PROJECT SYNOPSIS

Predicting Diseases With Edge Intelligence

Kazaree Basak (504), Rajdeep Das (515), Rajarshi Bhattacharya (537)

M.Sc. Computer Science, Semester 4

Under Guidance of
Dr. Tapalina Bhattasali

St. Xavier's College
[Autonomous], Kolkata

ACKNOWLEDGMENTS

We express our sincere thanks to Rev. Dr. Dominic Savio, S.J, Principal, St. Xavier’s College Kolkata.

We pay our deep sense of gratitude to our guide Dr. Tapalina Bhattasali, who encouraged us to the highest peak and whose valuable guidance and kind supervision has helped us throughout this project and to shape and present the work.

We pay our deep sense of gratitude to Prof. Romit Beed (HOD) of the Computer Science Department for providing us the opportunity to prepare the project.

St. Xavier's College
[Autonomous], Kolkata
2021

Kazaree Basak,
Rajdeep Das,
Rajarshi Bhattacharya

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

OBJECTIVE

Our objectives for this project will be to detect and track the person with any kind of disease, keep patient data records for analysis and decision making using edge intelligence,to apply edge intelligence in surveillance for faster detection.

So our main motive will be to create a generic DeepNet model which can predict any kind of diseases from the dataset provided. Then we will simulate this DeepNet model at the Edge part of the network using a Cloud simulator. The data will be provided locally and the model will predict the result based upon the input provided.

LITERATURE SURVEYS

An Edge device is any piece of hardware that controls data flow at the boundary between two networks. Edge Computing allows computing resources and application services to be distributed along the communication path, via decentralized computing infrastructure.

When edge computing is merged with machine learning, we get edge intelligence. Edge Intelligence deals with supporting the intelligence acquired through data at a local level.

Merits of Edge Intelligence

1. By processing the data closer to the source can greatly reduce latency and they can be quickly analysed and decisions can be made based on that faster.

2. Since more data is processed on local devices rather than sending it back to a central data center, the amount of data at risk at a time is reduced.

3. By combining colocation services with regional edge computing data centers, organizations can expand their edge network reach quickly and cost-effectively.

4. Scalability of edge computing makes it incredibly versatile. Edge data centers allow them to service end users efficiently with little physical distance or latency. This is especially valuable for content providers looking to deliver uninterrupted streaming services.

5. Even if a nearby data center is not available, IoT edge computing devices will continue to operate effectively on their own because they handle vital processing functions natively.

Deep Learning is a collection of algorithms inspired by the workings of the human brain in processing data and creating patterns for use in decision making, which are expanding and improving on the idea of a single model architecture called **Artificial Neural Network.**.

Reasons for using Deep Learning in Edge Intelligence

1. Deep Learning algorithms, having high resource demanding workload naturally suits for edge computing.

2. Low latency and low cost computing- Since DL services are deployed closed to the requesting user and cloud participates when required.

Simulating the Edge Environment

Edge computing is an emerging computing paradigm that aims to overcome the Cloud computing limitations by bringing its applications at the Edge of the network. Thus, reducing both the latency and the Cloud workload and leading to a more scalable network. Nevertheless, in these distributed environments where many devices need to offload their tasks to one another (either to increase their lifetime or to minimize the task completion delay) many issues such as resources management strategies have to be solved. Instead of testing them on a real distributed system, the simulation makes it possible to evaluate the proposed strategies and algorithms in a repeatable, controllable and cost-effective way before their actual deployment.

The simulator which we will use is designed to simulate Cloud and Edge computing environments. It allows to evaluate the performance of resources management strategies in terms of network usage, latency, resources utilization, energy consumption, etc. and enables the simulation of several scenarios such as the Internet of Things (IoT),Edge Computing environments (peer-to peer networks such as mobile devices Cloud), and mobile Edge computing.It grantees high scalability by enabling the simulation of thousands of devices. Besides, it supports the Edge devices heterogeneity (i.e. whether this device is mobile or not, whether battery-powered or not, different applications requirements: tasks file size, tasks CPU utilization,and latency requirement, etc.)

HARDWARE & SOFTWARE SPECIFICATION

Software - Python including Tensorflow, matplotlib, pandas, numpy, sklearn packages and PureEdge Simulator.

Hardware - Laptop/Desktop.

SYSTEM DESIGN

RESEARCH OBJECTIVES

We will be using the main factors like glucose level,blood pressure,age,insulin etc. for detecting diabetes disease and we will be using the main factors like age,gender,cholesterol,blood pressure,etc. for detecting heart disease. To train the Neural Network we will use these two datasets.

Deploying the model in Edge and Cloud will be in such a way that the actual computation will take place at the Edge and it will load the necessary data from the cloud and return the result to the IOT.

Here, the IOT devices will be used to collect the data from the patient to be evaluated. Then, the data will get delivered to the edge situated somewhere near the patient's IOT device and then the data stored in the cloud for the DeepNet model will be delivered to the Edge device. Then, Edge will process the patient's data according to the data provided by the cloud and at last, the result will be delivered to the patient delivering the predicted results.

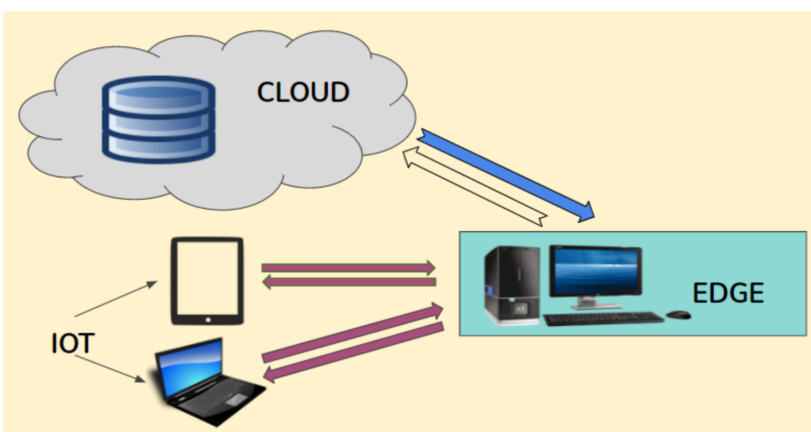

Figure 1 - DeepNet model design

PROPOSED METHODOLOGY

The proposed methodology consists of two phases which is DeepNet Model and the Edge Simulation Phase.

DeepNet Model

The neurons in input and the hidden layers are considered as per the features in the following dataset as the accuracy of the model depends on the number of features provided and as well as the number of neurons being considered. For example, for the Diabetes dataset we have used the following,

```
dlmod=dn.DeepNet(8, 1,'adam','binary_crossentropy',[12,8],"Diabetes")
```

We have considered 80 percent of the dataset as training dataset and rest as test dataset.

We have considered the epoch as 250 and batch size as 10.

Epoch means one pass through all of the rows in the training dataset.

Batch Size means one or more samples considered by the model before weights are updated.

For optimizer we have taken 'adam' and loss function as 'binary_crossentrophy'.

The reason to choose 'adam' as an optimizer as this is a popular version of gradient descent because it automatically tunes itself and gives good results in a wide range of problems. and for loss function 'binary_crossentrophy' is selected for its binary classification as the target values are in binary 0 and 1 with crossentrophy as loss argument.

Edge Simulation

The simulation in PureEdgeSim is based on a area of 2KM*2KM. The main properties are set as,

min_number_of_edge_devices=100

max_number_of_edge_devices=500

edge_device_counter_size=100

wlan_bandwidth=300

orchestration_architectures=EDGE_ONLY

orchestration_algorithms=ROUND_ROBIN

Applications_CPU_allocation_policy=SPACE_SHARED

Etc.

For 100 edge devices the task success rate is around 96% but for 500 the task success rate exceeds 97%, which indicates by increasing the number of edge devices, task success rate can be higher. These can be improved further by introducing Mist concept in the architecture.

During the whole simulation, the average delay is about 1.5s and the average waiting time is 70s.

The average bandwidth gets lower by increasing the number of edge devices. It starts from about 20Mbps, when the number of edge devices is 100 and decreases to almost 6Mbps when the number of edge devices is 6Mbps.

ANALYSIS

The Results we have got using the DeepNet model using Heart Dataset is approx. 80%, Diabetes Dataset is approx 75% and Corona Dataset is approx 90%

Test Results for Diabetes

==========================

5/5 [==============================] - 0s 1ms/step - loss: 0.5352 - accuracy: 0.7632

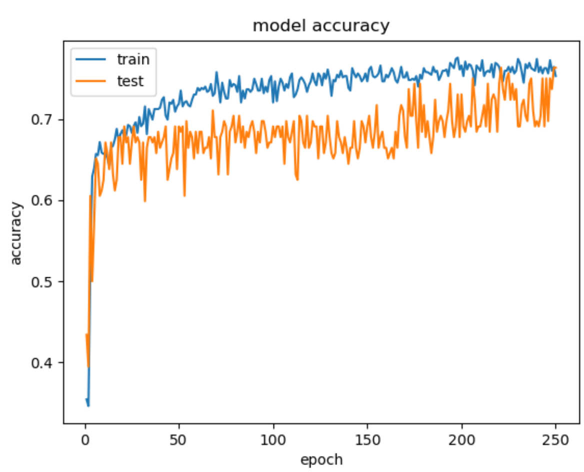

Figure 1 - DeepNet Model Accuracy for Diabetes

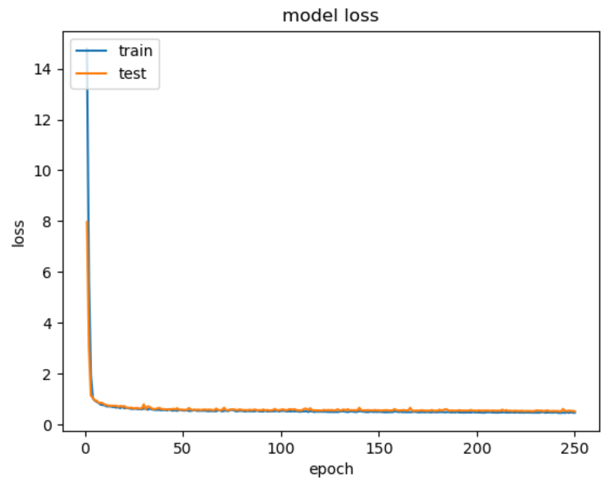

Figure 2 - DeepNet Model Loss for Diabetes

Test Results for Heart

==========================

2/2 [==============================] - 0s 0s/step - loss: 0.5233 - accuracy: 0.8167

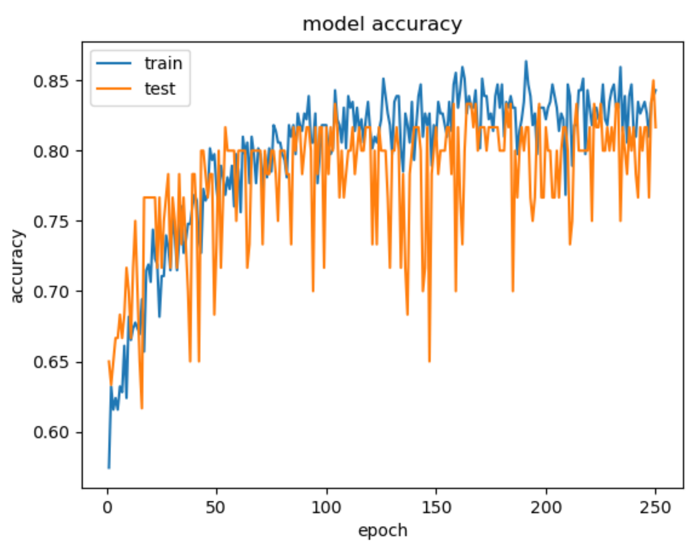

Figure 3 - DeepNet Model Accuracy for Heart

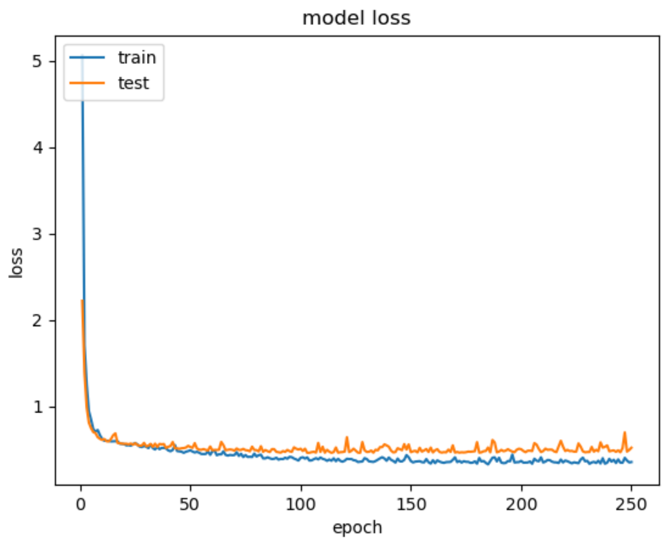

Figure 4 - DeepNet Model Loss for Heart

For 100 edge device simulation

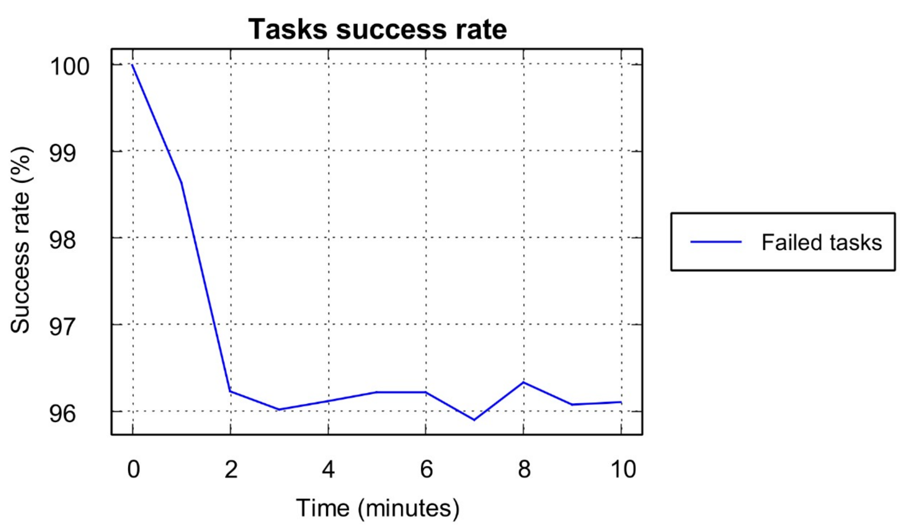

Figure 6 - Graph for task success rate of 100 edge devices

Comparisons between different Machine Learning and DeepNet model

Our model aims at predicting whether a person is affected by a disease or not using Edge Intelligence with minimum latency. We are using a labeled dataset consisting of various symptoms of diseases. Dataset is divided into subsets of training dataset and testing dataset. Using training dataset, our model finds patterns and using that knowledge history it finds outcomes from test dataset.

We have tried various machine learning techniques that fit for binary classification and compare among themselves to find the best fit to the dataset. For building each suitable model with optimal hyper parameters setting, grid search is performed within k-fold cross validation. We have calculated accuracy, precision, recall and latency for each techniques where,
Accuracy = no. of correct prediction/ total no of examples
Precision = predicted as affected that is actually affected / total predicted as affected
Recall = predicted as affected that actually affected / total that actually affected

Logistic Regression (LR) models can be trained very fast and perform well with limited data. It's precision on the corona dataset is 0.99 but decreases to 0.83 when the number of observations increases. It's accuracy, when working with heart dataset with 13 features, is 0.83 and is 0.779 when working with diabetes consisting of 8 features.

SVM performs well with limited data. It's precision on the corona dataset is 0.99 but decreases to 0.82 when the number of observations increases. But the prediction speed of SVM is slow compared to Logistic regression. For example, when working with 1000 observations from a corona dataset, prediction speed for 200 data, in case of SVM it is 0.002s and in case of LR it is 0.00026s.

Random Forest model performs with 0.76 accuracy in case of diabetes dataset and 0.869 for heart dataset. For the whole dataset of corona it performs with accuracy 0.76 and predicts the test set within 0.03s.

DeepNet model takes much time to train and performs well if a huge amount of data with more features is fed to it but it has low performance with limited data. It predicts the testing data for the corona model within 0.57 secs where the size of the dataset is near about 1.5 lacs with 0.91 accuracy also, it predicts the results for the Heart dataset in 0.060 secs for the testing data and in 0.125secs for the Diabetes dataset's testing data.

Performance of machine learning algorithms deteriorates with increasing size of datasets,while deep learning algorithms can handle large datasets well. Our deepnet model performs well when the number of features in the dataset increases.

INNOVATIVENESS & USEFULNESS

This project aims to generate the idea of a generic deep learning model which can predict any disease based upon the dataset provided. The further usefulness that we did incorporate is that the whole computation is done within the edge which can be more faster and effective than the traditional cloud computing based systems.

SUMMARY OF CONTRIBUTION

Team

Project Group: 11

Project Guide: Dr. Tapalina Bhattasali

Name of the Team Member	**Role**	**Tasks and Responsibilities**
Kazaree Basak (504)	**Coder, tester**	**Coding, testing, requirements gathering and analysis, test reports, implementation**
Rajdeep Das (515)	**Data collector, quality tester**	**Problem definition, data collection,system design, quality assurance, planning, documentation**
Rajarshi Bhattacharya (537)	**Coder, documentation**	**Documentation, coding, requirements gathering and analysis, implementation, planning and integration, editor**

REFERENCES

[1] Zhi zhou , Xu Chen , En Li, Liekang Zeng, Ke Luo, and Junshan Zhang, Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing, 2019

[2] Xiaofei Wang, Yiwen Han, Victor C.M. Leung,Dusit Niyato, Xueqiang Yan, Xu Chen, Convergence of Edge Computing and Deep Learning: A Comprehensive Survey, 2020

[3] Quoc-Viet Pham, Dinh C. Nguyen, Thien Huynh-The, Won-Joo Hwang, and Pubudu N. Pathirana, Artificial Intelligence (AI) and Big Data for Coronavirus (COVID-19) Pandemic: A Survey on the State-of-the-Arts, 2020

[4] Wu T, Ge X, Yu G, Hu E, Open-source analytics tools for studying the covid-19 coronavirus outbreak. medRxiv,2020

[5] Mechalikh, C., Taktak, H., Moussa, F.: PureEdgeSim: A Simulation Toolkit for Performance Evaluation of Cloud, Fog, and Pure Edge Computing Environments. The 2019 International Conference on High Performance Computing & Simulation,700-707, 2019

[6] Mechalikh, C., Taktak, H., & Moussa, F. , A Fuzzy Decision Tree Based Tasks Orchestration Algorithm for Edge Computing Environments. In International Conference on Advanced Information Networking and Applications, Springer, Cham, (pp. 193-203),April,2020

[7] Miotto, R., Li, L., Kidd, B.A., and Dudley, J.T. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. Nature Scientific Reports, 6: 26094, 2016

[8] Miotto, R., Li, L. and Dudley, J.T. Deep Learning to Predict Patient Future Diseases from the Electronic Health Records. In the Proceedings of the European Conference on Information Retrieval (ECIR). Springer International Publishing, pp. 768 – 774, 2016

[9] Cisco, Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are, 2015. [Abstract]. Available: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf. [Accessed November 18, 2020]