PROJECT REPORT

CRUDE OIL PRICE PREDICTION

submittedby

PNT2022TMID13358

RAJAGURU G	– 951919CS077
SIVA S	– 951919CS092
PRASANTH S	– 951919CS068
SEBASTIAN PRABU M	– 951919CS087

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

a.PROJECT OVERVIEW

Machine learning and deep learning play an important role in computer technology and artificial intelligence. With the use of deep learning and machinelearning, human effort can be reduced in recognizing, learning, predictions and inmany more areas.

Oil is one of the most important and valuable natural resources in the world economy. According to the Energy Information Administration (EIA), the world currently consumes 85.64 million barrels of crude oil daily. That is about 2 liters of oil for every single person on the planet every day. Therefore, it has been sometimes called “black gold” or “life blood”. Most countries heavily relied on imported crude oil in order to meet their energy needs.

b.PURPOSE

Changes in world crude oil prices are becoming an increasing source of concern for government’s economic and organizational decisions. Knowing that every economic sector in the world is dependent on crude oil. Hence any increase or decrease in the price of crude oil has a ripple effect on the global economy Researchers provided some possible evidences of this effect.

CHAPTER 2

LITERATURE SURVEY

a.EXISTING PROBLEM

Content-based filtering is one populartechnique of recommendation or recommender systems. The content or attributes of the things you like are referred to as "content."

Here, the system uses your features and likes in order to recommend you with things that you might like. It uses the information provided by you over the internet andthe ones they are able to gather and then they curate recommendations according to that.

The goal behind content-based filtering is to classify product swith specific keywords, learn what the customer likes, look up those terms in the database, and then recommend similar things.

b.REFERENCES

[1] F. Shen, J. Chao, J. Zhao, Forecasting exchange rate using deep belief networks and conjugate gradient method, Neurocomputing 167 (2015) 243 – 253.

[2] R. Gupta, M. Wohar, Forecasting oil and stock returns with a qual var using over 150 years off data, Energy Economics 62 (2017) 181–186.

[3] D. M. Zhu, W. K. Ching, R. J. Elliott, T. K. Siu, L. M. Zhang, Hidden markov models with threshold effects and their applications to oil price forecasting, Journal of Industrial and Management Optimization 13 (2) (2017) 757–773.

[4] J. Wang, J. Wang, Forecasting energy market indices with recurrent neural networks: Case study of crude oil price fluctuations, Energy 102 (2016) 365–374.

[5] H. Chiroma, S. Abdulkareem, T. Herawan, Evolutionary neural network model for west texas intermediate crude oil price prediction, Applied Energy 142 (2015) 266 – 273.

[6] H. Chiroma, S. Abdul-kareem, A. S. M. Noor, A. I. Abubakar, N. S. Safa, L. Shuib, M. F. Hamza, A. Y. Gital, T. Herawan, A review on artificial intelligence methodologies for the forecasting of crude oil price, Intelligent Automation and Soft Computing 22 (3) (2016) 449–462.

[7] Z. H. Ling, L. Deng, D. Yu, Modeling spectral envelopes using restricted boltzmann machines and deep belief networks for statistical parametric speech synthesis, IEEE Transactions on Audio, Speech, and Language Processing 21 (10) (2013) 2129–2139.

[8] J. B. Heaton, N. G. Polson, J. H. Witte, Deep learning for finance: deep portfolios, Applied Stochastic Models in Business and Industry 33 (1) (2017) 3–12.

[9] M. H. Rafiei, H. Adeli, A novel machine learning model for estimation of sale prices of real estate units, Journal of Construction Engineering and Management 142 (2) (2016) 04015066. [10] H. Y. Zeng, M. D. Edwards, G. Liu, D. K. Gifford, Convolutional neural network architectures for predicting dna-protein binding, Bioinformatics 32 (12) (2016) 121–127.

[11] A. Dedinec, S. Filiposka, A. Dedinec, L. Kocarev, Deep belief network based electricity load forecasting: An analysis of macedonian case, Energy 115, Part 3 (2016) 1688 – 1700, sustainable Development of Energy, Water and Environment Systems.

[12] G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504.

[13] Y. Bengio, Learning Deep Architectures for AI, Now Publishers, 2009.

[14] H. Lee, R. Grosse, R. Ranganath, A. Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: International Conference on Machine Learning, 2009, pp. 609 – 616.

[15] G. E. Hinton, S. Osindero, Y. W. Teh, A fast learning algorithm for deep belief nets, Neural Computation 18 (7) (2014) 1527–1554.

[16] I. Arel, D. C. Rose, T. P. Karnowski, Deep machine learning - a new frontier in artificial intelligence research [research frontier], IEEE Computational Intelligence Magazine 5 (4) (2010) 13–18.

[17] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735.

[18] F. A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with lstm. neural computation 12(10): 2451-2471, Neural Computation 12 (10) (2000) 2451–2471.

PROBLEM STATEMENT DEFINITION

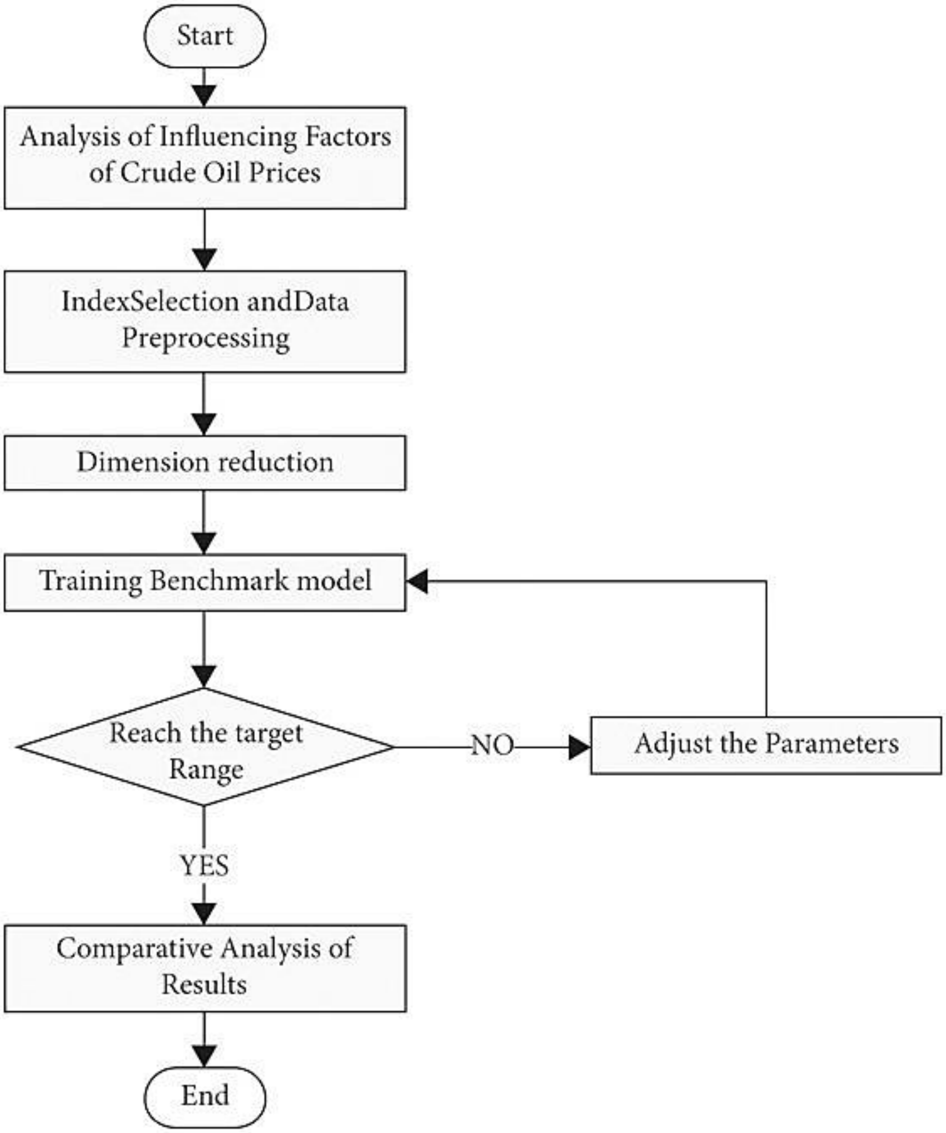
For years, the traffic department has been combating traffic law violators.

These offenders endanger not only their own lives, but also the lives of other individuals. Punishing these offenders is critical to ensuring that others do not become like them. Identification of these offendersis next to impossible becauseit is impossible for the average individual to write down the licenseplate of a recklessdriver.

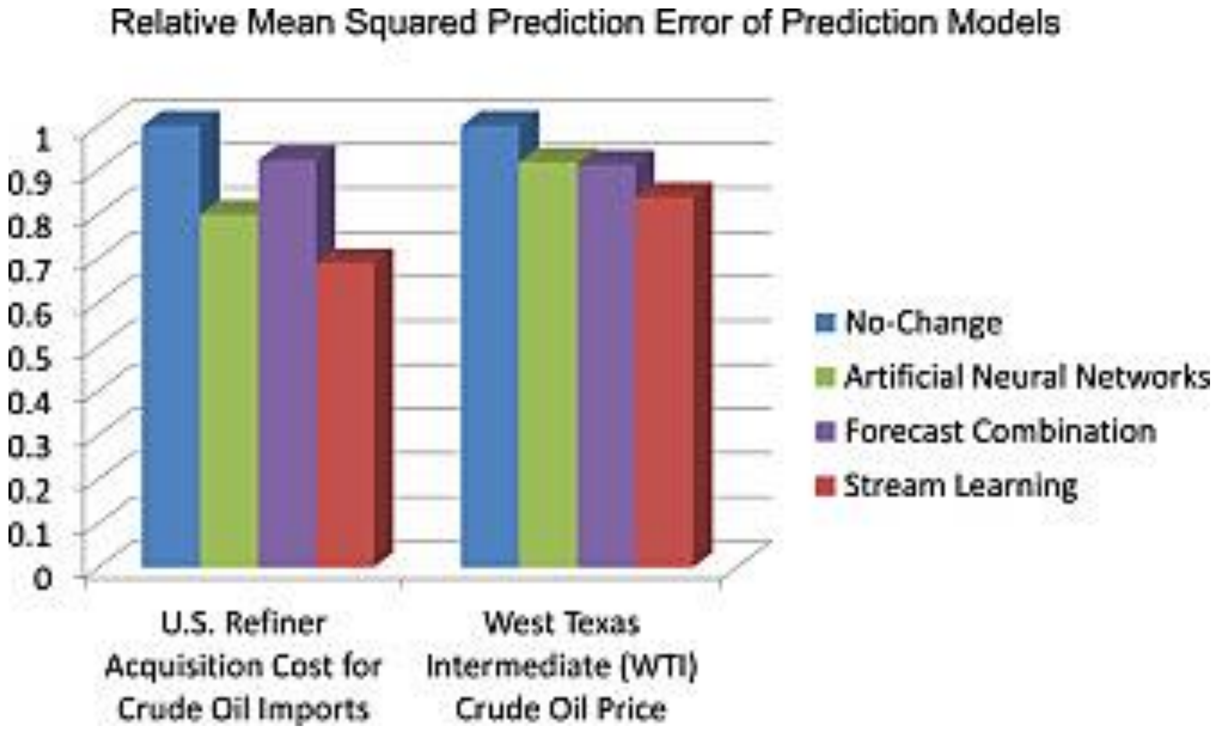
CHAPTER 3

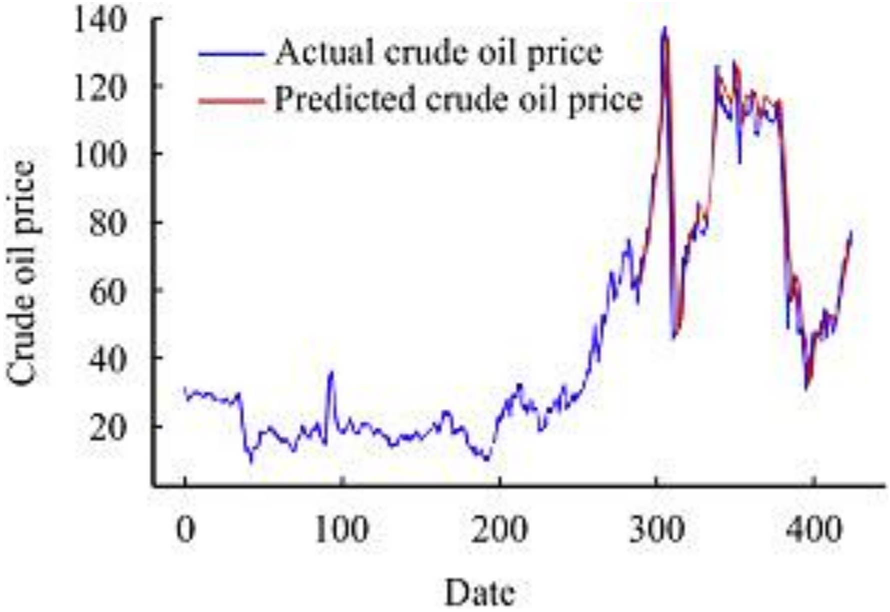
IDEATION AND PROPOSED SOLUTION

a. EMPATHY MAP CANVAS


Start
Analysis of Influencing Factors of Crude Oil Prices
IndexSelection andData Preprocessing
Dimension reduction
Training Benchmark model
Reach the target Range
NO
Adjust the Parameters
YES
Comparative Analysis of Results
End

a. IDEATION & BRAINSTORMING


Relative Mean Squared Prediction Error of Prediction Models
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
U.S. Refiner Acquisition Cost for Crude Oil Imports
West Texas Intermediate (WTI) Crude Oil Price
No-Change
Artificial Neural Networks
Forecast Combination
Stream Learning

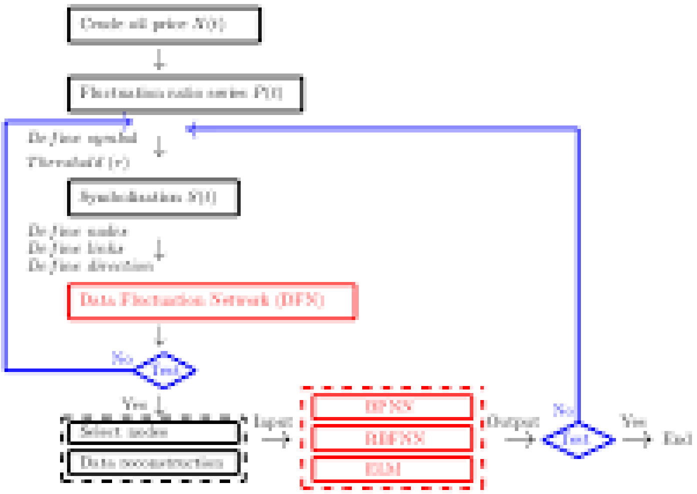

140
120
100
80
60
40
20
0
100
200
300
400
Crude oil price
Date
Actual crude oil price
Predicted crude oil price

c.PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem tobe solved)	Changes in world crude oil prices are becoming an increasing source of concern for government's economic and organizational decisions. Knowing that every economic sector in the world is dependent on crude oil.
2.	Idea / Solution description	Developing an AI predictive model to predict the crude oil price prediction.

3.	Novelty / Uniqueness	The system not only produces aclassification of the digit but also a rich description of the instantiation parameters which can yieldinformation such as oil value
4.	Social Impact/ CustomerSatisfaction	A good prediction tool is crucial to be developed and enterprises can accurately predict the fluctuation of oil price, it will sharply reduce the cost risk and maintain the stable growth of enterprise profits.
5.	Business Model (RevenueModel)	It is used in the detection of vehicle numbers, banks for reading cheques, post offices for arranging oil price, and many other tasks.

d.PROBLEM SOLUTION FIT



CHAPTER 4

REQUIREMENT ANALYSIS

a. FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration	Registration through FormRegistration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Upload image	Image upload via files Image upload via foldersImage upload via drive Image upload via web Image upload viascan/camera
FR-4	Spelling support	Identifies handwriting of different stylesand fonts Spelling check
FR-5	Translation	Handwritten digitsfrom the imageare extracted. Conversion of handwritten digits into

		machine readable form
FR-6	Log out	Log out / sign out.

b.NON FUNCTIONAL REQUIREMENTS

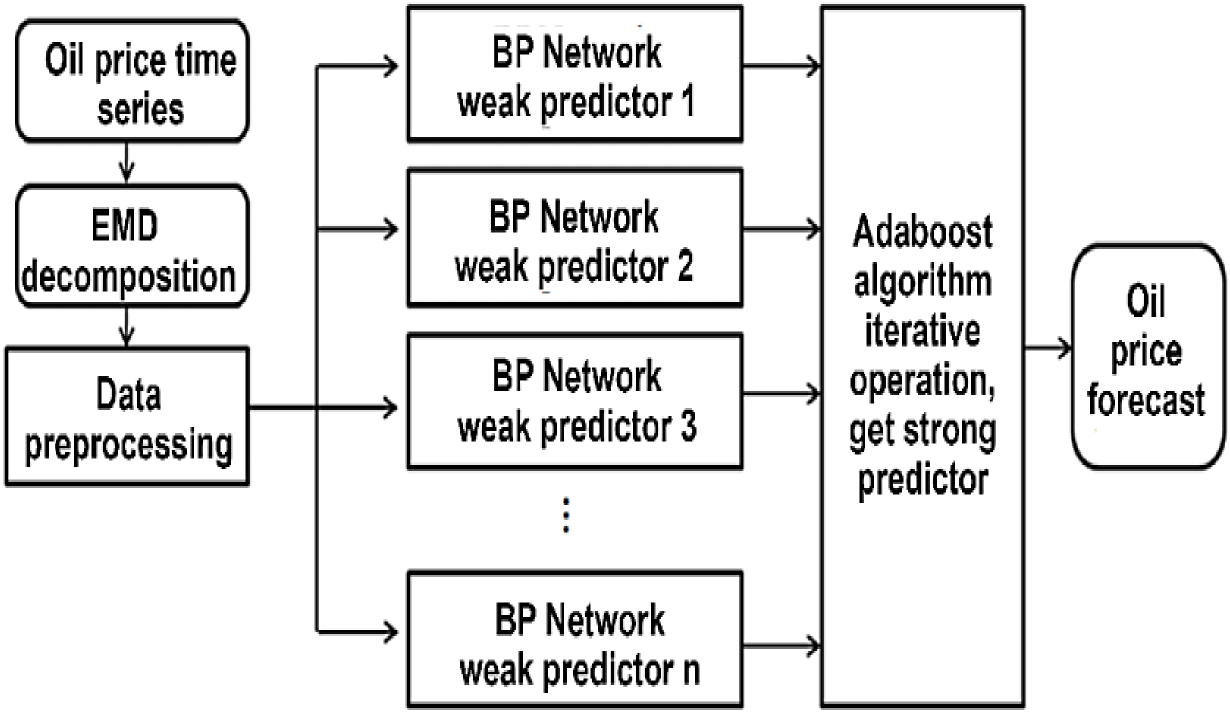
NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	The proposed system gives good results for images that contain oil price different days, different month.
NFR-2	Security	Only authorized people canaccess the system dataand modify the database.
NFR-3	Reliability	The Database is frequently updated with handwriting of different styles and sizeand will rollback when any update fails.
NFR-4	Performance	The proposed system is advantageous as it uses fewer features to train the neural network, which results infaster convergence.

NFR-5	Availability	The system functionality andservices are available for use with all operations.
NFR-6	Scalability	The website traffic limitmust bescalable enough to support2 lakhs usersat a time

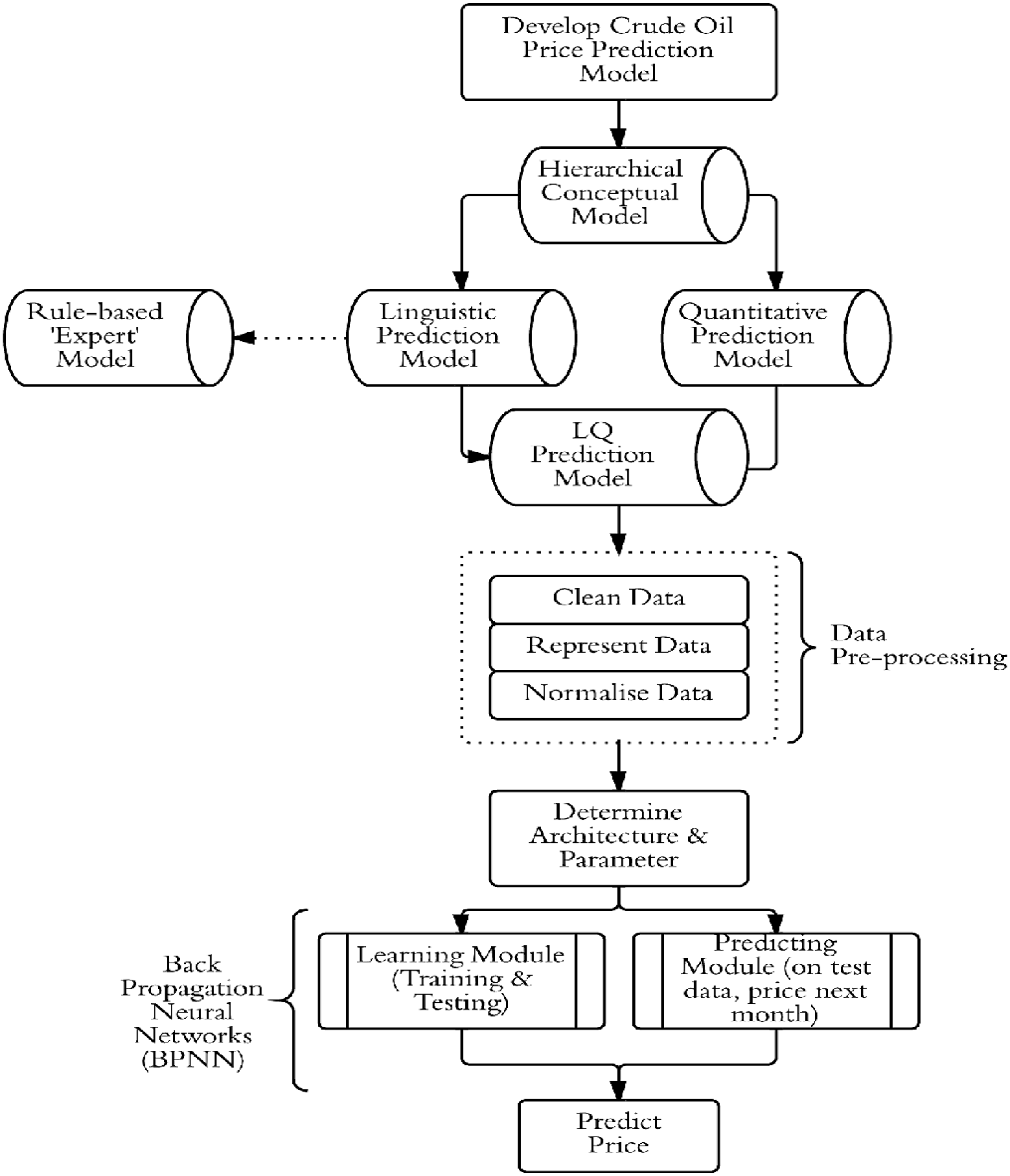
CHAPTER 5

PROJECT DESIGN

a. DATA FLOW DIAGRAM


Oil price time series
EMD decomposition
Data preprocessing
BP Network weak predictor 1
BP Network weak predictor 2
BP Network weak predictor 3
BP Network weak predictor n
Adaboost algorithm iterative operation, get strong predictor
Oil price forecast

b.SOLUTION & TECHNICAL ARCHITECTURE


Develop Crude Oil Price Prediction Model
Hierarchical Conceptual Model
Rule-based 'Expert' Model
Linguistic Prediction Model
Quantitative Prediction Model
LQ Prediction Model
Clean Data
Represent Data
Normalise Data
Data Pre-processing
Determine Architecture & Parameter
Back Propagation Neural Networks (BPNN)
Learning Module (Training & Testing)
Predicting Module (on test data, price next month)
Predict Price

c.USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobileuser)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint- 1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-2

		USN-3	As a user, I can register for the application throughgmail or facebook	I can register & access the dashboard with Facebook Login	Medium	Sprint-2
	Login	USN-4	As a user, I can loginto	I can login to the	High	Sprint-1

			the applica tionby enterin gemail & password	applicatio n		

	Dashboard	USN-5	Go to dashboard and refer the content about our project	I can read instructionsalso and thehome page is user-friendly.	Low	Sprint-1
	Upload Image	USN-6	As a user, I can able to input the images of digital documentsto the application	As a user, I can able to input the images of digital documentsto the application	High	Sprint-3
	Predict	USN-7	As a user I can able to get the recognised digit as output fromthe	I can access the recognized digits from digital docume	High	Sprint-3

			images of digital docume nts or images	nt or images		
		USN-8	As a user, I willtrai n and test the	I can able to trainand test the	Medium	Sprint-4

			input to get themaximu m accuracy of output.	applicatio nuntil it gets maximum accuracy of the result.		
Customer (Webuser)	Login	USN -9	As a user, I can use the application by entering my email, password.	I can accessmy account	Mediu m	Sprint -4
Custome rCare Executive	Dashboar d	USN -10	upload the image	Recognize and get the output	High	Sprint -1

Administrator	Security	USN-11	updated the features	checking the security	Medium	Sprint-1

CHAPTER 6

PROJECT PLANNINGAND SCHEDULING

a. SPRINT PLANNINGAND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story /Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings.	10	Low	prasanth s rajaguru g sebastian prabu m

Sprint-1	Data Preprocessing	USN-2	As a user, I can load the dataset, handling the missing data,scaling and split data into train andtest.	10	Medium	prasanth s rajaguru g sebastian prabu m
Sprint-2	Model Building	USN-3	As a user, I will get an application with ML model which provides high accuracy of recognized handwrittendigit.	5	High	prasanth s rajaguru g siva s
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layersto it.	5	High	prasanth s rajaguru g siva s

Sprint-2	Compilingthe model	USN-5	With both thetraining data defined and model defined, it's time to configure the learning process.	2	Medium	prasanth s rajaguru g siva s
Sprint-2	Train & test themodel	USN-6	As a user, letus train our model with our image dataset.	6	Medium	prasanth s rajaguru g siva s

Sprint-2	Save the model	USN-7	As a user, the model is saved & integrated with anandroid application or webapplication inorder to predict something.	2	Low	prasanth s rajaguru g siva s
Sprint-3	Building UI Application	USN-8	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	5	High	rajaguru g sebastian prabu m
Sprint-3		USN-9	As a user, I can know thedetails of the fundamental usageof the application.	5	Low	rajaguru g sebastian prabu m

Sprint-3		USN -10	As a user, I can see the predicted / recognized digitsin the application.	5	Medium	rajaguru g sebastian prabu m

Sprint-4	Train the modelon IBM	USN-11	As a user, I train the model on IBMand integrate flask/Django with scoring end point.	10	High	prasanth s rajaguru g
Sprint-4	Cloud Deployment	USN-12	As a user, I can access the web application and make theuse of the product from anywhere.	10	High	prasanth s rajaguru g

b.SPRINT DELIVERYSCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story Points Completed (as on Planned End Date)	Sprint ReleaseDate (Actual)
Sprint -1	20	6 Days	29 Oct 2022	03 Nov 2022	20	03 Nov 2022
Sprint -2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint -3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint -4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

CHAPTER 7

CODING & SOLUTIONING

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
data_oil=scaler.fit_transform(np.array(data_oil).reshape(-1,1))
## **Data Visualization**
plt.title('Crude oil price')
plt.plot(data_oil)

## **Splitting data into Train and Test Data**
training_size=int(len(data_oil)*0.65)
test_size=len(data_oil)-training_size
train_data,test_data=data_oil[0:training_size,:],data_oil[training_size:len(data_oil),:1]
training_size,test_size
train_data.shape

## **Creating a dataset with sliding windows**
def create_dataset (dataset, time_step=1):
    dataX, dataY = [], []

    for i in range(len(dataset)-time_step-1):
        a = dataset[i:(i+time_step), 0]
        dataX.append(a)
        dataY.append(dataset[i + time_step, 0])

    return np.array(dataX),np.array(dataY)
time_step = 10

X_train, y_train=create_dataset(train_data,time_step)


X_test, y_test = create_dataset(test_data,time_step)
print(X_train.shape),print(y_train.shape)
print(X_test.shape),print(y_test.shape)
X_train
X_train.shape
X_train=X_train.reshape(X_train.shape[0],X_train.shape[1],1)
X_test=X_test.reshape(X_test.shape[0],X_test.shape[1],1)
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM
model=Sequential()
model.add(LSTM(50,return_sequences=True,input_shape=(10,1)))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
model.add(Dense(1))
model.summary()
model.compile(loss='mean_squared_error',optimizer='adam')
model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=3,batch_size=64,verbose=1)
##Transformback to original form
train_predict=scaler.inverse_transform(train_data)
test_predict=scaler.inverse_transform(test_data)
### Calculate RMSE performance metrics
import math
from sklearn.metrics import mean_squared_error
math.sqrt(mean_squared_error(train_data,train_predict))
from tensorflow.keras.models import load_model
model.save("crude_oil.h5")
### Plotting
look_back=10
trainpredictPlot = np.empty_like(data_oil)
trainpredictPlot[:, :]= np.nan
trainpredictPlot[look_back:len(train_predict)+look_back, :] = train_predict
# shift test predictions for plotting
testPredictplot = np.empty_like(data_oil)
testPredictplot[:,: ] = np.nan
testPredictplot[look_back:len(test_predict)+look_back, :] = test_predict
# plot baseline and predictions
plt.plot(scaler.inverse_transform(data_oil))
plt.show()
len(test_data)
x_input=test_data[2866:].reshape(1,-1)
x_input.shape
temp_input=list(x_input)
```

```
temp_input=temp_input[0].tolist()
temp_input
lst_output=[]
n_steps=10
i=0
while(i<10):
    if(len(temp_input)>10):
#print(temp_input)
        x_input=np.array(temp_input[1:])
        print("{} day input {}".format(i,x_input))
        x_input=x_input.reshape(1,-1)
        x_input = x_input.reshape((1, n_steps, 1)) #print(x_input)
        yhat = model.predict(x_input, verbose=0)
        print("{} day output {}".format(i,yhat))
        temp_input.extend(yhat[0].tolist())
        temp_input=temp_input[1:] #print(temp_input)
        lst_output.extend(yhat.tolist())
        i=i+1
    else:
        x_input = x_input.reshape((1, n_steps,1))
        yhat = model.predict(x_input, verbose=0)
        print(yhat[0])
        temp_input.extend(yhat[0].tolist())
        print(len(temp_input))
        lst_output.extend(yhat.tolist())
        i=i+1
day_new=np.arange(1,11)
```

```
day_pred=np.arange(11,21)
len(data_oil)
plt.plot(day_new, scaler.inverse_transform(data_oil[8206:]))
plt.plot(day_pred, scaler.inverse_transform(lst_output))
df3=data_oil.tolist()
df3.extend(lst_output)
plt.plot(df3[8100:])
df3=scaler.inverse_transform(df3).tolist()
plt.plot(scaler.inverse_transform(data_oil))
# @hidden_cell
# The following code contains the credentials for a file in your IBM Cloud Object Storage.
# You might want to remove those credentials before you share your notebook.
metadata_1 = {
    'IAM_SERVICE_ID': 'iam-ServiceId-b9c18a80-9780-4270-bf88-fc8afee2ca22',
    'IBM_API_KEY_ID': 'bM-UBMSwFCa7R0jKoO6AyQaUYNWQfm0p3Oqyqmh7so4x',
    'ENDPOINT': 'https://s3.private.us.cloud-object-storage.appdomain.cloud',
    'IBM_AUTH_ENDPOINT': 'https://iam.cloud.ibm.com/oidc/token',
    'BUCKET': 'crudeoilpriceprediction-donotdelete-pr-ikcsbsjvdluquo',
    'FILE': 'Crude Oil Prices Daily.xlsx'
}
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "bM-UBMSwFCa7R0jKoO6AyQaUYNWQfm0p3Oqyqmh7so4x"
}

client = APIClient(wml_credentials)
```

```
def guid_from_space_name(client, space_name):

   space = client.spaces.get_details()

   return (next(item for item in space['resources'] if item['entity']["name"] ==
space_name)['metadata']['id'])

space_uid = guid_from_space_name(client, 'crudeoilspace')

print("Space UID : ", space_uid)

client.set.default_space(space_uid)

client.software_specifications.list()software_spec_uid =
client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")

software_spec_uid

model_details = client.repository.store_model(model="trainedModel.tgz", meta_props={

   client.repository.ModelMetaNames.NAME: "sequential",

   client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid,

   client.repository.ModelMetaNames.TYPE: "tensorflow_2.7"})

model_id = client.repository.get_model_uid(model_details)

model_id

X_train[0]

model_id

model.predict(([[0.11335703],

      [0.11661484],

      [0.12053902],

      [0.11550422],

      [0.1156523 ],

      [0.11683696],

      [0.1140234 ],

      [0.10980305],

      [0.1089886 ],

      [0.11054346]]))
```

flask file:

```
import numpy as np
import os
from flask import Flask, render_template, request
from tensorflow.keras.models import load_model
from werkzeug.utils import secure_filename, redirect
from gevent.pywsgi import WSGIServer
from flask import send_from_directory



model = load_model("crude_oil.h5")


app = Flask (__name__,template_folder='templates')
@app.route('/')
def home():
    return render_template("index.html")
@app.route('/about')
def home1():
    return render_template("index.html")
@app.route('/predict')
def home2():
    return render_template("web.html")
```

```
@app.route('/predict', methods=['POST'] )

def login() :

    x_input=str(request.form['year'])

    x_input=x_input.split(',')

    print(x_input)



    for i in range(0, len(x_input)):

       x_input[i] = float(x_input[i])

    print(x_input)

    x_input=np.array(x_input).reshape(1,-1)

    temp_input=list(x_input)

    temp_input=temp_input[0].tolist()

    lst_output=[]

    n_steps=10

    i=0

    while(i<1):

      if(len(temp_input)>10):

         x_input=np.array(temp_input[1:])

         print("{} day input {}".format(i,x_input))

         x_input=x_input.reshape(1,-1)

         x_input = x_input.reshape((1, n_steps, 1))

         yhat = model.predict(x_input, verbose=0)

         print("{} day output {}".format(i,yhat))

         temp_input.extend(yhat[0].tolist())

         temp_input=temp_input[1:]

         lst_output.extend(yhat.tolist())
```

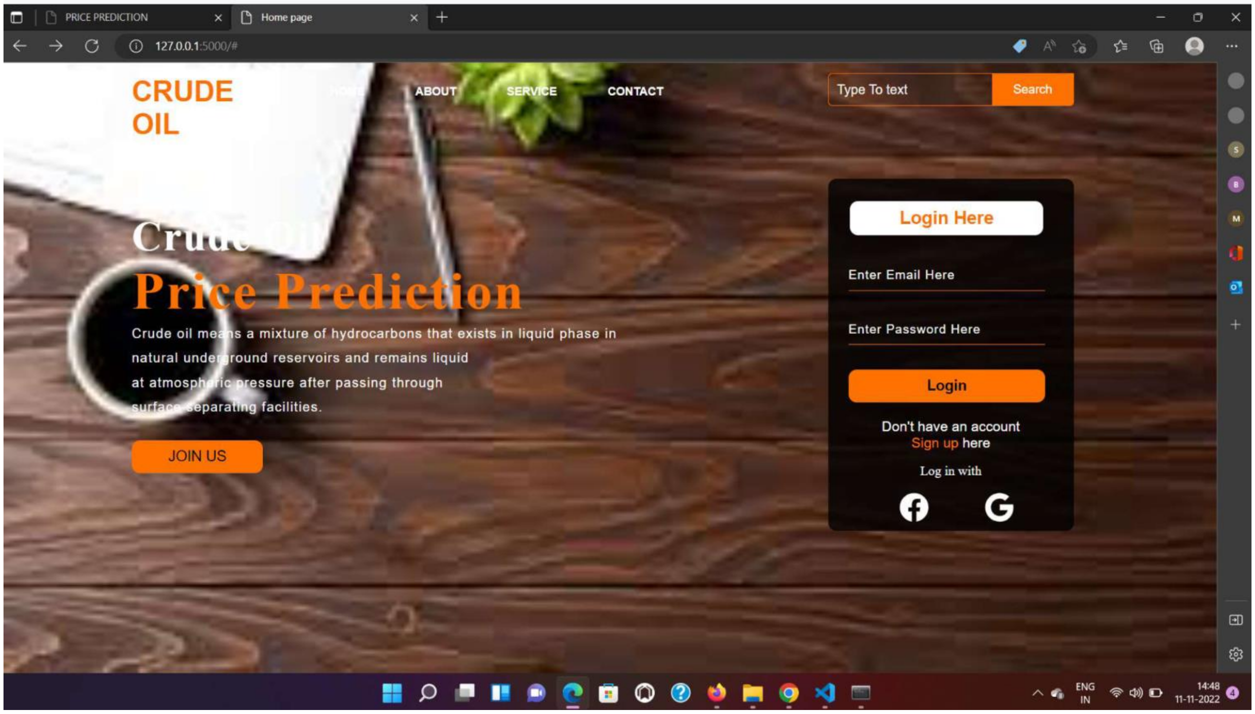
```
            i=i+1
        else:
            x_input = x_input.reshape((1,n_steps,1))
            yhat = model.predict(x_input,verbose=0)
            print(yhat[0])
            temp_input.extend(yhat[0].tolist())
            print(len(temp_input))
            lst_output.extend(yhat.tolist())
            i=i+1
        print(lst_output)
        return render_template("web.html", showcase = 'The next day predicted value is:'+str(lst_output))


port = os.getenv('VCAP_APP_PORT','8080')


if __name__ == '__main__' :
    app.secret_key = os.urandom(12)
    app.run(debug = True,port=port,host='0.0.0.0')
```

CHAPTER 9

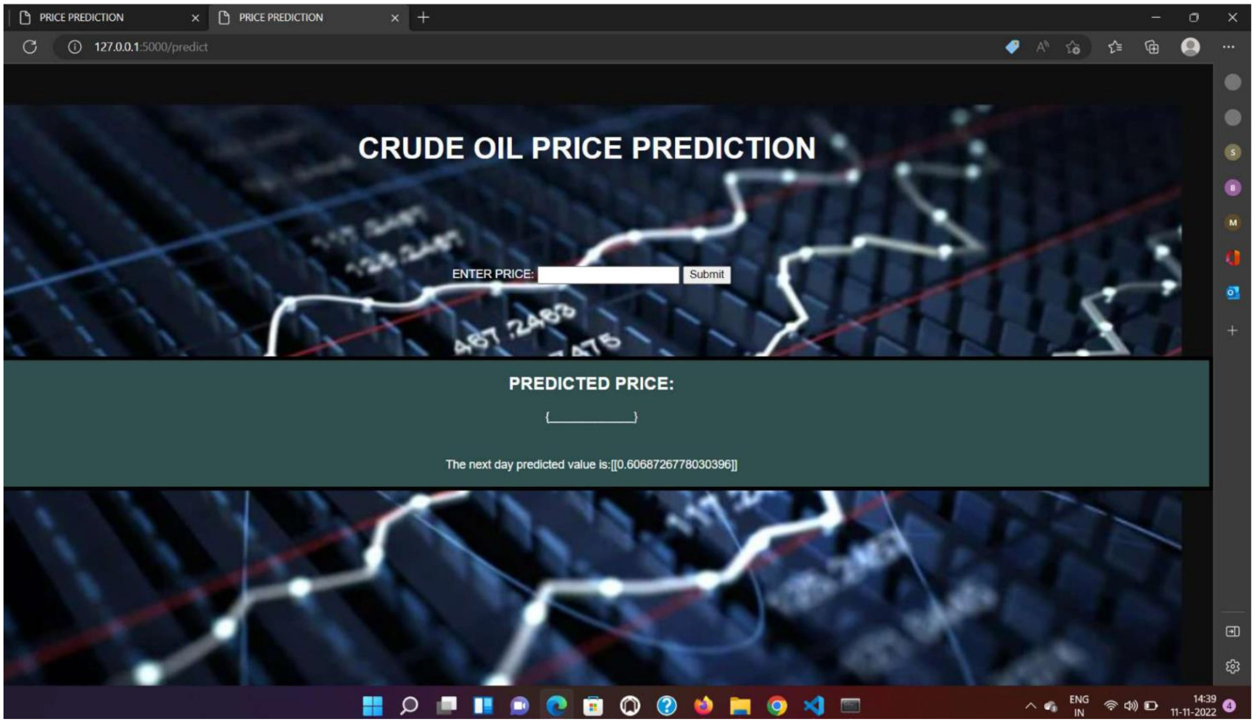
RESULTS



Crude Oil Price Prediction

Crude oil means a mixture of hydrocarbons that exists in liquid phase in natural underground reservoirs and remains liquid at atmospheric pressure after passing through surface separating facilities.

Don't have an account Sign up here



CRUDE OIL PRICE PREDICTION

ENTER PRICE: Submit

PREDICTED PRICE:

{___________}

The next day predicted value is:[[0.6068726778030396]]

CHAPTER10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

i. Reduces manual work

ii. More accurate than averagehuman

iii. Capable of handlinga lot of data

iv. Can be used anywherefrom any device

DISADVANTAGES

i. Cannot handle complexdata

ii. All the data must be in digital format

iii. Requires a high performance server for fasterpredictions

iv. Prone to occasional errors

CHAPTER 11

CONCLUSION

This project demonstrated a web application that uses machine learning to recognise handwritten numbers. Flask, HTML, CSS, and a few other technologies wereused to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61% recognition rate. The proposed project is scalable and can easily handle a huge number of users.Since it is a web application, it is compatible with any device that can run a browser.This project is extremely useful in real-world scenarios such as recognizing numberplates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand(tax forms) and so on. There is so much room for improvement, which can be implemented in subsequent versions.

CHAPTER 12

FUTURE SCOPE

This project is far from complete and there is a lot of

room for improvement.

Some of the improvements that can be made to this projectare as follows:

i. Add support to detect from digits multipleimages and save the results

ii. Add support to detect multiple digits

iii. Improve model to detect digitsfrom complex images

iv. Add support to different languagesto help users from all over the world

This project has endless potential and can always be enhanced to become better.Implementing this concept in the real world will benefit several industries and reducethe workload on many workers,enhancing overall work efficiency.

APPENDIX

SOURCE CODE

MODEL CREATION

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home page</title>
    <link rel="stylesheet" href="../static/css/style.css">
</head>
<body>


    <div class="main">
        <div class="navbar">
            <div class="icon">
                <h2 class="logo">CRUDE OIL</h2>
            </div>


            <div class="menu">
                <ul>
                    <li><a href="#">HOME</a></li>
                    <li><a href="#">ABOUT</a></li>
                    <li><a href="#">SERVICE</a></li>
                    <li><a href="#">CONTACT</a></li>
                </ul>
            </div>


            <div class="search">
        <input class="srch" type="search" name="" placeholder="Type To text">
        <a href="#"> <button class="btn">Search</button></a>
      </div>


    </div>
    <div class="content">
      <h1>Crude Oil<br><span>Price Prediction</span><br></h1>
      <p class="par"> Crude oil means a mixture of hydrocarbons that exists in liquid phase
in<br>
        natural underground reservoirs and remains liquid <br>at atmospheric pressure
        after passing through <br>surface separating facilities.</p>


        <button class="cn"><a href="register.html">JOIN US</a></button>


        <div class="form">
          <h2>Login Here</h2>
          <input type="email" name="email" placeholder="Enter Email Here">
          <input type="password" name="" placeholder="Enter Password Here">
          <button class="btnn"><a href="/predict">Login</a></button>


          <p class="link">Don't have an account<br>
          <a href="#">Sign up </a> here</a></p>
          <p class="liw">Log in with</p>


          <div class="icons">
            <a href="#"><ion-icon name="logo-facebook"></ion-icon></a>
```

```
                        <a href="#"><ion-icon name="logo-google"></ion-icon></a>
                    </div>


                </div>
                    </div>
                </div>
        </div>
    </div>
    <script src="https://unpkg.com/ionicons@5.4.0/dist/ionicons.js"></script>
</body>
</html>
```

web.html

```
<!DOCTYPE html>
<html>


<head>
  <title>PRICE PREDICTION</title>
</head>


<body>
  <p>&nbsp;</p>


  <p>&nbsp;</p>
  <style>
    body {
```

```
  background-image: url('../static/images/background.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
}

form {
  min-height: 70%;
}

body,
form {
  padding: 0;
  margin: 0;
  outline: none;
  font-family: Roboto, Arial, sans-serif;
  font-size: 14px;
  color: #FFFFFF;
  line-height: 22px;
}

.myDiv {
  border: 5px outset #000000;
  background-color: #2F4F4F;
  text-align: center;
  font-family: Roboto, Arial, sans-serif;
```

```
   font-size: 14px;
   color: #FFFFFF;
  }
 </style>

 <h1 style="text-align:center"><span style="color:#ffffff"><strong><span
     style="font-family:Arial,Helvetica,sans-serif"><span style="font-size:36px">CRUDE OIL
PRICE
      PREDICTION&nbsp;</span></span></strong></span></h1>
 <p>&nbsp;</p>
 <p>&nbsp;</p>
 <p>&nbsp;</p>
 <form method="post" style="text-align:center">
  <label for="ENTER PRICE">ENTER PRICE:</label>
  <input type="text" id="PRICE" name="year">
  <input type="submit">
 </form>

 <p>&nbsp;</p>
 <p>&nbsp;</p>

 <div class="myDiv">
  <h2>PREDICTED PRICE:</h2>
  <p>{_____________}</P><br>
  {{ showcase }}
```

```
    </p>
  </div>

</body>

</html>
</style>
<table border="1" class="dataframe">
  <thead>
    <tr style="text-align: right;">
      <th></th>
      <th>Date</th>
      <th>Closing Value</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>0</th>
      <td>1986-01-02</td>
      <td>25.56</td>
    </tr>
    <tr>
      <th>1</th>
      <td>1986-01-03</td>
      <td>26.00</td>
    </tr>
    <tr>
      <th>2</th>
      <td>1986-01-06</td>
      <td>26.53</td>
    </tr>
    <tr>
      <th>3</th>
      <td>1986-01-07</td>
      <td>25.85</td>
    </tr>
    <tr>
      <th>4</th>
      <td>1986-01-08</td>
      <td>25.87</td>
    </tr>
  </tbody>
</table>
</div>
data.head()
<div>
<style scoped>
    .dataframe tbody tr th:only-of-type {
        vertical-align: middle;
    }
    .dataframe tbody tr th {
        vertical-align: top;
    }

    .dataframe thead th {
        text-align: right;
    }
</style>
<table border="1" class="dataframe">
  <thead>
    <tr style="text-align: right;">
      <th></th>
      <th>Date</th>
      <th>Closing Value</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>0</th>
      <td>1986-01-02</td>
      <td>25.56</td>
    </tr>
    <tr>
      <th>1</th>
      <td>1986-01-03</td>
      <td>26.00</td>
    </tr>
    <tr>
      <th>2</th>
      <td>1986-01-06</td>
      <td>26.53</td>
    </tr>
    <tr>
      <th>3</th>
      <td>1986-01-07</td>
      <td>25.85</td>
    </tr>
    <tr>
      <th>4</th>
      <td>1986-01-08</td>
      <td>25.87</td>
    </tr>
  </tbody>
</table>
</div>
```