**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  |  |  |
|  | **ABSTRACT** | **4** |
| **1** | **INTRODUCTION** | **5** |
| **2** | **OBJECTIVES** | **6** |
| **3** | **METHODOLOGY** | **7** |
| **4** | **PROGRAM AND OUTPUT** | **8** |
| **5** | **RESULTS AND DISCUSSIONS** | **13** |
| **6** | **CONCLUSION** | **14** |

**ABSTRACT**

A solid grasp of electrospun fiber structure-property relationships is critical for device design. However, These are naturally complicated relationships that are difficult to model using data from limited trial and error studies. Machine learning has evolved as an efficient method for modeling multidimensional interactions, but it requires a broad set of data to learn these relationships from. In this paper, we introduce a unique Electrospun fabric. Fiber Experimental Attributes Dataset (FEAD) was created by collecting experimental data from the literature and developing new algorithms. characteristics, as well as supplementing with our own experiments. A large fiber diameter is used to represent fiber diameter, an important parameter for influencing the electrical and thermal properties of electrospun polyvinylidene fluoride (PVDF) polymer.

A multi-model machine learning approach was used to determine the number of solution and electrospinning process experimental parametersA model-neutral, interpretable game-theoretic approach is used in conjunction with this to determine therelationships between the variables that are both absolute and relative. The most significant experimental parameters were discovered to be feed, polymer concentration, Flory-Huggins Chi parameter, and relative energy difference, simulation of fiber diameter. This study circumvents a number of shortcomings in previous research, including non-availability. Using meta datasets, the most recent machine learning methods, and cutting-edge methods for comparing these "black box" models, bridging the gap between computational and experimental studies. This improved ability to generalize structure–property relationships across any PVDF-polymer solvent system presents a promising ability to reduce expensive lab testing required for developing PVDF fibers of desired mechanical and electrical properties.

**CHAPTER 1**

**INTRODUCTION**

Electroresponsive polymer (ERP) materials are one of the most promising technologies with extensive mechanical and electronic applications in diverse fields such as sensing, actuating, energy harvesting, etc. Poly(vinylidene fluoride) (PVDF) possesses high dielectric constant, piezoelectric, pyroelectric and ferroelectric effects which makes it one of the highly investigated candidates for developing exciting technological devices.

Many fabrication methods have been explored to obtain these PVDF films used in sensors, batteries and nanogenerators. Electrospinning is preferred due to its simplicity of operation, where a liquid droplet is electrified to generate a jet that gets stretched and collected in the form of microfibers or nanofibers. However, the process is intricate and involves the understanding of electrostatics, fluid rheology, and polymer solution system properties to obtain targeted fiber properties. Studies have demonstrated that fiber properties and behavior could be enhanced or changed to a great extent by regulating their morphologies such as fiber diameter.

Considerable effort is required to understand, design, and optimize the electrospinning process to obtain fiber characteristics tailored for specific applications Zaarour et al. reported the impact of PVDF solution concentrations on fiber morphology, but nonlinear relationships in electro-hydrodynamics, solvent loss and material properties such as intrinsic viscosity or electrical conductivity made it difficult to accurately model fiber morphology both empirically and experimentally. Attempts have been made to model and analyze the effect of process variables and solution properties on electrospun fiber diameters.

Artificial neural networks (ANNs) have been trained to predict fiber diameters, and Response surface methodology (RSM) has been used to establish processing parameter-fiber diameter relationships. Ziabari et al. used RSM to establish processing parameter-fiber diameter relationships by looking into mean and standard deviation in the fiber diameter against processing variables. Nasouri et al used RSM with artificial neural networks (ANNs) to predict the highest production rate of fibers, but almost all the work reported includes only a few experimental data points (20-30) for a distinct polymer–solvent combination.

**CHAPTER 2**

**OBJECTIVE**

The FEAD dataset reported in this study allows for regression modeling of multivariate relationships between the predictand (fiber diameter) and the predictors representing the solution and process parameters.

The goal is to objectively identify the most relevant experimental parameters using ML approaches. In the context of materials, it will allow us to identify optimal experimental conditions to develop PVDF fibers of desired size.

Similar studies have applied ML to develop an automatic classification system for electrospun nanofibers and polymer nano-composites with high energy density.

A standard scaler was employed before being fitted to each model, hyperparameter tuning was performed using a grid search with fivefold cross validation, and the models were developed using standard 75:25 training/validation sampling.

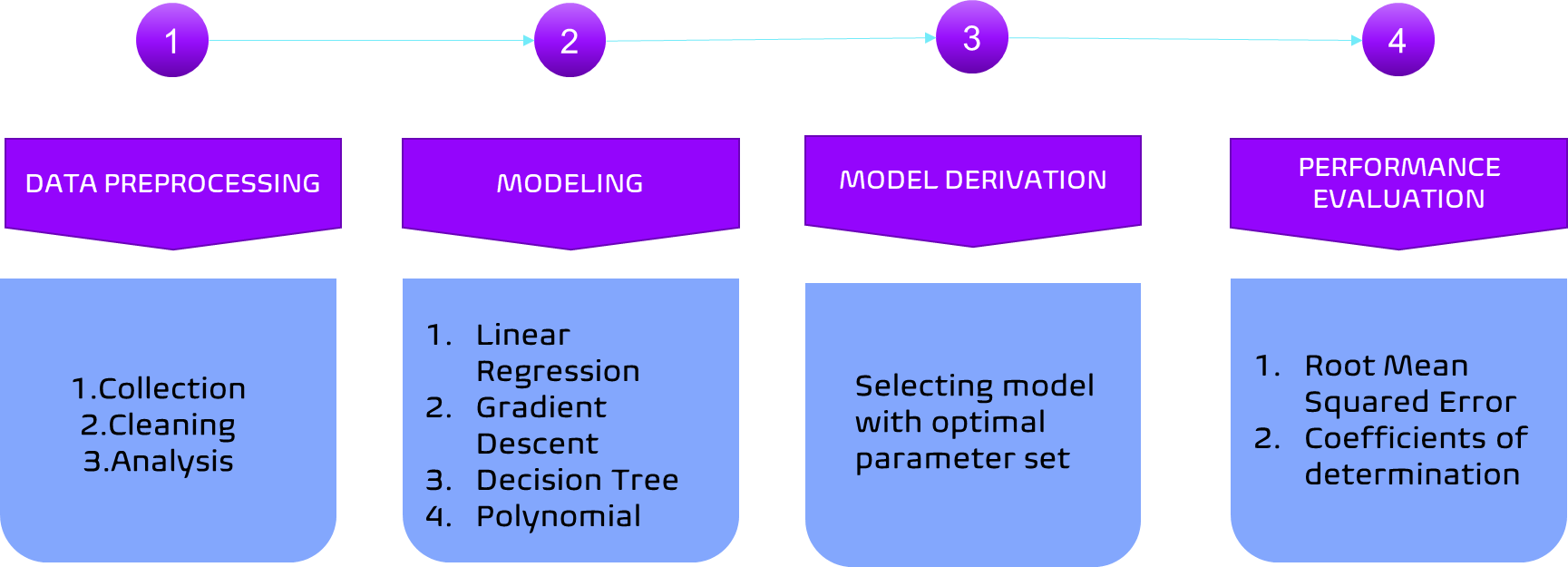
This work is to report the development of a novel dataset of electrospun PVDF fiber experimental properties, first-cut demonstration of the potential of such a dataset by modeling fiber diameter using a multi-model machine learning approach, and interpretable machine learning techniques to interpret structure property relationships learnt by the model.

Eight different models were developed using Python libraries such as numpy, pandas, and scikit-learn. To get control for obtaining targeted PVDF fiber size and lessen the requirement for carrying out repeated experiments for tests with multiple factors for parameter optimization.

‘

**CHAPTER 3**

**METHODOLOGY**

****

* Data preprocessing refers to the process of preparing and cleaning raw data to make it usable for analysis. It involves various techniques such as removing duplicates, handling missing values, scaling, normalization, and transforming data into a suitable format for analysis. The goal of data preprocessing is to improve data quality, eliminate inconsistencies, and ensure that the data is suitable for the analysis task at hand.
* The model is then used to make predictions or decisions about new data.This process typically involves selecting an appropriate algorithm, tuning its parameters, and evaluating its performance on a training dataset before testing it on a separate validation or test dataset.
* We use Performance evaluation for assessing how well a model is able to make accurate predictions on new, unseen data. The goal is to determine the model's effectiveness and identify areas where it may need improvement.

**CHAPTER 4**

**PROGRAM AND OUTPUT**

import pandas as pd

import numpy as np

import streamlit as st

#import matplotlib.pyplot as plt

#import tensorflow as tf

#from tensorflow import keras

from sklearn.model\_selection import train\_test\_split

#from keras.models import Sequential

#from keras.layers import Dense

#from tensorflow.keras.optimizers import Adam

from sklearn.preprocessing import MinMaxScaler,StandardScaler

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression,Lasso

from sklearn.preprocessing import PolynomialFeatures

from sklearn.metrics import r2\_score

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import GradientBoostingRegressor

from sklearn.ensemble import RandomForestRegressor

from sklearn.ensemble import AdaBoostRegressor

import xgboost as xgb

st.write("## Lab Data for fiber diameter")

df=pd.read\_excel("fiber\_data.xlsx")

st.dataframe(df, use\_container\_width=True)

df['Source'] = df['Source'].fillna(method='ffill')

#df=df.iloc[:,[0,5,6,7,8,9,10]]

df=df.iloc[:,[5,6,7,8,9,10,11]]

x=df.iloc[:,:-1]

x=x.replace('-',0)

x['Voltage (kV)'] = x['Voltage (kV)'].astype(float)

x['Distance (cm)'] = x['Distance (cm)'].astype(float)

x['Feed (mL/h)'] = x['Feed (mL/h)'].astype(float)

y=df.iloc[:,-1]

y=y.replace('-',0)

mm=MinMaxScaler()

x\_norm=mm.fit\_transform(x)

X\_train,X\_test,Y\_train,Y\_test= train\_test\_split(x,y,test\_size=0.2,random\_state=0)

**#Linear Regression**

lr=LinearRegression()

#lasso=Lasso(alpha=0.1)

#poly=PolynomialFeatures(degree=2)

#x\_poly=poly.fit\_transform(x\_norm)

#X\_train,X\_test,Y\_train,Y\_test= train\_test\_split(x,y,test\_size=0.2,random\_state=0)

lr.fit(X\_train,Y\_train)

#lasso.fit(X\_train,Y\_train)

y\_predict=lr.predict(X\_test)

score=r2\_score(Y\_test,y\_predict)

fiber = pd.concat([x, y], axis=1)

**#DecisionTree**

decision\_tree=DecisionTreeRegressor(max\_depth=3)

decision\_tree.fit(X\_train,Y\_train)

dt\_pred=decision\_tree.predict(X\_test)

accuracy = r2\_score(Y\_test, dt\_pred)

**#RandomForest**

rf=RandomForestRegressor(max\_depth=3)

rf.fit(X\_train,Y\_train)

rf\_pred=rf.predict(X\_test)

print(rf\_pred)

rf\_mse=mean\_squared\_error(Y\_test, rf\_pred)

rf\_rmse=(rf\_mse\*\*0.5)

print(rf\_rmse)

rf\_accuracy = r2\_score(Y\_test,rf\_pred)

**#Adaboost**

ar=AdaBoostRegressor(base\_estimator=DecisionTreeRegressor(max\_depth=6),n\_estimators=50,learning\_rate=0.1)

ar.fit(X\_train,Y\_train)

ar\_pred=ar.predict(X\_test)

print(ar\_pred)

ar\_mse=mean\_squared\_error(Y\_test, ar\_pred)

ar\_rmse=(ar\_mse\*\*0.5)

print(ar\_rmse)

ar\_accuracy = r2\_score(Y\_test, ar\_pred)

**#XgbBoost**

xgb=xgb.XGBRegressor(learning\_rate=0.1,max\_depth=6,n\_estimators=120)

xgb.fit(X\_train,Y\_train)

xgb\_pred=xgb.predict(X\_test)

print(xgb\_pred)

xgb\_mse=mean\_squared\_error(Y\_test, xgb\_pred)

xgb\_rmse=(xgb\_mse\*\*0.5)

print(xgb\_rmse)

xgb\_accuracy = r2\_score(Y\_test, xgb\_pred)

**#GradientBooster**

gbm = GradientBoostingRegressor(n\_estimators=150, max\_depth=5, learning\_rate=0.1, random\_state=0)

gbm.fit(X\_train, Y\_train)

gbm\_pred = gbm.predict(X\_test)

gbm\_acc = r2\_score(Y\_test, gbm\_pred)

print(gbm\_acc)

poly = st.number\_input('Polymer Concentration (wt%) 8–30.3', 3.00, 32.00, 15.00)

red = st.number\_input('RED 0.0751–0.49', 0.0600, 0.4000, 0.3544)

volt = st.number\_input('Voltage (kV) 5-70', 0.0, 80.0, 13.0)

dist = st.number\_input('Distance (cm) 3 – 27.7', 3.0, 30.0, 15.0)

feed = st.number\_input('Feed(mL/h) 0.06 – 14.21', 0.00, 16.00, 14.21)

flory\_x = st.number\_input('Flory-Huggins X parameter 0.004 – 0.1651', 0.0040, 0.3000, 0.0887)

new\_data=[poly,red,volt,dist,feed,flory\_x]

def predict\_data(x):

x=(np.array(x)).reshape(1,6)

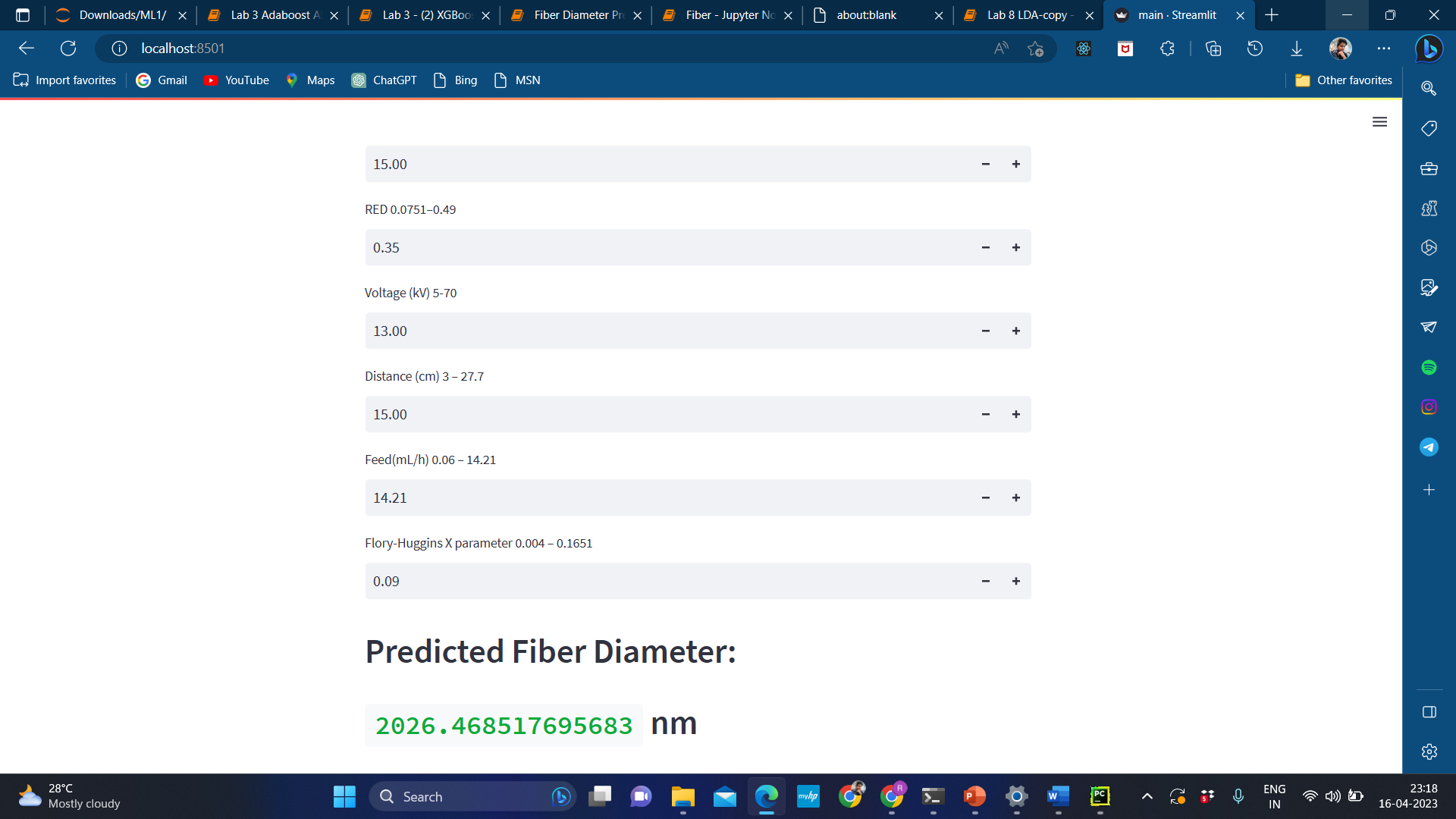
gbm\_pred = gbm.predict(x)

st.write("## Predicted Fiber Diameter:","\n ##",\*gbm\_pred.tolist(),"nm")

print(gbm\_pred)

predict\_data(new\_data)

**Ouput:**



**CHAPTER 4**

**RESULTS AND DISCUSSIONS**

A model should be generalized to the point where it can learn from the complete dataset and explain the variance while reducing overfitting. We evaluated the efficacy of commonly used machine learning and statistical models in estimating fiber diameter in order to identify an appropriate model. Models were trained and evaluated using two traditional model performance measures: the coefficient of determination (R2) and the Root Mean Square Error (RMSE).

**Linear Regression:**

Linear regression approaches have found usage in modeling polymer nanocomposites . Since this is a multivariate problem, we haveused the multiple linear regression (MLR)-based approach, which forms a regression matrix to construct the set of simultaneous equations and then uses the least square fit to estimate the unknown coefficients.

**Decision tree regression (DTR):**

DTR is a non-parametric supervised learning method useful for both classification and regression tasks. They are constructed by algorithmically identifying ways to split the dataset based on different criteria. Thebasic concept of a decision tree is to split a complex decision into multiple smaller decisions through binary recursive partitioning and minimizing the sum of the squared deviations from the mean in the separate parts. It has several advantages such as high model interpretability and

flexibility for a broad range of response types such as categorical and numerical data. Gradient Boosting Regressors (GBR) are regressor models based on ensemble decision trees with the advantage of high predictive power and robustness towards outliers in output space.

**Gradient Boosting Regressor (GBR):**

GBR is a generalized form of gradient boosting and involves a lossfunction, a weak learner, and an additive model. Its main advantage is its scalability due to the sequential nature of boosting.

**CHAPTER 5**

**CONCLUSION**

Using interpretable ML models we demonstrated the multivariate relationships between fiber diameter, a keycharacteristic of electrospun fibers, and a large number of solution and process experimental parameters were successfully modeled using a multi-model machine learning approach. The best model was chosen based on an objective selection criterion and the relative significance of the influencing factors were established using SHAP, a game theoretic approach to explain the output of any machine learning model. Parameters such as feed, polymer concentration, and relative energy difference are confirmed to have a higher and proportionally increasing impact on fiber diameter. While distance and voltage were observed to have a negative correlation with fiber diameter. A limitation of this dataset is its small sample size and non-inclusion of failed experiments due to general unavailability of such materials datasets in public. However, we are optimistic that the methodology that we have proposed will spur a discussion in the community on the need for such datasets and robust computational studies. An additional goal of this paper is to reach out to other research groups and companies that

maintain datasets related to PVDF-based fibers and how we can collaborate to develop an increasingly valuable resource that is openly and readily available to the scientific community. Currently, the database is collated and enhanced using experimental results reported in literature. But the publications often don’t report the full spectrum of experimental conditions. Data on failed experiments in the context of fiber formation also remain unreleased and under-utilized in computational studies. There is an opportunity here for the community to develop standards for communicating experimental results and develop community databases that would spur computational studies in the area

and greatly ameliorate the small sample problem that often plague materials datasets. Based on such a community dataset, we can develop robust prognostic models generalizable over diverse experimental conditions and aid experimentalists to reduce the cost of conducting experiments and develop better understanding of these complex interactions.