Appendix G: Regression model

Code for prediction of single phase flow

```
# -*- coding: utf-8 -*-
"""Oil_flow_prediction.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1YCbu46eNylfIQ-JNJBUwLch4woa4v0gT
"""

# Commented out IPython magic to ensure Python compatibility.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# %matplotlib inline

from google.colab import files
import io
data = files.upload()

df = pd.read_csv(io.StringIO(data['OILFLOW_V1.csv'].decode('utf-8')))

df.head()

df.describe()

df.columns

sns.heatmap(df.corr())

X=df[['AVG_PDT121 [mbar]', 'MIN_PDT121', 'MAXPDT121']]
y=df['oil_kg_min']
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=51)

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
sc.fit(X_train)
X_train = sc.transform(X_train)
X_test = sc.transform(X_test)

from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train,y_train)
lr.score(X_test,y_test)

from sklearn.svm import SVR

svr_rbf = SVR(kernel = 'rbf')
svr_rbf.fit(X_train,y_train)
svr_rbf.score(X_test,y_test)

svr_linear = SVR(kernel = 'linear')
svr_linear.fit(X_train,y_train)
svr_linear.score(X_test,y_test)

svr_poly = SVR(kernel = 'poly',degree = 2)
svr_poly.fit(X_train,y_train)
svr_poly.score(X_test,y_test)

y_pred = lr.predict(X_test)
y_pred

y_test
```

```
from sklearn.metrics import mean_squared_error
import math
result = math.sqrt(mean_squared_error(y_test,y_pred))
# Print the result
print("RMSE:", result)

df.hist(bins=50, figsize=(20,15))
plt.show()

def plot_diff(y_test, y_pred, title=''):
    plt.title(title)
    plt.plot(y_test,label="True Values")
    plt.plot(y_pred,label="Predicted Values")
    plt.xlim(plt.xlim())
    plt.ylim(plt.ylim())
    plt.legend()
    plt.show()

def plot_metrics(metric_name, title, ylim=5):
    plt.title(title)
    plt.ylim(0, ylim)
    plt.plot(history.history[metric_name], color='blue', label=metric_name)
    plt.plot(history.history['val_' + metric_name], color='green', label='val_' + metric_name)
    plt.show()

y_pred

y_test

y_test = np.array(y_test)

plot_diff(y_test,y_pred,title="Oil Flow Prediction")
prediction
```