literature Review section.

II. LITERATURE REVIEW

The proposed approach is an experiment to understand whether or not the lightly trained models if used appropriately are capable of providing nearly equal results or not. Here, there are certain domains explained in detail from which various characteristics of the new architecture are adapted while designing the work flow.

A. Federated Machine Learning

Federated Machine Learning is a relatively new field of study which primarily is focused on using multiple instances of model and sending those to the data sources for training instead of transferring the actual data from one point to another [1]. This is mainly used to cope with the issues faced for using the data in conventional manner. There are many problems that are described as:

- Privacy concerns: The first and foremost problem is the privacy concern of the data as in current situation, companies are mainly using one's personal data to train their models to make them more user friendly and user specific. However, this brings up concerns regarding the ethical use of the data and maintaining the privacy of the data at all costs [2, 3].

- Data transfer security: Another problem is that the security of the data while transferring it is also a major concern for many companies as the user's personal data is considered to be a gold mine of insights for many domains [1].

- Data transfer cost: This is also a very crucial factor when considered the size of user base across the globe. Transferring this much amount of data and storing it (even temporarily) is very costly even though the price for technology becomes cheaper.

Thus, in federated machine learning, to address these issue, a local model is made as the instance of the main model which is then sent to each end user for training [4, 5]. That knowledge is then transferred to the global model. The local model is initialised every time for different modes considering the current state of the global model thus inducing the model' capacity to remember the learned patterns and trends.

There are many approaches for federated learning that are currently used in the industry. They are the vertical and the horizontal approach. Here, the local model instances are trained separately on different data and then combined either using the horizontal architecture or the vertical architecture for combining the knowledge of those instances to the uniform global model [2, 4].

B. Feature Importance

In this experiment, feature importance will act as the control unit for the decision making process which will be explained in detail in the coming sections. Here, the importance of certain features is taken into account in order to determine the actual driving forces and factors behind certain target variable in the dataset. This can be done by simple correlation between the variables. However, it is not recommended to do so as the correlation functions are only able to catch the linear trends in the data. Though, when the trend pattern is non-linear in nature, the correlations do not depict the perfect behavior of the features.

Thus, for the given experiment, the factor of non-linearity in trend and dependency of target variable over the independent variables is considered important. Hence, instead of linear correlations, the feature importance using SelectKBest algorithm is considered [8]. Here, the library function from sklearn is used which determines as to which of the following features is more important than others and gives scores to all the columns. However, for the given experiment, there is only a small problem. The given dataset is for image data and the Select k best specially works for the tabular data. Hence, flattening the images would be important without losing the structural information of the same.

C. Ensemble Learning

Ensemble learning is a very wide concept and has many research going on in the field. It is mainly described as a means of combining the outputs of two different models in order to get more generalised and reliable output [9, 10]. For the given experiment, the ensemble learning is just the concept, where the two selected lightly trained models will be used in order to make the relevant predictions. There are many approaches available, however, the baseline would be to test the working with simple sum or mean of the prediction probabilities of the classes and determining the best approach. This can also be done for weighted means of the predictions. The ensemble approach helps to create a more reliable model prediction instead of a bias one.

D. Additional points

These are some of the characteristics of various domains in the field of machine learning that are intended to be done in the proposed experiment. The characteristics of federated learning to use small instances of a machine learning model, the properties of ensemble learning to combine the results of 2 models is used and the feature importance acts as the filter in determining the model instance that is supposed to be icked in the proposed architecture.

An additional point to take into account is that, for the SelectKBest, there should be a limited number of features. As the images of the dataset are 28*28 in dimension. When flattened, they have 784 features columns and considering all of them is a tedious task. Hence, PCA (Principal Component Analysis) is applied to reduce that dimensionality to certain number of features to test the proposed architecture effectively.

III. DATA EXPLORATION AND PROCESSING

The dataset that is being used for the proposed experiment is the MNIST dataset for handwritten digit recognition. This consists of 70,000 images of size 28*28 pixels. The images are in greyscale and hence there is no third dimension for the same. Furthermore, the dataset consist of labels for each of the image providing the actual number in the image.

As regards to the preprocessing, the dataset will be flattened and stored in the tabular form for testing the feature importance.

Here, the PCA algorithm (Principal Component Analysis) is applied in order to reduce the dimensionality of the dataset. However, doing so creates a bias at initial stages for testing the importances of all the actual features instead of the orthogonal dimensions provided by the PCA algorithm. Although, this can be considered to test for future segments as for lesser number of dimensions, it can be assumed the the importance classification would be more efficient and the initial loss of information might not affect the predictions very much.

Hence, this dataset is considered for the experiment. Though, a note is that the train and test dataset is exactly same for the baseline model and the new approach in order to test the model to the best possible scenario, keeping maximum number of external influences such as the reduced number of dimensions and neural network model layers to be consistent.

IV. BASELINE MODEL

This small section which describes that what was done to get the best possible model by the normal approaches with only basic hyperparamater tuning. Here, a basic image recognition model is made using Artificial Neural Networks. This model is tested for its performance using the accuracy based on the confusion matrix, F1 score (with 'macro' weighted average for multiple classes prediction), F1 score (with 'micro' weighted average for multiple classes prediction) and the AUC score. This is considered as the baseline model as for the proposed architecture, the structure of the model is exactly the same and the hyperparameters too for efficiently comparing the two performances. There are mainly 2 baseline models describing the behavior of the predictions.

A. Baseline model I

The first baseline model is done using flattening the 28*28 images and predicting the class according to that. As it has an input shape of 784 dimensions, the model parameters are very high due to that.

The parameters for baseline model I: 111,146

As the model is heavier due to input size, it is more likely to give a bit higher accuracy.

B. Baseline model II

The second baseline model is the one where PCA algorithm is applied in order to attain the baseline accuracy for a model trained on all the data with reduced dimensionality. This model had fairly lesser dimensions than the actual data and thus the neural network model parameters are also comparatively lesser than the baseline model I.

The parameters for baseline model II: 16,554

This model was tested for various dimension values ranging from 10 to 50 dimensions. Finally, the number of dimensions were taken to be 45 which provided the best accuracy. Here, only the best model is represented to check if the proposed model actually works as well as expected or not. The model parameters mentioned are according to input shape taken to be 45.

C. Model Architecture

The general model architecture used for all the models is shown in the Figure 1. This represents 3 hidden dense layers after a flattening layer and lastly 1 dense layer with 10 filters to predict 10 different classes.

D. Results

The Table 2 provides the results of the the prediction accuracies for both the baseline models.

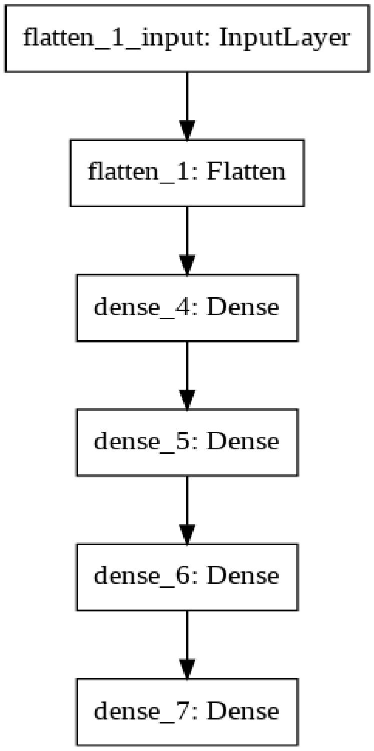
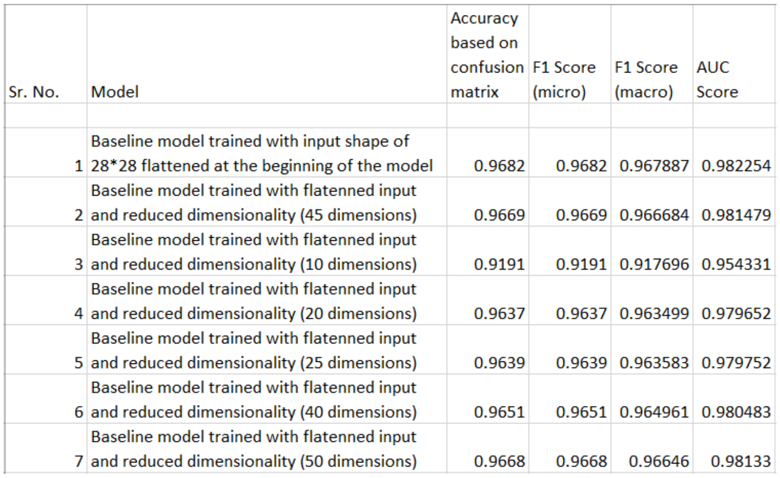


FIG. 1. This is the flowchart to depict the basic model architecture.



Sr. No.	Model	Accuracy based on confusion matrix	F1 Score (micro)	F1 Score (macro)	AUC Score
1	Baseline model trained with input shape of 28*28 flattened at the beginning of the model	0.9682	0.9682	0.967887	0.982254
2	Baseline model trained with flatenned input and reduced dimensionality (45 dimensions)	0.9669	0.9669	0.966684	0.981479
3	Baseline model trained with flatenned input and reduced dimensionality (10 dimensions)	0.9191	0.9191	0.917696	0.954331
4	Baseline model trained with flatenned input and reduced dimensionality (20 dimensions)	0.9637	0.9637	0.963499	0.979652
5	Baseline model trained with flatenned input and reduced dimensionality (25 dimensions)	0.9639	0.9639	0.963583	0.979752
6	Baseline model trained with flatenned input and reduced dimensionality (40 dimensions)	0.9651	0.9651	0.964961	0.980483
7	Baseline model trained with flatenned input and reduced dimensionality (50 dimensions)	0.9668	0.9668	0.96646	0.98133

FIG. 2. This is the accuracy table for both the baseline models.

V. NEW APPROACH

This section explains in detail that what was the approach proposed and its implementations over the same dataset as that of the baseline model. The given FIG 3 shows a simple flowchart of the architecture that is supposed to be implemented. There is be multiple chunks of data taken from the actual dataset and trained over separate instances of the same model architecture. These datasets is also be tested for determining the dominant features and their importance level in the given data chunk. Finally, the test dataset is also be checked for the important features and the list of features matching with the respective data chunk's priority list to the most and to the least is considered. The most important columns number is checks for its position in all the different training datasets. The lists are in descending order of importance and thus the minimum number of index position of the given column in any of the training dataset is the training data that is most bias for the same feature compared to others. Similarly, the maximum index position represents the later part of the importance ranking and thus shows the least bias dataset. Both these models are also tested for accuracy in predictions. Then finally combining the predictions of both the most and the least bias models, a single prediction is generated. That is done by either taking the sum of the prediction probabilities made by each model or taking the average of the prediction probabilities and determining the final class of the predictions based on that. In the end, the predictions of both the individual models and the combined predictions results are compared with that of the baseline model to determine the performance, behavior and its reasons for the same. There are two types of combinations used in the models. The prediction probabilities are summed together for the first type of competition and the second one is done using average of list of probability.

The table 4 represents the final output of all the test datasets with their most bias, least bias and combined model predictions and their accuracy scores. Apart from that, it also shows the accuracy scores for prediction the full test dataset altogether in the most bias, least bias and combined model prediction values to get a general idea for comparison with the baseline models.

VI. DISCUSSION

The baseline model I performs well significantly. To a certain extent, the baseline II model is close to the accuracy score of the model I but beats it in some metrics and not in some. This is due to the difference in the parameters of the models. Though the hidden layers and output layers are same, the input layer has a vast difference in the input dimensions and hence the model parameters change thus a deeper model has a slightly better accuracy metric in some aspects. However, the difference is not significant so it can be deduced that if the model II was similarly large enough, it could perform better. This is not done because to do so, the parameters for the hidden layers must be changes which would question the consistent environment of models to compare all the 3 cases.

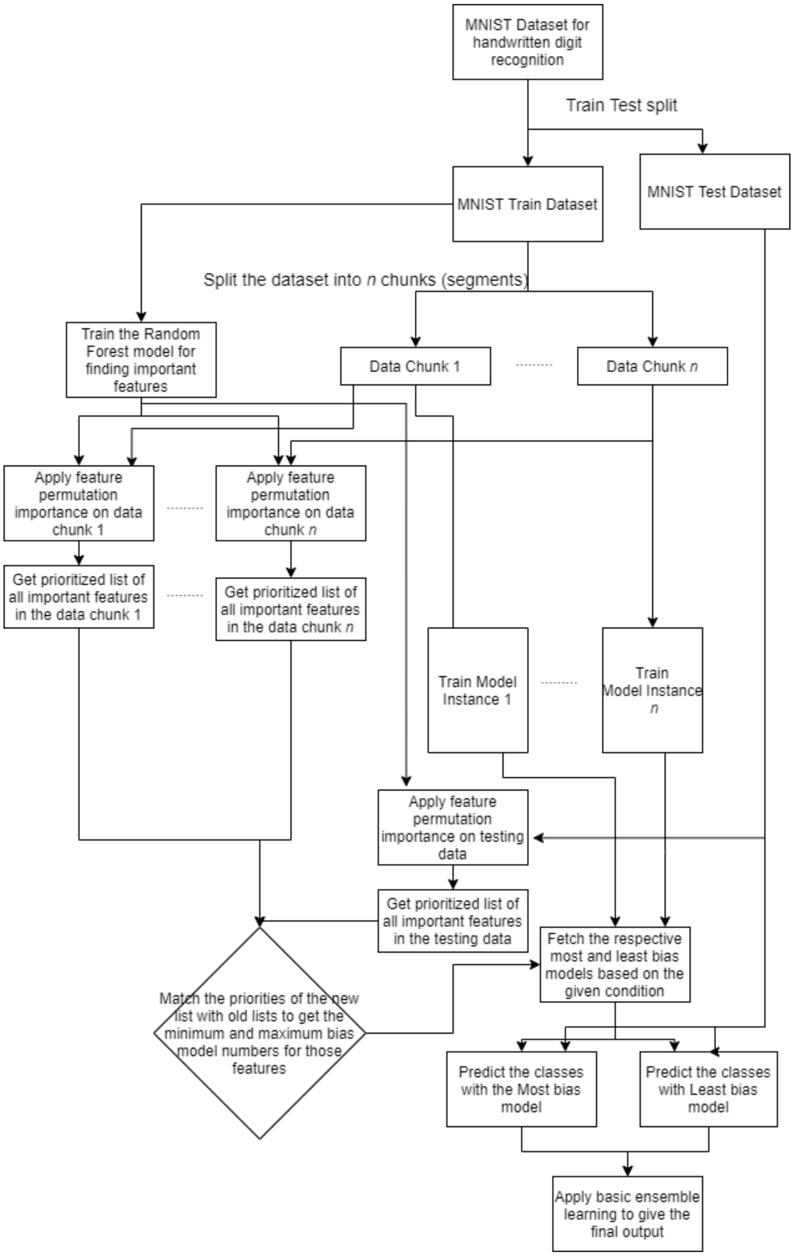
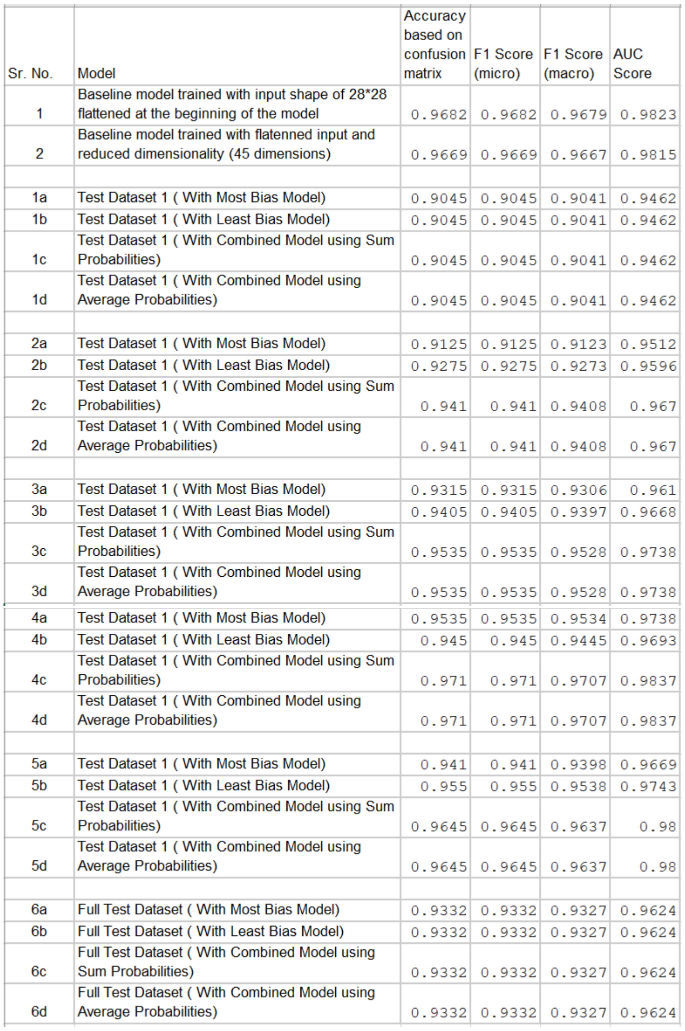

MNIST Dataset for handwritten digit recognition
Train Test split
MNIST Train Dataset
MNIST Test Dataset
Split the dataset into *n* chunks (segments)
Train the Random Forest model for finding important features
Data Chunk 1
....
Data Chunk *n*
Apply feature permutation importance on data chunk 1
....
Apply feature permutation importance on data chunk *n*
Get prioritized list of all important features in the data chunk 1
....
Get prioritized list of all important features in the data chunk *n*
Train Model Instance 1
....
Train Model Instance *n*
Apply feature permutation importance on testing data
Get prioritized list of all important features in the testing data
Match the priorities of the new list with old lists to get the minimum and maximum bias model numbers for those features
Fetch the respective most and least bias models based on the given condition
Predict the classes with the Most bias model
Predict the classes with Least bias model
Apply basic ensemble learning to give the final output

FIG. 3. This is the flowchart to depict the basic outline of the flow of the proposed architecture.

For the proposed model, the test dataset is also split into a few data chunks to get different datasets with different important feature rankings. It is observed that for a few initial datasets, the predictions accuracies are not as good sometimes. However, for the later datasets, the models predict more accurately. For this, the possible explanation is that the later models must have higher precedence in the ranking list for the training datasets of the feature that is important for the given test dataset. Thus, that specific model provides more efficient predictions as it is trained on a data that has similar bias for the specific feature.

To a certain extents, the smaller and lightly trained models are predict almost equally or maybe sometimes even better than the baseline model for example the 4th model gives better accuracy. Some of the accuracy val-



Sr. No.	Model	Accuracy based on confusion matrix	F1 Score (micro)	F1 Score (macro)	AUC Score
1	Baseline model trained with input shape of 28*28 flattened at the beginning of the model	0.9682	0.9682	0.9679	0.9823
2	Baseline model trained with flatenned input and reduced dimensionality (45 dimensions)	0.9669	0.9669	0.9667	0.9815
1a	Test Dataset 1 (With Most Bias Model)	0.9045	0.9045	0.9041	0.9462
1b	Test Dataset 1 (With Least Bias Model)	0.9045	0.9045	0.9041	0.9462
1c	Test Dataset 1 (With Combined Model using Sum Probabilities)	0.9045	0.9045	0.9041	0.9462
1d	Test Dataset 1 (With Combined Model using Average Probabilities)	0.9045	0.9045	0.9041	0.9462
2a	Test Dataset 1 (With Most Bias Model)	0.9125	0.9125	0.9123	0.9512
2b	Test Dataset 1 (With Least Bias Model)	0.9275	0.9275	0.9273	0.9596
2c	Test Dataset 1 (With Combined Model using Sum Probabilities)	0.941	0.941	0.9408	0.967
2d	Test Dataset 1 (With Combined Model using Average Probabilities)	0.941	0.941	0.9408	0.967
3a	Test Dataset 1 (With Most Bias Model)	0.9315	0.9315	0.9306	0.961
3b	Test Dataset 1 (With Least Bias Model)	0.9405	0.9405	0.9397	0.9668
3c	Test Dataset 1 (With Combined Model using Sum Probabilities)	0.9535	0.9535	0.9528	0.9738
3d	Test Dataset 1 (With Combined Model using Average Probabilities)	0.9535	0.9535	0.9528	0.9738
4a	Test Dataset 1 (With Most Bias Model)	0.9535	0.9535	0.9534	0.9738
4b	Test Dataset 1 (With Least Bias Model)	0.945	0.945	0.9445	0.9693
4c	Test Dataset 1 (With Combined Model using Sum Probabilities)	0.971	0.971	0.9707	0.9837
4d	Test Dataset 1 (With Combined Model using Average Probabilities)	0.971	0.971	0.9707	0.9837
5a	Test Dataset 1 (With Most Bias Model)	0.941	0.941	0.9398	0.9669
5b	Test Dataset 1 (With Least Bias Model)	0.955	0.955	0.9538	0.9743
5c	Test Dataset 1 (With Combined Model using Sum Probabilities)	0.9645	0.9645	0.9637	0.98
5d	Test Dataset 1 (With Combined Model using Average Probabilities)	0.9645	0.9645	0.9637	0.98
6a	Full Test Dataset (With Most Bias Model)	0.9332	0.9332	0.9327	0.9624
6b	Full Test Dataset (With Least Bias Model)	0.9332	0.9332	0.9327	0.9624
6c	Full Test Dataset (With Combined Model using Sum Probabilities)	0.9332	0.9332	0.9327	0.9624
6d	Full Test Dataset (With Combined Model using Average Probabilities)	0.9332	0.9332	0.9327	0.9624

FIG. 4. This is the final results table depicting the accuracy results of the both selected baseline models and the proposed model.

ues for combined and in some cases for all the models is similar because the bias of that specific important feature of the test dataset might not have affected the model training very much. However, it can be deduced from the experiment that the models with lesser training instances can be used to provide near to heavily trained prediction model.

Apart from that, another most important result was that though the most and least bias models gave predictions with lesser accuracy, the combination of both the predictions provided a comparatively higher accuracy scores. This represents that rather than using only 1 model, using 2 models that are very bias and very less bias and combining their results could provide better results even better than the simpler basline approach for certain cases where the the training model has a much higher precedence (most bias) or much lower precedence (least bias) of the feature that is important in the test dataset. Apart from that, the final code was run using training datasets with 9600 images and the full testing dataset had 10000 images and still the predictions were close to the baseline models. This fact also supports the claim that using biases in models to our advantage can be an approach that might provide better results.

However, one factor to consider is that for initial test datasets, the prediction accuracy was significantly lower than that of the baseline model. Here, it is difficult to explain the behavior as the model selected for predictions is similar for another test datasets though the accuracies show a vast difference. The reason could be the same that the precedence values of the most important feature might be low in the training datasets. Though, there is no means to be perfectly certain for the same. Furthermore, another small setback is that in a few cases, a certain metric values are lower than baseline but other are higher than that of the baseline models. Here, sometimes, it becomes difficult to analyse that is the proposed approach working well or not.

VII. CONCLUSIONS

In a nut-shell, the experiment proved to be efficient enough to compete with the general approach for the prediction of the images. Though there were a certain behaviors of the model that cannot be explained with proper clarity, the model proved to be accurate enough though trained with significantly lesser data (20% - 40%) as compared to the baseline model. Furthermore, the ensemble learning characteristics of combining the predictions of the most and least bias models proved to be a success. There are cases where certain models have lesser AUC score than the baseline models (0.982254057, 0.981478589). However, there are some models which showed better accuracy scores like the model used for 4th data chunk test dataset(0.983721632186567). Though, all these scores were comparable with the performance of the combined predictions. The bias models by themselves were no match for the baseline model itself. This shows a small proof-of-concept that the bias in the training data, which is reflected in the neural network model as well, can be used to advantage in a proper manner. Also, this proves the idea of the experiment for lightly trained models being able to work as efficiently as the heavily trained models.

VIII. FUTURE SCOPE

The future prospects of this are very vast as this is just a small proof-of-concept of a simple idea. However, if used for other types of machine learning models, this proposed architecture can prove to be a good breakthrough in the perspectives of achieving higher prediction accuracy. In addition to that, there could be a case where such model innstances are trained on larger dataset and then compete with the baseline models with most and least bias models. The idea is very general and thus can be tested on various other aspects as well such as sentiment analysis, classification, regression problems and so on.

DATA AVAILABILITY

Public link to access information on the dataset : https://keras.io/api/datasets/mnist/

CODE AVAILABILITY

Public Github link to access information on the codes : https://github.com/DhruvilATrivedi/DSCI552Project

ACKNOWLEDGMENTS

I would like to acknowledge Dr. Marcin Abram for guiding me through the process and helping me understand the concept and explore in the right direction. Furthermore, I would also like to thank Tirth Patel and Vishal Ajaybhai Kapadia for helping me in getting good ideas and insights and giving me reviews for all my work. Furthermore, I would also thank all the peers who will be reviewing my work and am grateful for the suggestions that they provide to me regarding the research.

IX. REFERENCES

1. Li, T., Sahu, A. K., Talwalkar, A., Smith, V. (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), 50-60.

2. Yang, Q., Liu, Y., Chen, T., Tong, Y. (2019). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), 1- 19.

3. Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., Yu, H. (2019). Federated learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 13(3), 1-207.

4. Mansour, Y., Mohri, M., Ro, J., Suresh, A. T. (2020). Three approaches for personalization with applications to federated learning. arXiv preprint arXiv:2002.10619.

5. Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., Zhou, Y. (2019, November). A hybrid approach to privacy-preserving federated learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security (pp. 1-11).

6. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics (pp. 1273-1282). PMLR.

7. Li, J., Khodak, M., Caldas, S., Talwalkar, A. (2019). Differentially private meta-learning. arXiv preprint arXiv:1909.05830.

8. Feature importance: SelectKBest

9. Dietterich, T. G. (2002). Ensemble learning. The handbook of brain theory and neural networks, 2, 110-125.

10. Zhou, Z. H. (2009). Ensemble learning. Encyclopedia of biometrics, 1, 270-273.