BTP Seminar

Edge Intelligence in IIoT

Vaishnavi Munghate(18CS30045)

Supervised by :
Prof. Sudip Misra

Mentored by :
Dr. Nurzaman Ahmed

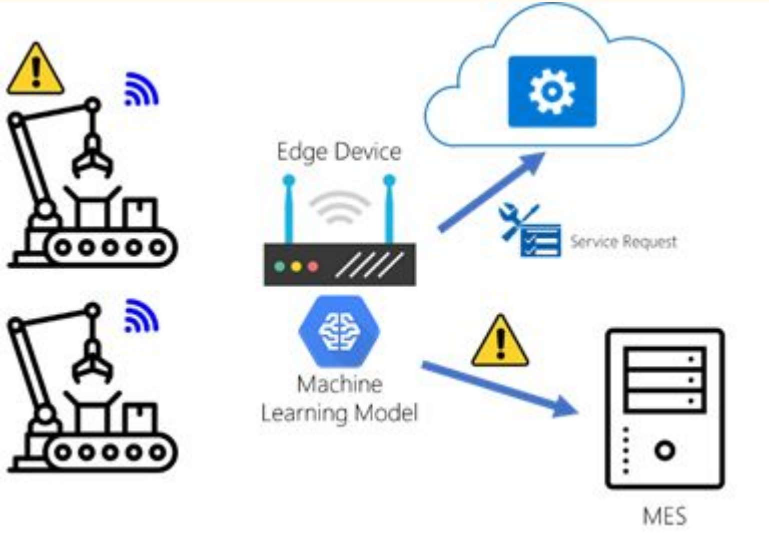
Contents

- Introduction
- Understanding the topic
- Proposed scheme
- Architecture
- Minimal implementation
- Conclusion

INTRODUCTION

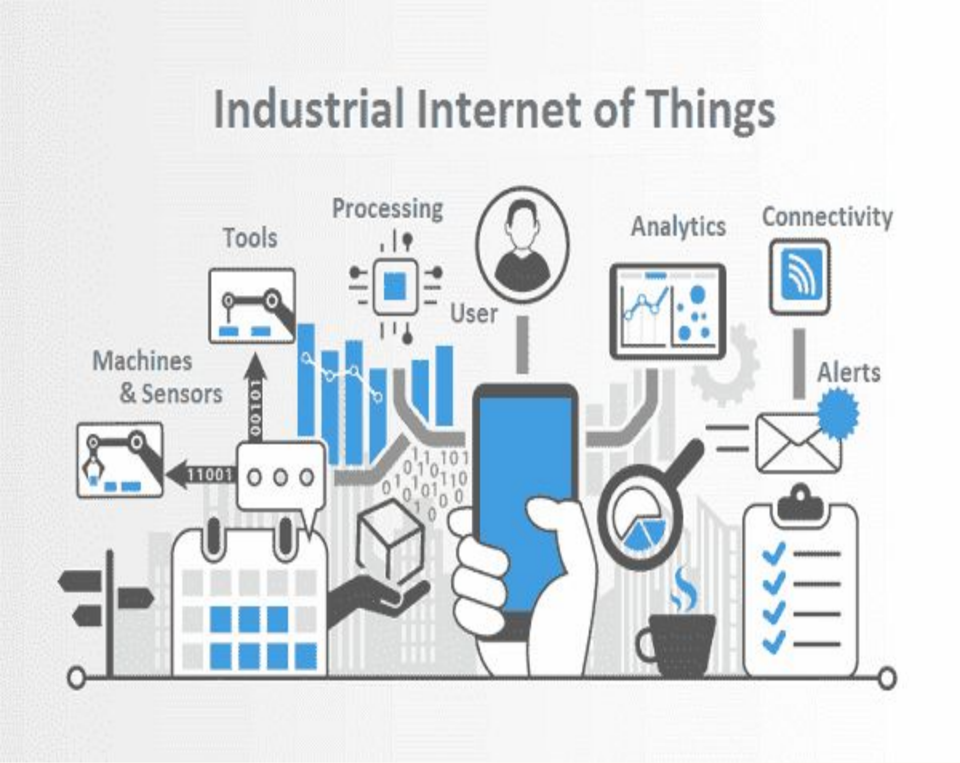
- The project aims to provide an edge intelligent scheme in Industrial IoT.
- We would understand the title of the project and its applications.
- The need of the scheme is as mostly edge computing is used but use of edge intelligence is quite less for now.
- We provided a probable solution.
- A solution is proposed with a minimal implementation.
- The quality of dataset provided also affects quality of the solution.

Edge Intelligence in IIoT ?



When Edge computing is combined with machine learning intelligent algorithms it is called Edge Intelligence. Edge computing is analysis of data at the edge. Use of edge intelligence for industrial purpose with the help of IoT devices is called as edge intelligence in IIoT.

Applications



Smart Industry is an example of Industrial IoT.

PROPOSED SCHEME

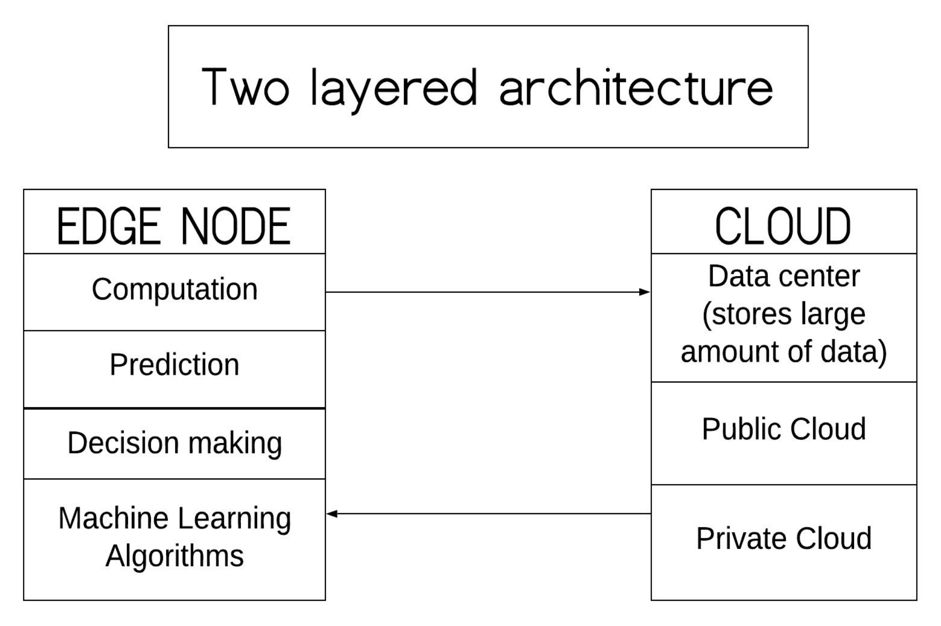
- We propose a reconfigurable edge network for industrial IoT.
- The cloud provides the required data to the edge server and the edge runs a machine learning model to train the data and use it for prediction purpose.
- The edge takes action based on the real time sensor data from the sensors and do the predictions.

ARCHITECTURE

- The architecture consists of basically two layers, one is the cloud and second is the edge device/server.
- The cloud: The cloud is also called as the data centre or data warehouse as it stores massive amount of data. It provides the required data to the edge server.
- The edge: The edge server will receive the data, train and model it and use it for predictions. The edge takes action based on the real time sensor data from the sensors and do the predictions.

ADVANTAGES

- Increase in performance.
- Decrease in latency.
- Availability
- Data Security



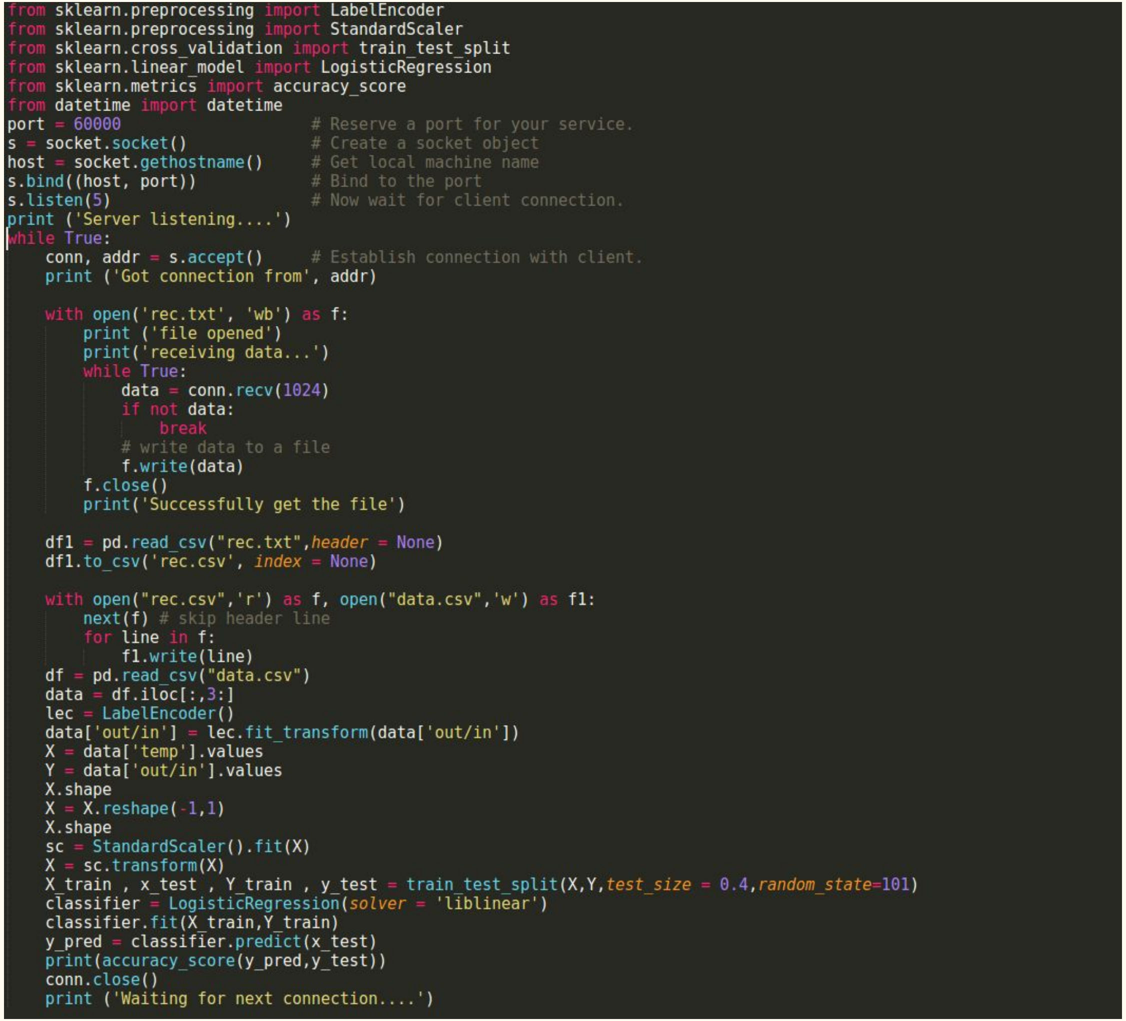
Minimal Implementation

We considered a temperature dataset inside and outside of a room.

Our implementation predicts whether the temperature is of either inside or outside of the room.

The given code snippet is from server.py file which is equivalent to the edge server according to our proposed scheme.

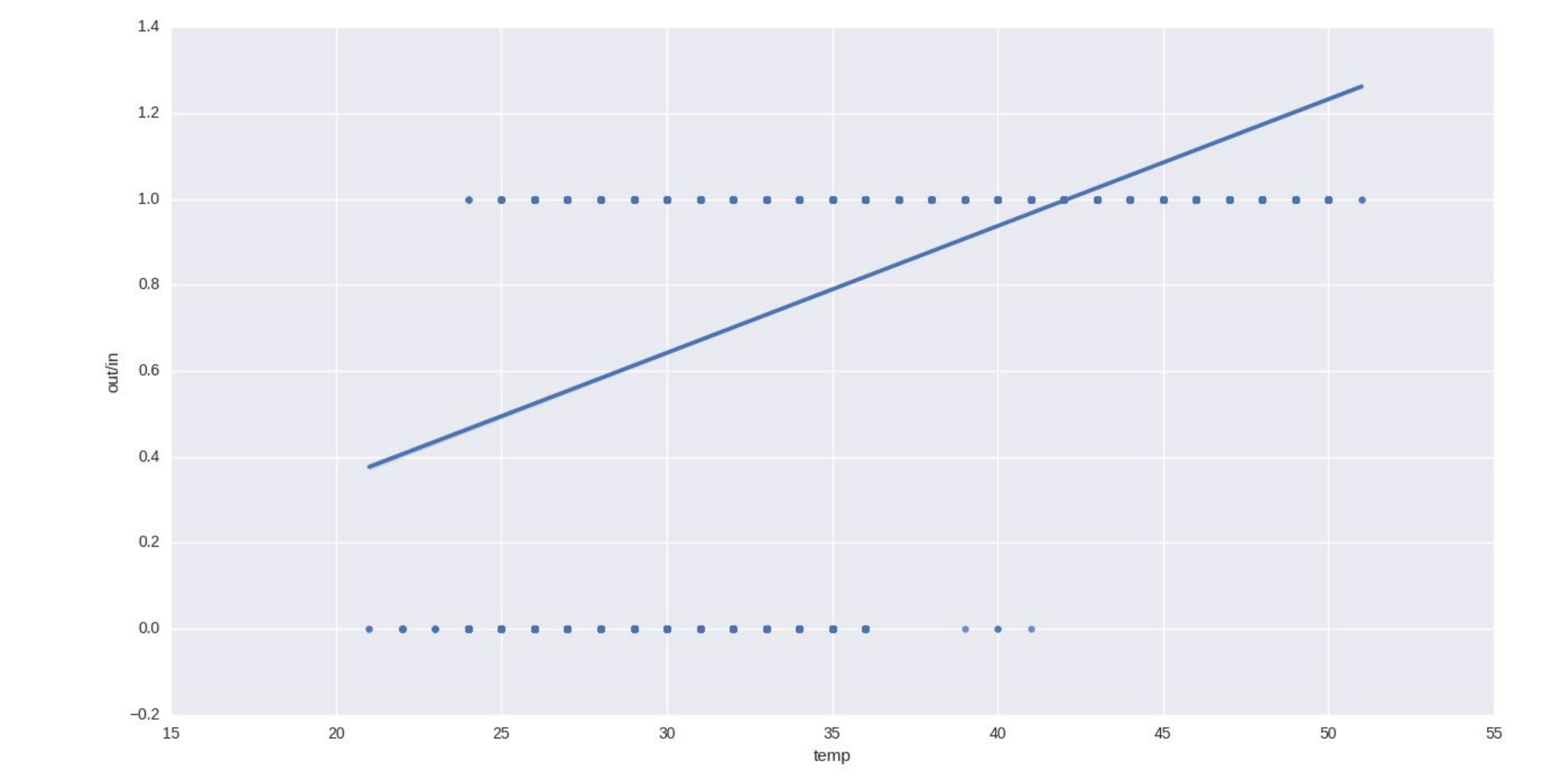
We used TCP network as it is more reliable.



```
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.cross_validation import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from datetime import datetime
port = 60000                    # Reserve a port for your service.
s = socket.socket()             # Create a socket object
host = socket.gethostname()     # Get local machine name
s.bind((host, port))            # Bind to the port
s.listen(5)                     # Now wait for client connection.
print ('Server listening....')
while True:
    conn, addr = s.accept()     # Establish connection with client.
    print ('Got connection from', addr)

    with open('rec.txt', 'wb') as f:
        print ('file opened')
        print('receiving data...')
        while True:
            data = conn.recv(1024)
            if not data:
                break
            # write data to a file
            f.write(data)
        f.close()
        print('Successfully get the file')

    df1 = pd.read_csv("rec.txt",header = None)
    df1.to_csv('rec.csv', index = None)

    with open("rec.csv",'r') as f, open("data.csv",'w') as f1:
        next(f) # skip header line
        for line in f:
            f1.write(line)
    df = pd.read_csv("data.csv")
    data = df.iloc[:,3:]
    lec = LabelEncoder()
    data['out/in'] = lec.fit_transform(data['out/in'])
    X = data['temp'].values
    Y = data['out/in'].values
    X.shape
    X = X.reshape(-1,1)
    X.shape
    sc = StandardScaler().fit(X)
    X = sc.transform(X)
    X_train , x_test , Y_train , y_test = train_test_split(X,Y,test_size = 0.4,random_state=101)
    classifier = LogisticRegression(solver = 'liblinear')
    classifier.fit(X_train,Y_train)
    y_pred = classifier.predict(x_test)
    print(accuracy_score(y_pred,y_test))
    conn.close()
    print ('Waiting for next connection....')
```

Explanation

- We predicted whether the data provided is the temperature of inside the room or outside the room.
- We used socket programming to send and receive data. We used TCP network.
- We took two parameters for the prediction. We took the temperature and set the in/out value to 1 or 0. We then used classification machine learning algorithm that is logistic regression.
- We modelled and trained the data and finally found the accuracy.



line plot of 'temp' vs 'in/out'

CONCLUSION

- According to the dataset that we used and the algorithm used to train the model (logistic regression) we got an accuracy of 0.77(approx.) in our prediction.
- As same IP address was used for server and client in our implementation the latency was found to be 0.620s(maximum RTT).
- As TCP connection is used data loss is minimized.

Thank You

Vaishnavi Munghate
18CS30045