SURFACE MINING

APPLICATION OF ARTIFICIAL INTELLIGENCE IN ACID MINE DRAINAGE PREDICTION

TEAM MEMBERS:

Kavya Venkatagiri - Roll no 18MI33019

Samiksha Sharma - Roll no 18MI31015

Gunti Jyothsna sri - Roll no18MI33007

Arindam Bera - Roll no 18MI33003

CONTENTS

OBJECTIVES

Application of AI for predicting most important components of AMD

To Study Acid Mine Drainage effects and causes

AGENDA

AM
D

- Most serious environmental problems in the mining industry.
- Disruption of the hydrology of mining areas
- Most focussed research topic

- While knowledge of the acid formation process is incomplete, several factors are known to control the production

- In this mini project, we discuss various factors, review the techniques and models currently used to predict mine drainage quality.

Factors controlling AMD

1. **PRIMARY FACTORS**
   (pyrite and other sulphide minerals, oxygen, water, ferric iron and iron oxidizing bacteria)

2. **SECONDARY FACTORS**
   (carbonate minerals such as calcite($CaCO_3$) and dolomite($CaMg(CO_3)_2$)

3. **TERTIARY FACTORS**
   (physical aspects)

LOW pH -> HIGH SULPHATE

Acid mine drainage arises from

Sulfide Sources	**Water Sources**
• Tailings piles • Waste rock piles • Ore stockpiles • Heap leach materials • Pit walls • Underground workings • Sulfides in unmined ore Host rock	• Infiltration of rainwater • Groundwater seepage • Surface water ingress

Mixing zones	**Discharge mechanism**
• Unsaturated zones in tailings/waste rock/ore piles • Open pits above pit lakes • Infiltration zones in underground mines	• Water pumped from mine workings • Gravity-driven discharge from workings • Seepage

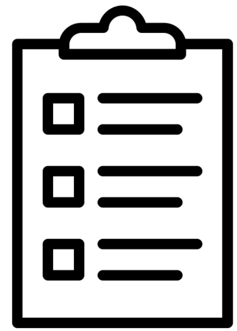
↓

Recieving environment

- Surface water
- Groundwater
- Soil

Acid Mine Drainage Prediction Techniques

WHY PREDICTION??

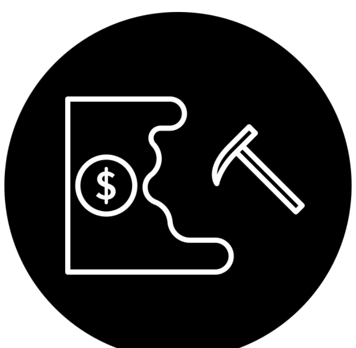


Mine site regulatory standards

Created by Justin Blake

Protect the environment

Wise expenditure of resources



Created by Astatine Lab
from Noun Project

Government Regulatory Standards

- pH -5.5 to 9.0
- Sulphide (as S)mg/L max -5.0
- Chemical Oxygen Demand (COD) -250 mg/l
- Oil & grease (O&G) -10 mg/l

Note: Standards for Coal Mines, Stipulated by Ministry of Environment and Forests (MoEF)

Acid Mine Drainage Prediction Techniques

Methods to predict acid mine drainage can be divided into five groups:

1. **Geographical mining comparisons;**
2. **Paleoenvironmental and geological models;**
3. **Geochemical static tests;**
4. **Geochemical kinetic tests;**
5. **Mathematical models.**
6. **Artificial Intelligence for AMD Quality prediction**

AI for AMD

Different AI Techniques Used to Predict AMD Quality

Common AI tools used for predictive modeling of water quality can be classified into:

KNOWLEDGE DRIVEN

DATA DRIVEN

The widely used data-driven tools

1. Regression analysis
2. Artificial neural networks (ANNs)
3. Random forest (RF)
4. Decision trees
5. Support vector machines (SVMs),
6. Radial base functions (RBFs)
7. and Genetic algorithms.

APPLICATION OF AI FOR THE PREDICTION OF MINE DRAINAGE IS NOT NEW

METHODOLOGY

This study will investigate the use of Artificial neural networks(ANNs) for mine water quality prediction by implementing an AI based model

PH of mine water

Rainfall (recharge)

Water table

Temperature

Sulphate concentration

Data collection:

LIMITATION (Privacy matters for mine areas)

Data provided in research paper

Webplot designer (Convert the image data in the paper to dataset csv file)

Approximated data set

Coal mine in Carolina Town, Mpumalanga Province of South Africa

WebPlotDigitizer

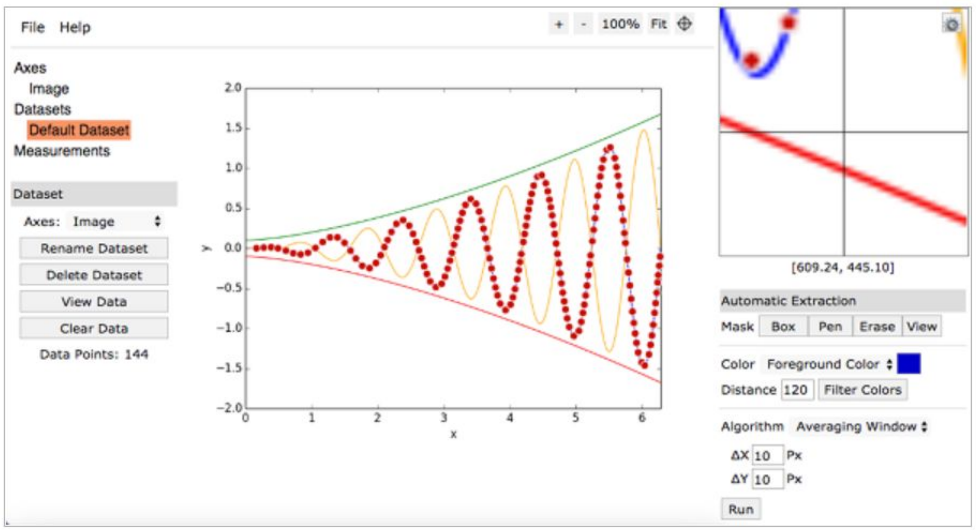
Web based tool to extract data from plots, images, and maps

English

Launch

Download

Home Blog Tutorials Citation Privacy



Web Application

English

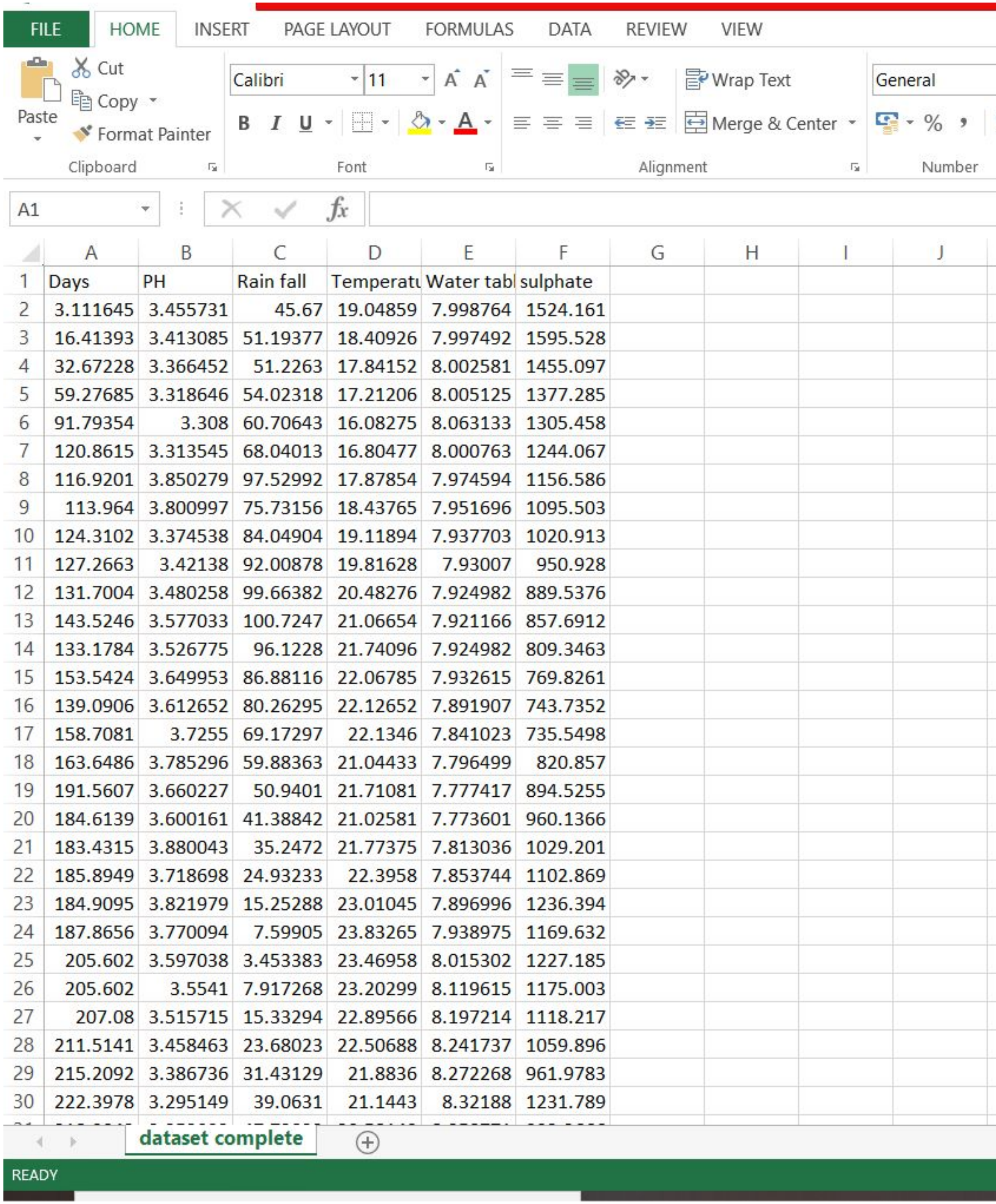
Launch Now!

Desktop Version

View Source

GitHub

It is often necessary to reverse engineer images of data visualizations to extract the underlying numerical data. WebPlotDigitizer is a semi-automated tool that makes this process extremely easy:

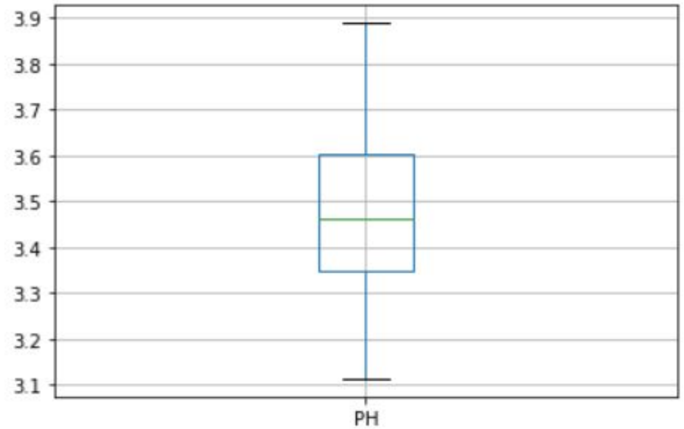


	Days	PH	Rain fall	Temperatu	Water tabl	sulphate
2	3.111645	3.455731	45.67	19.04859	7.998764	1524.161
3	16.41393	3.413085	51.19377	18.40926	7.997492	1595.528
4	32.67228	3.366452	51.2263	17.84152	8.002581	1455.097
5	59.27685	3.318646	54.02318	17.21206	8.005125	1377.285
6	91.79354	3.308	60.70643	16.08275	8.063133	1305.458
7	120.8615	3.313545	68.04013	16.80477	8.000763	1244.067
8	116.9201	3.850279	97.52992	17.87854	7.974594	1156.586
9	113.964	3.800997	75.73156	18.43765	7.951696	1095.503
10	124.3102	3.374538	84.04904	19.11894	7.937703	1020.913
11	127.2663	3.42138	92.00878	19.81628	7.93007	950.928
12	131.7004	3.480258	99.66382	20.48276	7.924982	889.5376
13	143.5246	3.577033	100.7247	21.06654	7.921166	857.6912
14	133.1784	3.526775	96.1228	21.74096	7.924982	809.3463
15	153.5424	3.649953	86.88116	22.06785	7.932615	769.8261
16	139.0906	3.612652	80.26295	22.12652	7.891907	743.7352
17	158.7081	3.7255	69.17297	22.1346	7.841023	735.5498
18	163.6486	3.785296	59.88363	21.04433	7.796499	820.857
19	191.5607	3.660227	50.9401	21.71081	7.777417	894.5255
20	184.6139	3.600161	41.38842	21.02581	7.773601	960.1366
21	183.4315	3.880043	35.2472	21.77375	7.813036	1029.201
22	185.8949	3.718698	24.93233	22.3958	7.853744	1102.869
23	184.9095	3.821979	15.25288	23.01045	7.896996	1236.394
24	187.8656	3.770094	7.59905	23.83265	7.938975	1169.632
25	205.602	3.597038	3.453383	23.46958	8.015302	1227.185
26	205.602	3.5541	7.917268	23.20299	8.119615	1175.003
27	207.08	3.515715	15.33294	22.89566	8.197214	1118.217
28	211.5141	3.458463	23.68023	22.50688	8.241737	1059.896
29	215.2092	3.386736	31.43129	21.8836	8.272268	961.9783
30	222.3978	3.295149	39.0631	21.1443	8.32188	1231.789

This is how our input dataset looks like. We can add more columns as input parameters if we have its data.

```
print(df.boxplot(column='PH'))
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```

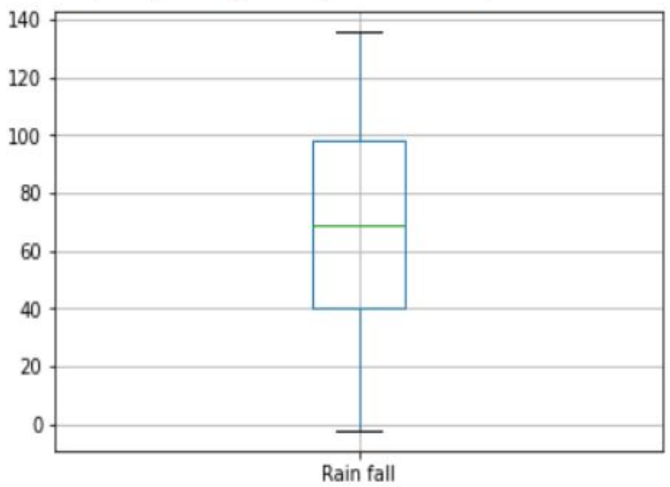


As we can see, pH levels show acidic levels.

Inference : pH should be in 5.5 to 9 range but it is much lower than that

```
[91] print(df.boxplot(column='Rain fall'))
```

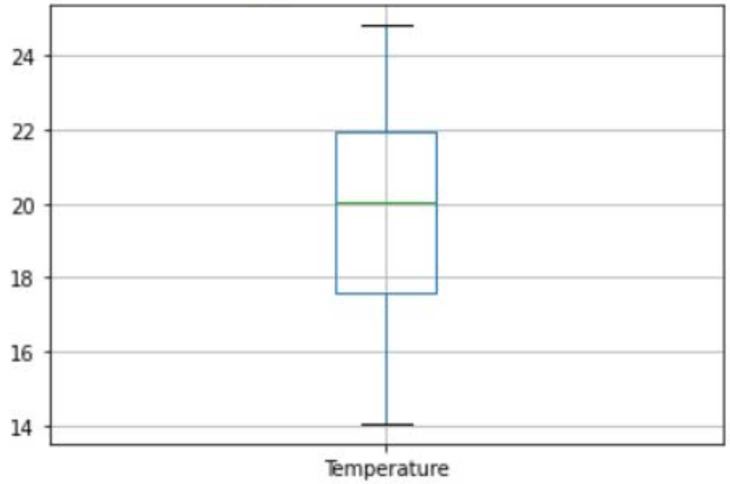
```
AxesSubplot(0.125,0.125;0.775x0.755)
```



Rainfall

```
print(df.boxplot(column='Temperature'))
```

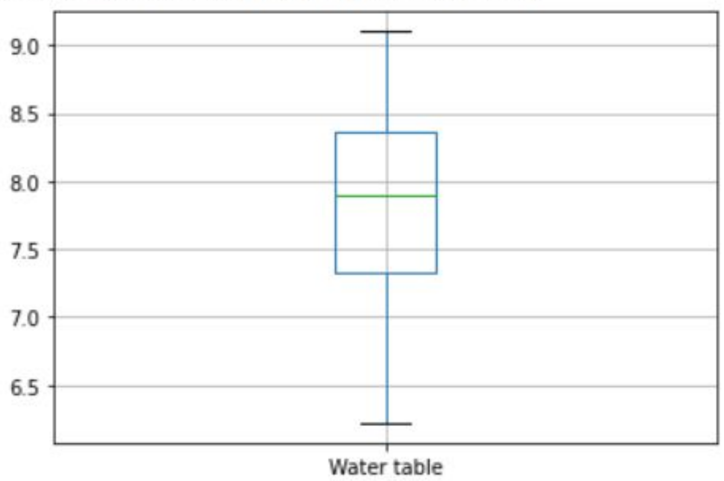
AxesSubplot(0.125,0.125;0.775x0.755)



Temperature

```
print(df.boxplot(column='Water table'))
```

AxesSubplot(0.125,0.125;0.775x0.755)

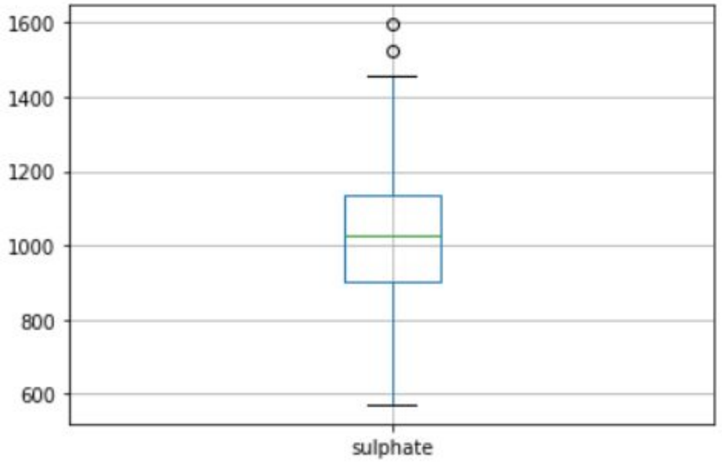


Water Table

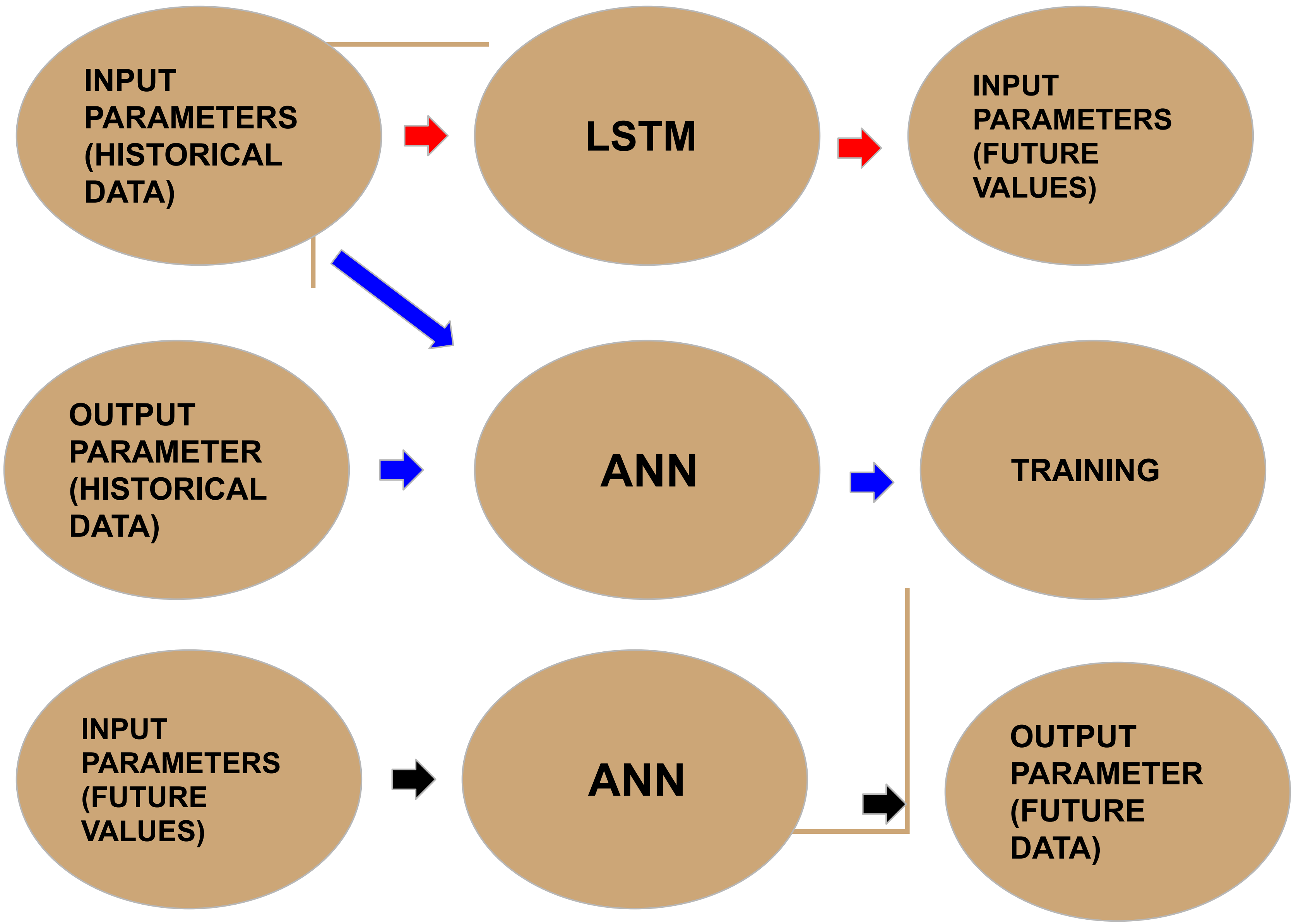
Sulphate

```
4] print(df.boxplot(column='sulphate'))
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



FLOW CHART:


INPUT PARAMETERS (HISTORICAL DATA)
LSTM
INPUT PARAMETERS (FUTURE VALUES)
OUTPUT PARAMETER (HISTORICAL DATA)
ANN
TRAINING
INPUT PARAMETERS (FUTURE VALUES)
ANN
OUTPUT PARAMETER (FUTURE DATA)

DATA SET

```
df.head()
```

	Days	PH	Rain fall	Temperature	Water table	sulphate
0	3.111645	3.455731	45.670000	19.048588	7.998764	1524.161480
1	16.413930	3.413085	51.193774	18.409260	7.997492	1595.527896
2	32.672277	3.366452	51.226296	17.841517	8.002581	1455.097208
3	59.276846	3.318646	54.023181	17.212062	8.005125	1377.284794
4	91.793540	3.308000	60.706435	16.082747	8.063133	1305.457950

```python
X=df.drop(['sulphate'],axis=1).values
Y=df['sulphate'].values
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(X,Y,test_size = 0.20, random_state = 0)
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(128, 5)
(128,)
(33, 5)
(33,)
```

ANN

```
#ANN
classifier = Sequential()

classifier.add(Dense(units= 6, kernel_initializer = 'normal', activation = 'relu',input_dim=5))
classifier.add(Dense(units= 6, kernel_initializer = 'normal', activation = 'relu'))
classifier.add(Dense(units = 1, kernel_initializer = 'normal', activation = 'sigmoid'))

classifier.compile(optimizer = 'adam', loss='mean_squared_error',metrics=['accuracy'])

classifier.fit(x_train, y_train, batch_size = 10, epochs = 100)
```

```
Epoch 47/100
13/13 [==============================] - 0s 2ms/step - loss: 438614.4955 - accuracy: 0.0000e+00
```

ACCURACY

```python
import sklearn.metrics as sm

# Mean squared error: This is the average of the squares of the errors of all the data points in the given dataset.
# It is one of the most popular metrics out there!
print("Mean squared error =", round(sm.mean_squared_error(y_test, y_pred), 2))

# Explained variance score: This score measures how well our model can account for the variation in our dataset.
# A score of 1.0 indicates that our model is perfect.
print("Explain variance score =", round(sm.explained_variance_score(y_test, y_pred), 2))

# R2 score: This is pronounced as R-squared, and this score refers to the coefficient of determination.
# This tells us how well the unknown samples will be predicted by our model.
# The best possible score is 1.0, but the score can be negative as well.
print("R2 score =", round(sm.r2_score(y_test, y_pred), 2))
```

```
Mean squared error = 38808.86
Explain variance score = 0.17
R2 score = -0.26
```

Results

Converting date to time series format to recognize the change in our base parameters(ph, rainfall, temperature, etc) with respect to time.

AI model will predict the sulphate concentration to be present in the future considering current trends.

Given the methodology we have followed, if we obtain a real time dataset, our prediction will be much more accurate and scalable.

We have an idea of considering the groundwater parameters like aquifer details to contribute as a parameter to understand the impact of the AMD pollution on groundwater at a particular place.

Limitations of AI Techniques in Prediction of AMD Quality

AI techniques have their own limitations which include:

a) Time consuming processing and training of the model as optimization is heuristically and repeatedly performed until a good model or learning technique is obtained.

b) Most experts find it difficult to interpret and explain decisions made by AI models, suggesting its black box nature and limiting its successful application.

For any prediction, including mine drainage quality, a thorough investigation and understanding of the fundamental physical processes is compulsory. However, it is challenging to involve domain expert knowledge directly into the data-driven approaches

Hence, hybrid approaches are developed to mitigate this problem.

Reference

- **IMWA 2019 “Mine Water: Technological and Ecological Challenges” Wolkersdorfer, Ch.; Khayrulina, E.; Polyakova, S.; Bogush, A. (Editors)   141 mine water quality, (intro)**
- **Pre-Mine Prediction of Acid Mine Drainage K.D. FERGUSONl and P.M. ERICKSON2**
- **https://automeris.io/WebPlotDigitizer/ =>Web plot digitizer**
- https://link.springer.com/content/pdf/10.1007%2F978-3-642-61362-3_2.pdf
- https://www.imwa.info/docs/imwa_2019/IMWA2019_Sakala_140.pdf
- https://www.researchgate.net/publication/345006253_Mine_Water_Treatment_and_the_Use_of_Artificial_Intelligence_in_Acid_Mine_Drainage_Prediction
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5981990/
- https://www.icontrolpollution.com/articles/acid-mine-drainage-amd-management-283-295.pdf
- https://www.icontrolpollution.com/articles/acid-mine-drainage-amd-management-.php?aid=45718
- https://www.sciencedirect.com/science/article/pii/S088329270400143X?casa_token=rgohtE1mQIsAAAAA:kb1iE6hQ3Qii4qnpzcCFzpkaT19c7eh6Kq42LEDHhrAGrC2GDU_bpYMTQlKifpZeoAXMuGh71Kji
- https://link.springer.com/content/pdf/10.1007%2F978-3-642-61362-3_2.pdf
- https://www.icontrolpollution.com/articles/acid-mine-drainage-amd-management-.php?aid=45718
- https://www.icontrolpollution.com/articles/acid-mine-drainage-amd-management-283-295.pdf
- https://link.springer.com/content/pdf/10.1007%2F978-3-642-61362-3_2.pdf
- https://www.researchgate.net/publication/345006253_Mine_Water_Treatment_and_the_Use_of_Artificial_Intelligence_in_Acid_Mine_Drainage_Prediction

THANK
YOU