CHAPTER 5

A MATHEMATICAL MODEL FOR PREDICTION OF TENSILE BEHAVIOR

CHAPTER 5
A MATHEMATICAL MODEL FOR PREDICTION OF TENSILE BEHAVIOR

5.0 Introduction

Braided ropes find widespread use in various industries, including marine, construction, and sports, where strength and durability are essential. The strength of a braided rope depends on multiple factors, including the strength of the individual yarns and the braiding pattern. However, conducting numerous experiments to determine the strength of ropes with different configurations is time-consuming and resource-intensive. Therefore, developing a mathematical model to predict rope strength based on key parameters is highly desirable. In this study, we propose a mathematical model using curve fitting techniques to predict the strength of braided ropes.

Previous studies have explored various approaches to predict the strength of ropes and fibers. Some researchers have focused on empirical models based on experimental data, while others have employed mathematical modeling techniques. Curve fitting methods, such as polynomial regression and least squares, have been widely used to establish relationships between different variables. However, limited research exists specifically on predicting the strength of braided ropes using mathematical models. This study aims to fill this gap by developing a robust and accurate model for predicting rope strength.

5.1 Mathematical Model

The strength of the braided rope depends upon the strength of the single yarn as well as the number of yarns that are braided. Finding the strength of braided rope with ‘N’ number of yarns becomes challenging in terms of performing more experiments, which will result in loss of time, material, energy, etc. So, to predict the strength of the braided rope, it was suggested to have a mathematical model which will predict the rope's strength by inputting parameters like single yarn strength and the number of such yarns to be braided. Looking at the problem statement, it was suggested to use the ‘curve fitting’ method to solve it.

5.2 Curve Fitting

A curve fitting is a relationship which is found to exist between two or more variables. e.g., There exists a relationship between the following pair of observations:

(i) The height and the weight of a person.
(ii) Income and expenditure of a family.
(iii)Rainfall and the yield crops.
(iv)Price and demand of an article.

Sometimes it is necessary to express this relationship in mathematical form by an equation involving the variables. The observations of two such variables are plotted graphically in a rectangular coordinate system and then we draw a smooth wave approximating the given data. The process of finding such a curve or its equation which is the best fitting to the given data is called curve fitting.

5.3 Types of Curve Fitting Method

- **Linear Regression**

Suitable for fitting linear relationships between variables. Assumes a linear relationship between the independent and dependent variables.

- **Polynomial Regression**

Fits a polynomial function to the data. Can capture nonlinear relationships between variables by using higher-degree polynomials.

- **Exponential Regression**

Fits an exponential function to the data. Useful for modeling growth or decay processes.

- **Power Regression**

Fits a power function to the data. Suitable for modelling relationships where one variable is a power of another.

- **Logarithmic Regression**

Fits a logarithmic function to the data. Useful for modelling relationships where the dependent variable changes at a decreasing rate.

- **Spline Interpolation**

Divides the data into segments and fits a polynomial function to each segment. Provides

a smooth curve that passes through each data point.

5.4 Principle of Least Squares

The method of least square is a process of finding the equation of a curve which is the best fitting to the given data.

Let (x_i , y_i), i=1,2, 3,...,n be n pairs of observations of the variables x & y of a given data. Let curve c: y=f(x) be the best fitted curve to the data. We plot n pairs of observations (x_i , y_i) in a rectangular co-ordinate system. We get n points P_i (x_i , y_i), i=1,2,…,n. Some of these points P_i may lies above, below or on the curve C.

For a given value of , say $x = x_i$, we get two values of y i.e., one value y_i which is the observed value of y_i and the other value y_i which is the expected value of y. The value y_i is obtained from the equation : $Y = f(x)$ for $x = x_i$. The difference $E_i = Y_i - y$ is called error (or residual deviation) in y at point $x = x$. This error E_i can be positive, negative or zero according as the point P_i is above below or on the curve C. From the sum

$$S = E_1^2 + E_2^2 + \cdots + E_n^2 = \sum_{i=1}^{n} E_i^2$$

The measure of the value of S is called the **goodness of fitting of a curve** C to the given data. S will be different for different curves. If the value of S is decreasing, then the curve C approaches to the given data. By the principle of Least Square, the best fitted curve out of all these curves approximating the given data is that curves for which S is minimum and the curve obtained in this way is called the least square curve or the best fitted curve.

- **Fitting a straight line = $ax + b$**

Let $y = ax + b$ be the straight line to be fitted to the given set of data points (x_1, y_1), (x_2, y_2),..., (x_n, y_n)

$$e_i = y_i - (x_i) = y_i - (ax_i + b)$$

$$\therefore E = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(y_i - f(ax_i + b)\right)^2$$

Now by the principle of least square, for the curve of best fit, E is minimum.

$$\therefore \frac{\partial E}{\partial a} = 0, \frac{\partial E}{\partial b} = 0$$

That implies,

$$\sum_{i=1}^{n} y_i = a \sum_{i=1}^{n} x_i + nb \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$\sum_{i=1}^{n} x_i y_i = a \sum_{i=1}^{n} {x_i}^2 + b \sum_{i=1}^{n} x_i \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Using these two Equations we can find constants and fit the linear curve $y = ax + b$.

Similarly, we can find the constants of the higher degree polynomial such that the error (Observed data and the predicted data) would be less and fit nonlinear curve. The proposed methodology involves collecting experimental data on the strength of braided ropes with different configurations, including varying numbers of yarns and yarn strengths. The principle of least squares is then applied to fit mathematical curves to the data, minimizing the error between observed and predicted values. Both linear and non-linear curve fitting techniques are explored to determine the best-fitted model for predicting rope strength. The constants of the mathematical equations are calculated to enable accurate predictions.

5.5 Curve Optimization

Data: *y*i: Maximum Force (predicted) and i: Number of yarn.

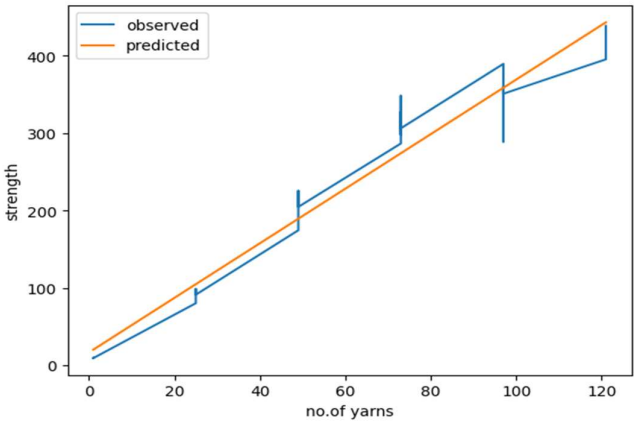
```
X=[1,1,1,1,1,25,25,25,25,25,49,49,49,49,49,73,73,73,73,73
,97,97,97,97,97, 121,121,121]

Y=[9.326,9.2663,9.1977,9.3428,9.2154,80.1597,94.8652,98.5
508,93.4778,91.1795,174.373,202.975,204.361,225.216,204.5
69,286.455,326.997,297.881,347.97,306.05,389.315,348.886,
288.735,360.326,350.727, 395.171,408.189,438.338]
```

Checking error for the linear curve:

```
import numpy as np
import matplotlib.pyplot as plt
c=np.polyfit(x,y, 1)
Y=np.polyval(c,x)
per_error= 0
for i in range (0,len(x)):
per_error+=((y[i]-Y[i])/y[i])
print ("Error",abs(per_error))
```

```
plt.plot(x,y)
plt.plot(x,Y)
plt.legend(['observed','predicted'])
plt.xlabel("no.of yarns")
plt.ylabel("strength")
```



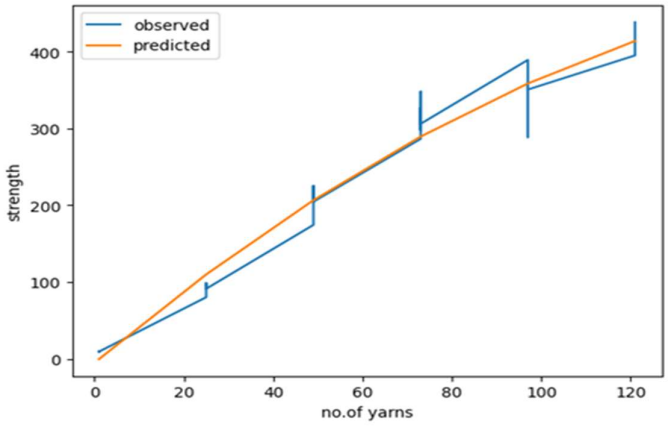
- ***Output:***

```
Error
6.135007260782372
```

Figure 5. 1 Plot of linear curve

Checking error for Quadratic curve:

```
c=np.polyfit(x,y,2)
Y=np.polyval(c,x)
per_error= 0
for i in range(0 ,len (x)):
per_error+=((y[i]-Y[i])/y[i])
print("Error",abs(per_error))
```



- ***Output:***

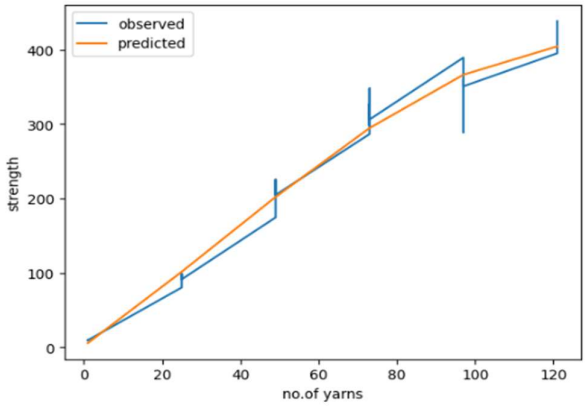
```
Error
4.272381857180427
```

Figure 5. 2 Plot of quadratic curve

Checking error for Cubic curve:

```
c=np.polyfit(x,y, 3)
Y=np.polyval(c,x)
per_error= 0
```

```
for i in range (0, len(x)):
per_error+=((y[i]-Y[i])/y[i])
print ("Error",abs(per_error))
```



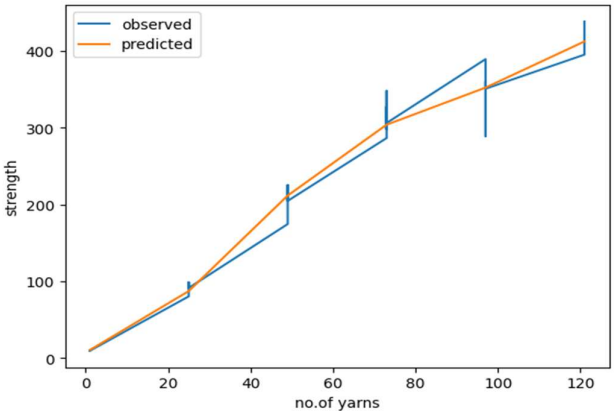
- ***Output:***

```
Error
1.498050715967315
```

Figure 5. 3 Plot of cubic curve

Checking error for 4 th degree polynomial:

```
c=np.polyfit(x,y, 4)
Y=np.polyval(c,x)
per_error= 0
for i in range (0, len (x)):
per_error+=((y[i]-Y[i])/y[i])
print ("Error",abs(per_error))
```



- ***Output:***

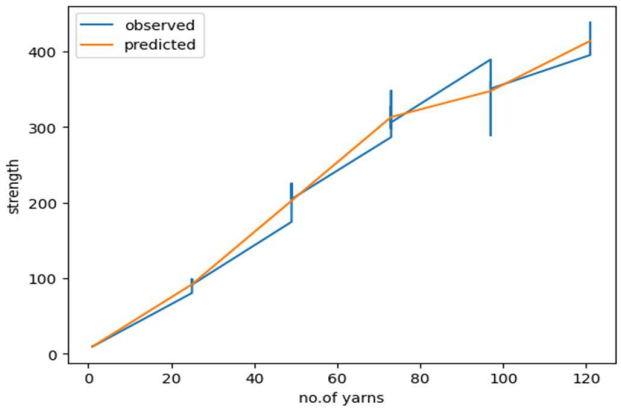
```
Error
0.518525998442550
1
```

Figure 5. 4 Plot of bi-quadratic curve

Checking error for 5 th degree polynomial:

```
c=np.polyfit(x,y, 5)
Y=np.polyval(c,x)
per_error= 0
for i in range(0,len (x)):
per_error+=((y[i]-Y[i])/y[i])
```

```
print("Error" ,abs(per_error))
```



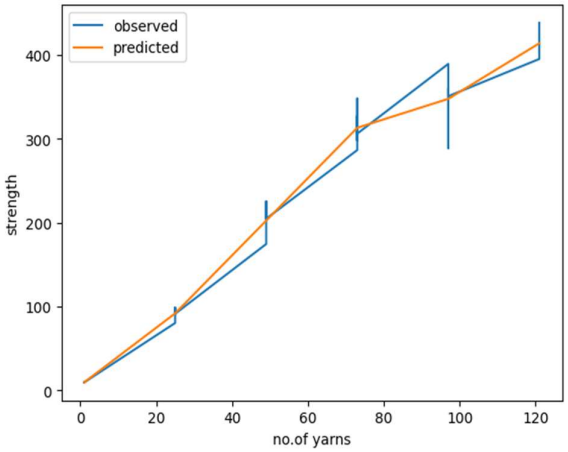
- ***Output:***

```
Error
0.13727735205851255
```

Figure 5. 5 Plot of quintic curve

Checking error for 6 th degree polynomial:

```
c=np.polyfit(x,y, 6)
Y=np.polyval(c,x)
per_error= 0
for i in range (0,len (x)):
per_error+=((y[i]-Y[i])/y[i])
print ("Error",abs(per_error))
```



- ***Output:***

```
Error
0.137277352058733
```

Figure 5. 6 Plot of hexic curve

For this data, Due to comparatively less error, we have fitted 5*th* degree non-linear curve i.e. 5*th* degree polynomial, where:

x: Number of yarn

y: Maximum force

Let $y = ax^5 + bx^4 + cx^3 + dx^2 + ex +$ be the straight line to be fitted to the given set of data points $(x1, y1)$, $(x2, y2)$,…., (xn, yn).

Where y_i : Maximum Force

x_i : Number of yarn

$e_i = y_i - (x_i)$

$= y_i - (ax^5 + bx^4 + cx^3 + dx^2 + ex + f)$

$$\therefore E = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} ((y_i) - (ax^5 + bx^4 + cx^3 + dx^2 + ex + f))^2$$

Now by the principle of least square, for the curve of best fit, E is minimum.

$$\therefore \frac{\partial E}{\partial a} = 0, \frac{\partial E}{\partial b} = 0, \frac{\partial E}{\partial c} = 0, \frac{\partial E}{\partial d} = 0, \frac{\partial E}{\partial e} = 0, \frac{\partial E}{\partial f} = 0$$

Solving these five equations we will get the value of constants a, b, c, d, e and f . Then we can fit the 5th-degree polynomial.

i.e., $y = ax^5 + bx^4 + cx^3 + dx^2 + ex + f$

Python Program:

```
import numpy as np
import matplotlib.pyplot as plt
x=[1,1,1,1,1,25,25,25,25,25,49,49,49,49,
49,73,73,73,73,73,97,97,97,97,97, 121,121,121]
y = [9.326,9.2663,9.1977,9.3428,9.2154,
80.1597,94.8652,98.5508,93.4778,91.1795,174.373,
202.975,204.361,225.216,204.569,286.455,326.997,
297.881,347.97,306.05,389.315,348.886,288.735,360.326,3
50.727,395.171,408.189,438.338]
c=np.polyfit(x,y, 5)
print('The values of coefficients a,b,c,d & f are\n',c)
Y=np.polyval(c,x)
print(' The values of y using obtained values of
a,b,c,d & f \n',Y)
plt.plot(x,Y)
plt.plot(x,y)
plt.legend(['observed','predicted'])
plt.xlabel("no.of yarns")
plt.ylabel("strength")
```

```
n=int(input("Enter the number of yarn :"))
Y=c[0]*n**5+c[1]*n**4+c[2]*n**3+c[3]*n**2+c[4]*n+c[5]
print("Maximum force will be ",eval(str(Y),{"n":n}))
```

Output:

```
The values of coefficients a, b, c, d & f are
[ 2.43420273e-07, -6.56923229e-05, 5.70006145e-03,-
1.74335818e-01, 5.22452489e+00, 4.21381631e+00]
The values of y using obtained values of a, b, c, d & f
[9.26964, 9.26964,9.26964,9.26964, 9.26964, 91.6466,
91.6466,91.6466, 91.6466, 91.6466, 202.2988,202.2988,
202.2988, 202.2988, 202.2988, 313.0706, 313.0706,
313.0706, 313.0706, 313.0706, 347.5978, 347.5978,
347.5978, 347.5978, 347.5978, 413.89933333,413.89933333
,413.89933333]
Enter the number of yarn :120
Maximum force will be  405.54247396555365
```

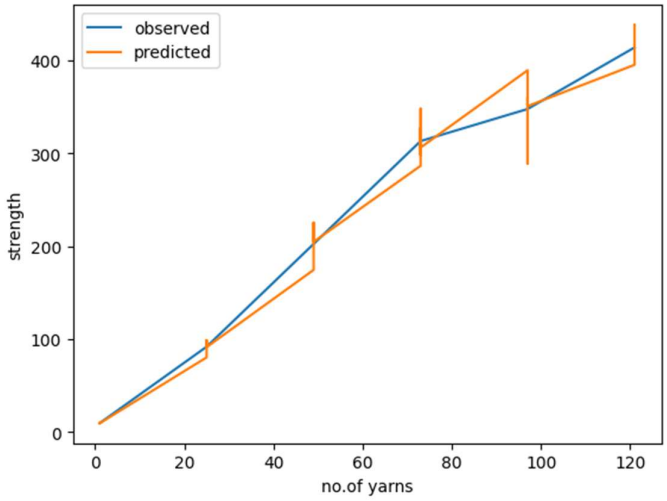


Figure 5. 7 Best fitted curve

Summary:

The real experimental data for the strength of single yarn as well as data for the strength of braided rope made from 25, 49, 73, 97, and 121 yarns were provided. The curve fitting method was used to solve this problem and a mathematical model was derived. The data were fitted in a five-degree polynomial,

$y = 2.43420273 * 10^{-7} x^5 - 6.56923229 * 10^{-5} x^4 + 5.70006145 * 10^{-3} x^3 + 0.174338 x^2 + 5.22452489\, x + 4.2138163$ and it was found that predicted data and observed data are fitting well with an error of 0.13727

5.6 Investigation of Strength Prediction Models for Braided Ropes Made from Different Materials

Braided ropes are essential in various industries, from maritime to construction, where their strength determines safety and reliability. Predicting their strength is crucial for selecting the right material and construction method to ensure optimal performance in different applications.

Table 5.1 Data of different materials of braided ropes

Layer	Single	Layer-1	Layer-2	Layer-3	Layer-4	Layer-5
No. Of Yarns	yarn	25	49	73	97	121
Material			Strength in kgf			
Nylon	25.532	657.8194	1395.374	1073.109	11528.45	1259.159
Polypropylene	9.26964	513.2984	627.9185	1051.514	742.5385	822.2742
Basalt	33.0818	508.315	601.3399	900.3487	973.4398	1073.109
Nylon +Basalt		523.2654	593.0341	694.3649	867.1255	1003.341
Polypropylene +Basalt		621.2738	6312408	694.3649	863.8032	847.1916

Here we find the strength of rope which made form different type of material such as Nylon, PP, Basalt, etc. with different number of yarns by using curve fitting. In the braided rope made from a mixture of nylon and basalt fibers, the composition consists of an equal ratio of 50% nylon and 50% basalt. In the braided rope composed of a blend of polypropylene (PP) and basalt fibers, the ratio is 35% polypropylene and 65% basalt.

5.6.1 Curve Fitting Analysis for Tensile Strength of Braided Rope

5.6.1.1 Finding strength of nylon rope

```
import numpy as np
import matplotlib.pyplot as plt
x = [1,25,49,73,97,121]
y = [25.532,657.819,1395.3744,1073.10936,1152.84504,
     1259.15928]
n=np.polyfit(x,y, 5)
Y=np.polyval(n,x)
print(" Observed values of strength \n",y)
print("Predicted values of strength \n",Y)
M=np.poly1d(n)
print ("The best curve fitted equation is \n",M)
per_error= 0
```

```
for i in range ( 0 , len (x)):
   per_error+=abs((y[i]-Y[i])/y[i])
print ( "Error" ,(per_error))
print ( "i.e. Error=" ,np.round((per_error)))
s=int(input("Enter the number of yarn :"))
op=n[0]*s**5+n[1]*s**4+n[2]*s**3+n[3]*s**2+n[4]*s+n[5]
print("Maximum force will be ",op," Kgf")
plt.scatter(x,y)
plt.plot(x,y,ls='-')
plt.plot(x,Y,ls='-.')
plt.xlabel("no.of yarns")
plt.ylabel("strength")
plt.legend(['observed','predicted'])
plt.title('Relation between no.of yarns & strength of
NYLON')
plt.show()
```

Output:

```
Observed values of strength
[25.532, 657.819, 1395.3744, 1073.10936, 1152.84504,
1259.15928]
Predicted values of strength
[25.532    657.819   1395.3744  1073.10936 1152.84504
1259.15928]
The best curve fitted equation is
-4.672e-06 x^5 +0.001475 x^4 -0.1616 x^3 +6.898 x^2 -
69.91 x + 88.71
Error 1.0120734286452277e-12
i.e. Error= 0.0
Enter the number of yarn :27
Maximum force will be  765.9017279983055  Kgf
```

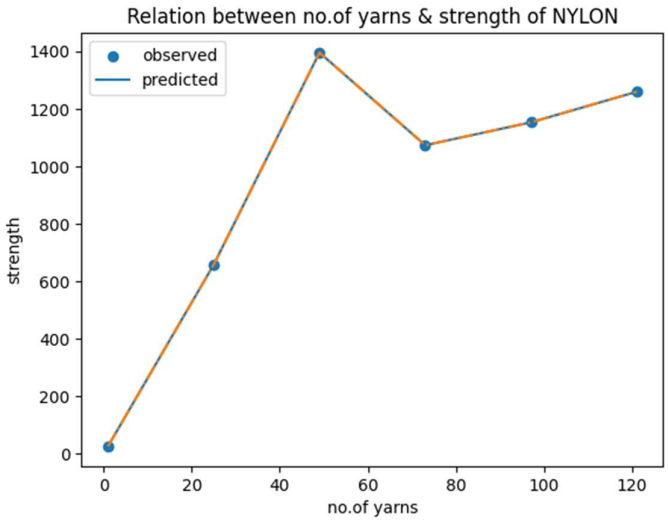


Figure 5. 8 Best fitted curve for nylon rope

In the same manner with the help of curve fitting analysis tensile strength was analyzed for all other materials.

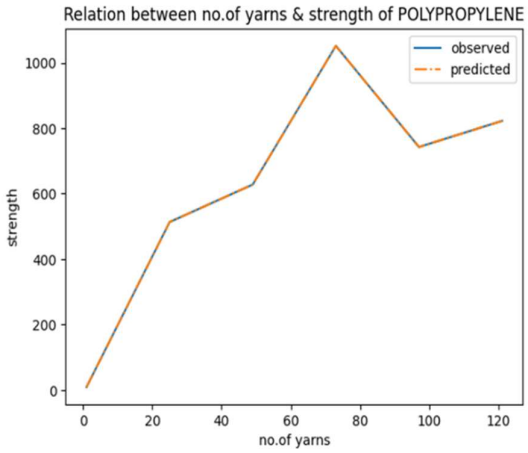


Figure 5. 9 Best fitted curve for polypropylene rope

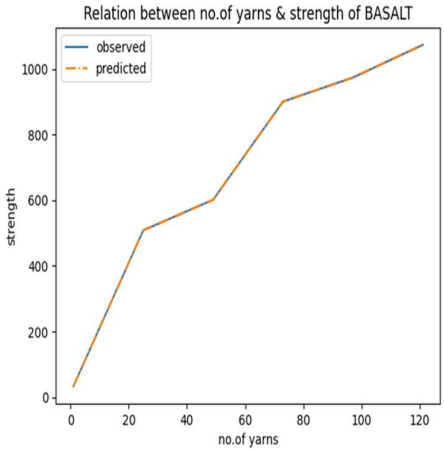


Figure 5. 10 Best fitted curve for basalt rope

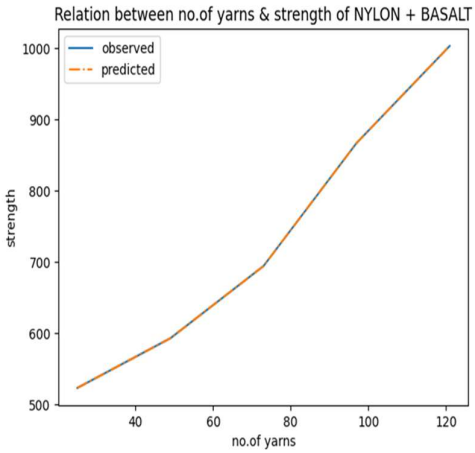


Figure 5. 11 Best fitted curve for nylon + basalt rope

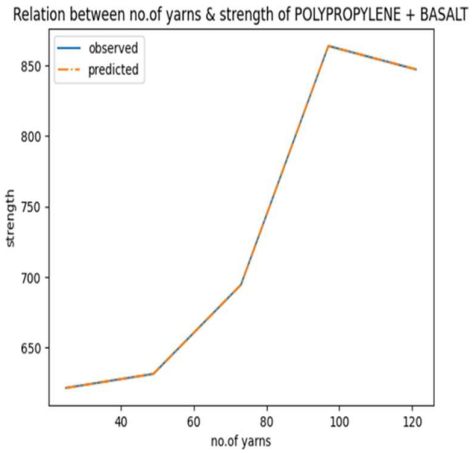


Figure 5. 12 Best fitted curve for polypropylene + basalt rope

5.6.2 Reverse Curve Fitting

5.6.2.1 Reversed data analysis

After fitting the curve with X as the independent variable and Y as the dependent variable, a reversal of roles was conducted to explore the relationship when Y becomes the independent variable and X becomes the dependent variable. This reversal aims to investigate how variations in tensile strength (Y) impact the number of yarns (X) in braided ropes.

5.6.2.2 Methodology for reversed curve fitting

To fit the curve for the reversed dataset, the same methodology used for the initial analysis was employed. However, adjustments were made to account for the reversed roles of X and Y. The dataset was rearranged accordingly, with Y values now treated as the independent variable and X values as the dependent variable.

5.6.2.3 Reverse curve fitting for nylon

```
import numpy as np
import matplotlib.pyplot as plt
x = [25.532,657.819,1395.3744,1073.10936,1152.84504,
     1259.15928]
y = [1,25,49,73,97,121]
n=np.polyfit(x,y, 5)
Y=np.polyval(n,x)
print(" Observed values of no. of yarns \n",y)
print("Predicted values of no. of yarns \n",Y)
M=np.poly1d(n)
print ("The best curve fitted equation is \n",M)
per_error= 0
for i in range ( 0 , len (x)):
   per_error+=abs((y[i]-Y[i])/y[i])
print ( "Error" ,(per_error))
print ( "i.e. Error=" ,np.round((per_error)))
s=float(input("Enter the strength in Kgf :"))
op=n[0]*s**5+n[1]*s**4+n[2]*s**3+n[3]*s**2+n[4]*s+n[5]
print("Minimum number of yarns ",(round(op)))
plt.plot(x,y,ls='-')
plt.plot(x,Y,ls='-.')
plt.xlabel("no.of yarns")
plt.ylabel("strength")
plt.legend(['observed','predicted'])
plt.title(' Relation between no.of yarns & strength of
NYLON')
plt.show()
```

Output:

```
 Observed values of no. of yarns
 [1, 25, 49, 73, 97, 121]
Predicted values of no. of yarns
 [ 1.  25.  49.  73.  97. 121.]
The best curve fitted equation is

-6.104e-12 x^5+2.418e-08 x^4-3.535e-05 x^3+0.02281 x^2-
5.606 +129.9
Error 7.811176991260006e-12
i.e. Error= 0.0
Enter the strength in Kgf :650
Minimum number of yarns  22
```

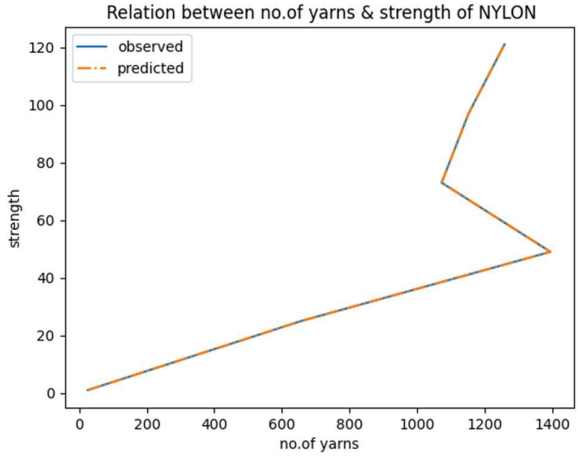


Figure 5. 13 Reverse curve for nylon rope

In the same manner with the help of Reverse curve fitting analysis tensile strength was analyzed for all other materials.

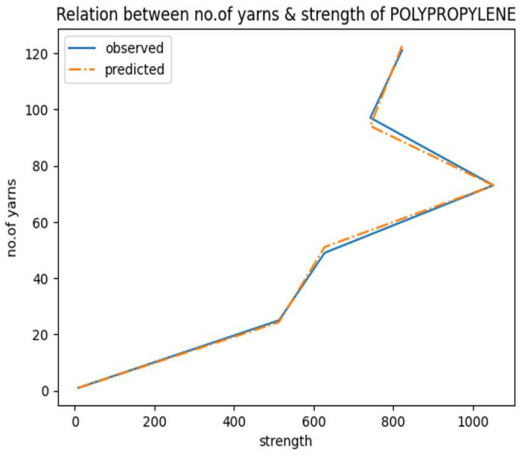


Figure 5. 14 Reverse curve for PP rope

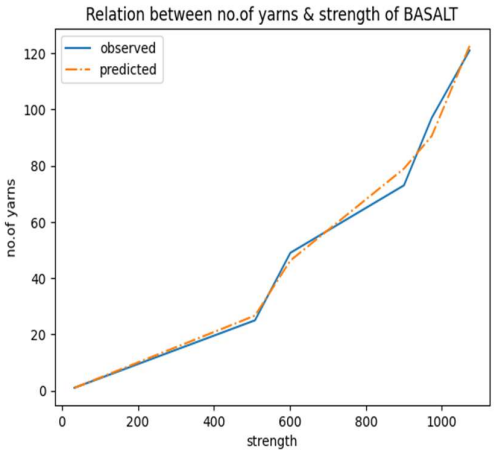


Figure 5. 15 Best fitted curve for basalt rope

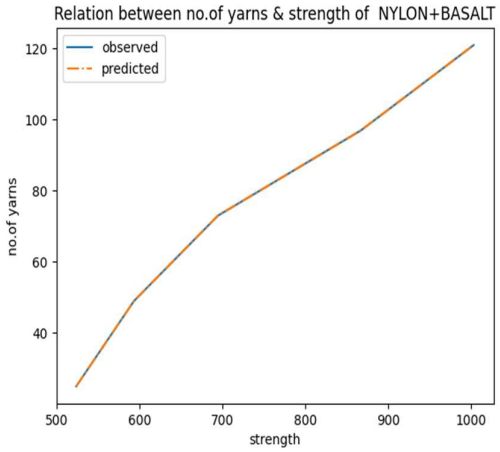


Figure 5. 16 Best fitted curve for nylon + basalt Rope

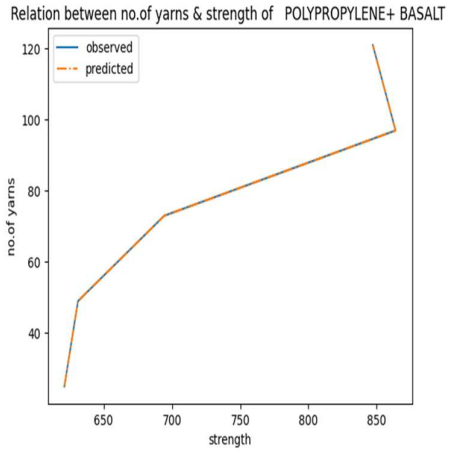


Figure 5. 17 Best fitted curve for polypropylene + basalt rope

5.6.3 Comparative Analysis

A comparative analysis was conducted on the curve fitting results for braided ropes made of different materials. This analysis aimed to elucidate the relationship between the number of yarns (X) and the tensile strength (Y) of these ropes.

5.6.3.1 Comparison of nylon, basalt & polypropylene rope

```
import matplotlib.pyplot as plt
x = [1,25,49,73,97,121]
ny =
[25.532,657.819,1395.3744,1073.10936,1152.84504,1259.1592
8]
by =
[33.0818,508.31496,601.33992,900.34872,973.43976,1073.109
4]
ppy=
[9.26964,513.29844,627.91848,1051.51428,742.53852,822.274
2]
plt.plot(x,ny,ls='solid')
plt.plot(x,by,ls='-.')
plt.plot(x,ppy,ls='--')
plt.legend(['Nylon','Basalt','Polypropylene'])
plt.title("Comparison of Nylon ,Basalt , Polypropylene
rope")
plt.xlabel('no.of yarns')
plt.ylabel('strength (Kgf)')
```

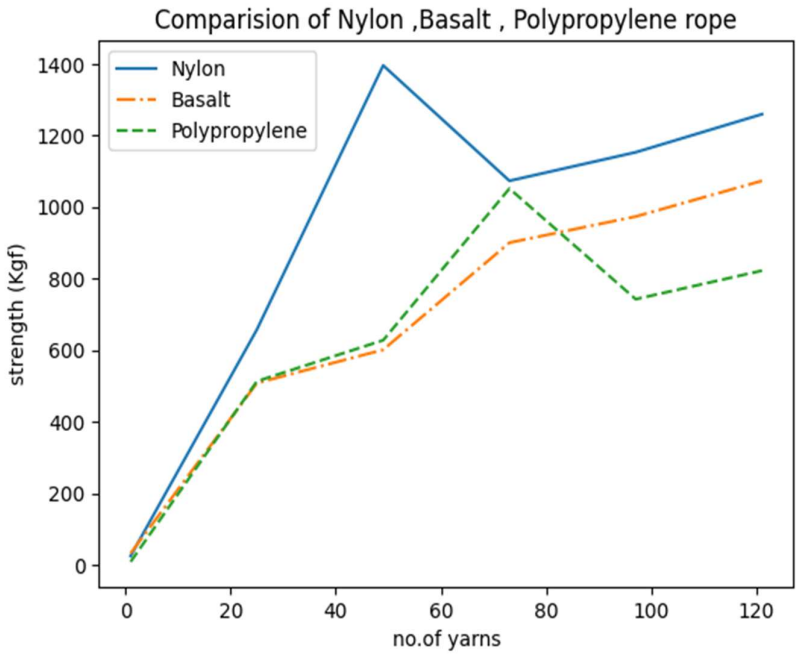


Figure 5. 18 Comparison of nylon, basalt, polypropylene rope

Based on the tensile strength data presented, the nylon rope demonstrates the highest maximum tensile strength. Therefore, for applications requiring high tensile strength, the nylon rope would be the preferred choice among the three materials studied.

5.6.3.2 Comparison of nylon, basalt and mix rope

```
import matplotlib.pyplot as plt
x = [1,25,49,73,97,121]
ny =
[25.532,657.819,1395.3744,1073.10936,1152.84504,1259.1592
8]
by =
[33.0818,508.31496,601.33992,900.34872,973.43976,1073.109
4]
nby =
[30,523.2654,593.03412,694.36488,867.12552,1003.34064]
plt.plot(x,ny,ls='solid')
plt.plot(x,by,ls='-.')
plt.plot(x,nby,ls='--')
plt.legend(['Nylon','Basalt','mix '])
plt.title("Comparison of Nylon ,Basalt , mix rope")

plt.xlabel('no.of yarns')
plt.ylabel('strength (Kgf)')
```

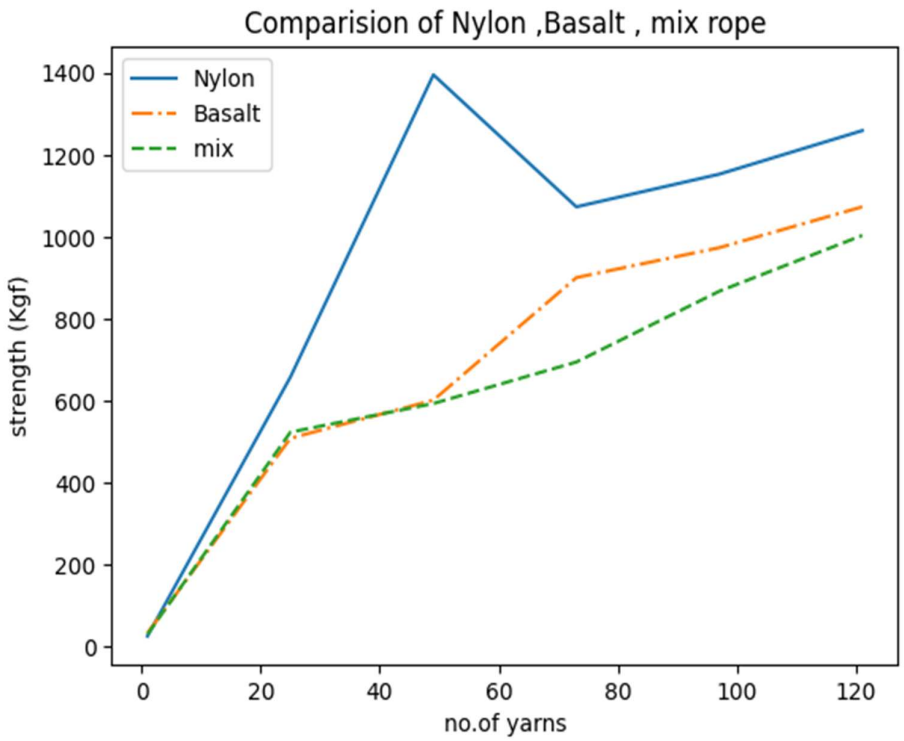


Figure 5. 19 Comparison of nylon, basalt, mix rope

Nylon ropes stand out for their exceptional strength and toughness in all tests. They are chosen for their impressive strength-to-weight ratio, flexibility, and excellent resistance to wear and tear. Nylon-basalt mix ropes offer a unique combination of materials. While the addition of basalt fibers doesn't significantly enhance the strength, it provides a balanced option that leverages the properties of both materials. Basalt ropes showcase the inherent toughness of basalt fibers. Although they are not as strong as nylon or the nylon-basalt mix, their potential lies in the unique properties of basalt, which could be further optimized with improved manufacturing techniques. Overall, nylon ropes emerge as the strongest, but the study underscores the importance of selecting the right materials for specific applications. There is potential to enhance the performance of basalt fibers and explore better manufacturing methods to fully utilize their strength. Additionally, considering factors like flexibility, wear resistance, and environmental conditions is crucial when choosing ropes for various tasks. This comprehensive approach ensures that the chosen ropes meet the specific demands of different applications effectively.

5.6.3.3 Comparison of basalt, polypropylene, and mix Rope

```
import matplotlib.pyplot as plt
x = [1,25,49,73,97,121]
by =
[33.0818,508.31496,601.33992,900.34872,973.43976,1073.109
4]
py=
[9.26964,513.29844,627.91848,1051.51428,742.53852,822.274
2]
bpy = [10,621.27384,631.2408,694.36488,863.8032,847.1916]
plt.plot(x,by,ls='solid')
plt.plot(x,py,ls='-.')
plt.plot(x,bpy,ls='--')
plt.title("Comparison of Basalt ,Polypropylene,mix rope")
plt.legend(['Basalt','Polypropylene','mixture'])
plt.xlabel('no.of yarns')
plt.ylabel('strength (Kgf)')
```

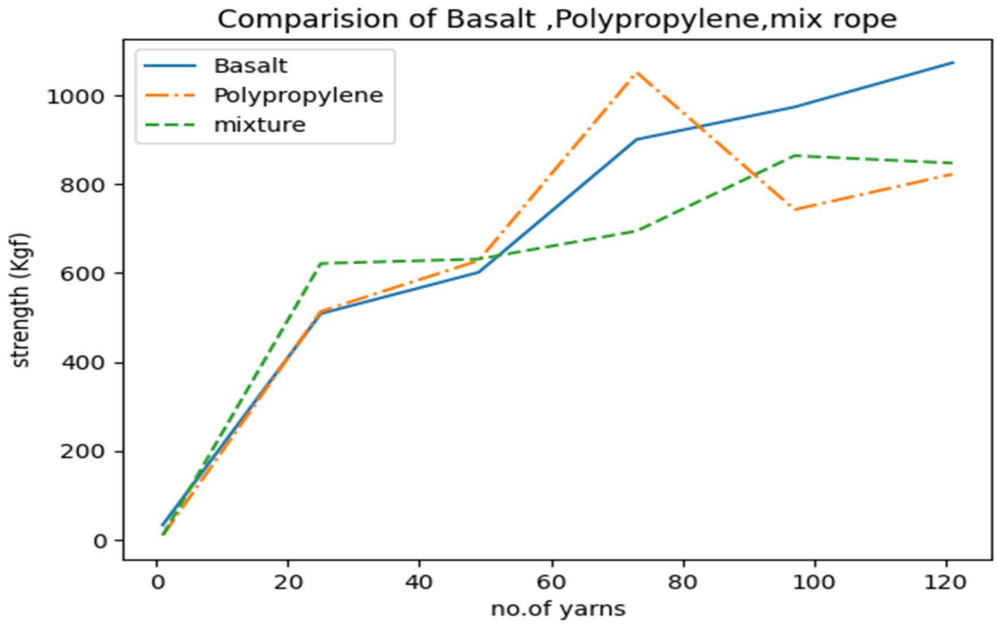


Figure 5. 20 Comparison of basalt, polypropylene, mix rope

Comparing basalt, polypropylene, and a mix of both in braided ropes gives us some important findings:

1. Basalt Ropes are Strong: Basalt ropes get stronger as more yarns are braided. Basalt fibers are known for being tough and resistant to damage from the environment, making them reliable for tough jobs.

2. Polypropylene Ropes Hold Up Well: Polypropylene ropes also get stronger with more braided yarns, but not as much as basalt. They're flexible and strong, but they might not be as tough as basalt, especially with more yarns.
3. Mixed Ropes Have Mixed Results: Combining basalt and polypropylene fibers gives mixed strength results. Depending on how the yarns are braided, the strength of the mix can vary. Adding polypropylene might make the ropes more flexible and cheaper, but it might not make them much stronger compared to basalt alone.

Overall, basalt ropes are consistently strong, making them great for tough jobs. Polypropylene ropes are flexible and strong too, but they might not be as tough as basalt. Mixing both fibers gives mixed results, and more research is needed to figure out the best mix for different jobs.

5.6.4 Importance of Curve Fitting in Material Analysis

Curve fitting serves as a valuable tool to understand the relationship between variables and predict outcomes in various materials. In the context of braided ropes, it helps to:

1. ***Sensitive to outliers:***

Identify and account for outliers that could skew the relationship between the number of yarns and tensile strength.

2. ***Assume linearity:***

Determine if a linear model adequately represents the relationship between X and Y or if a nonlinear model might be more appropriate.

3. ***Detect overfitting and underfitting:***

Ensure that the curve captures the underlying trend without being overly complex or too simple, leading to inaccurate predictions.

4. ***Optimize model parameters:***

Select the most suitable curve shape or type to accurately represent the data and improve predictive performance.

5. ***Enhance interpretability:***

Provide a clear and understandable representation of the relationship between the number of yarns and tensile strength, aiding in practical applications and design optimization.

5.7 Challenges in Curve Fitting

1. ***Sensitive to outliers:*** Even a single unusual data point can significantly impact the curve's fit and result in inaccurate predictions.
2. ***Assumes linearity*****:** The method presupposes that the relationship between the number of yarns and tensile strength is linear. If the relationship is nonlinear, the curve fitting may not accurately capture the true trend.
3. ***Overfitting*****:** Overfitting occurs when the curve closely follows the data points, including noise or random fluctuations, making it less effective for predicting new data.
4. ***Underfitting:*** Underfitting happens when the curve is too simplistic to capture the underlying trend, leading to poor predictive performance.
5. ***Choosing model details:*** The selection of appropriate model parameters, such as the curve type or degree of polynomial, is crucial for accurate curve fitting.
6. ***Limited interpretability:*** The curve may fit the data well but lack clear interpretation, particularly for complex nonlinear relationships.
7. ***Assumes equal variance:*** The method assumes that the variance of the residuals is consistent across all levels of the independent variable. Any violation of this assumption can lead to biased parameter estimates and inflated standard errors.
8. ***Multicollinearity:*** In the case of multiple independent variables, such as different materials, high correlation between them can result in unstable parameter estimates and complicate the interpretation of individual predictors.

A comprehensive analysis of the relationship between the number of yarns and tensile strength in braided ropes made of various materials was conducted using curve fitting techniques. This approach provided valuable insights into the factors influencing rope strength and allowed for a more robust prediction model. By considering different materials and their respective curve-fitting results, this study offers practical guidance for optimizing the design and performance of braided ropes in real-world applications.

5.8 Advanced Fiber-Reinforced Composites

Exploring the mathematical modelling of composite rods made from a blend of nylon, basalt, and polypropylene. Utilizing resin infusion processes, the study investigates the mathematical formulations to predict and optimize the mechanical strength, durability,

and other key properties of these hybrid materials. The research highlights their performance in various industrial and construction applications, emphasizing the importance of accurate mathematical modelling in enhancing composite material properties and applications.

5.8.1 Curve Fitting Analysis for Tensile Strength of Composite Rods

5.8.1.1 Nylon rod

```
import numpy as np
import matplotlib.pyplot as plt

x=[25,25,25,25,25,49,49,49,49,49,73,73,73,73,73,97,97,97,
97,97,121,121,121,121,121]

y=[192.484,176.338,194.113,187.645,190.0645,311.82408,285
.66756,314.46306,303.9849,307.90449,411.6077856,415.09123
92,377.0811792,401.260068,406.4339268,2525.510204,2360.20
4082,2106.122449,2442.857143,
2233.163265,2421.428571,2591.836735,2400,2506.632653,2495
.918367]

n=np.polyfit(x,y, 5)
Y=np.polyval(n,x)
print(" Observed values of strength \n",y)
print("Predicted values of strength \n",Y)
M=np.poly1d(n)
print ("The best curve fitted equation is \n",M)
per_error= 0
for i in range ( 0 , len (x)):
   per_error+=abs((y[i]-Y[i])/y[i])
print ( "Error" ,(per_error))
print ( "i.e. Error=" ,np.round((per_error)))
plt.plot(x,y,ls='-')
plt.plot(x,Y,ls='-.')
plt.xlabel("no.of yarns")
plt.ylabel("strength")
plt.legend(['observed','predicted'])
plt.title('Relation between no.of yarns & strength of
NYLON')
plt.show()
s=int(input("Enter the number of yarn :"))
op=n[0]*s**5+n[1]*s**4+n[2]*s**3+n[3]*s**2+n[4]*s+n[5]
print("Maximum force will be ",op," Kgf")
```

Output:

```
Observed values of strength
 [192.484, 176.338, 194.113, 187.645, 190.0645,
311.82408, 285.66756, 314.46306, 303.9849, 307.90449,
411.6077856, 415.0912392, 377.0811792, 401.260068,
406.4339268, 2525.510204, 2360.204082, 2106.122449,
2442.857143, 2233.163265, 2421.428571, 2591.836735, 2400,
2506.632653, 2495.918367]
Predicted values of strength
 [ 188.1289      188.1289      188.1289      188.1289
188.1289
  304.768818    304.768818    304.768818    304.768818
304.768818
  402.29483976  402.29483976  402.29483976  402.29483976
402.29483976
 2333.5714286  2333.5714286  2333.5714286  2333.5714286
2333.5714286
 2483.1632652  2483.1632652  2483.1632652  2483.1632652
2483.1632652 ]
The best curve fitted equation is
             5             4           3         2
-3.334e-06 x + 0.0005302 x + 0.02183 x - 6.776 x + 332.7
x - 4409
Error 0.8007715804184221
i.e. Error= 1.0
Enter the number of yarn :25
Maximum force will be  188.12889999999152  Kgf
```

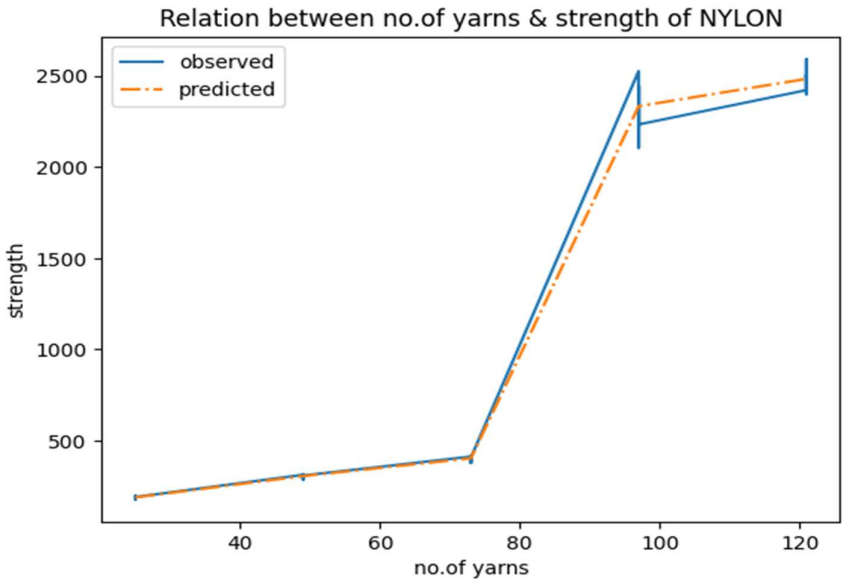


Figure 5. 21 Best fitted curve for nylon rod

In the same manner with the help of Reverse curve fitting analysis tensile strength was analyzed for all other materials.

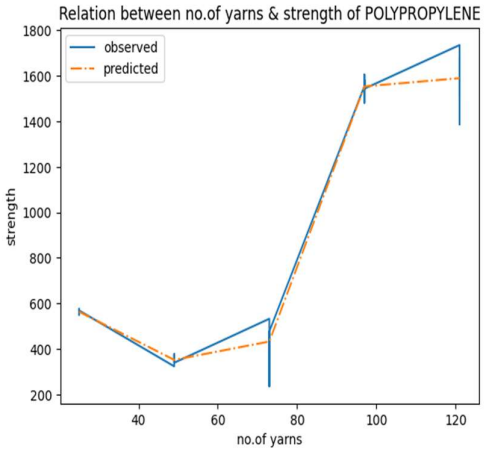


Figure 5. 22 Best fitted curve for polypropylene rod

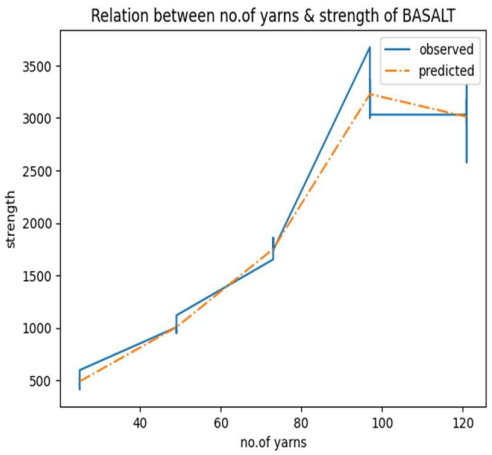


Figure 5. 23 Best fitted curve for basalt rod

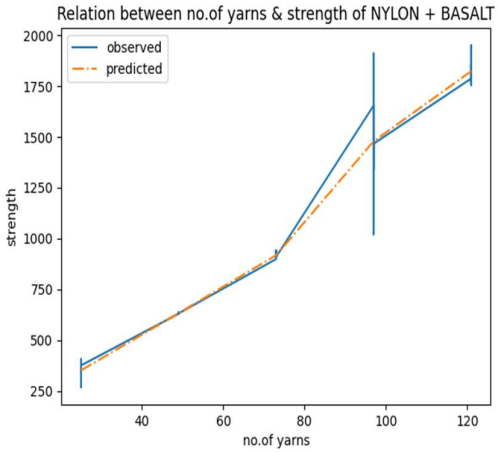


Figure 5. 24 Best fitted curve for nylon + basalt rod

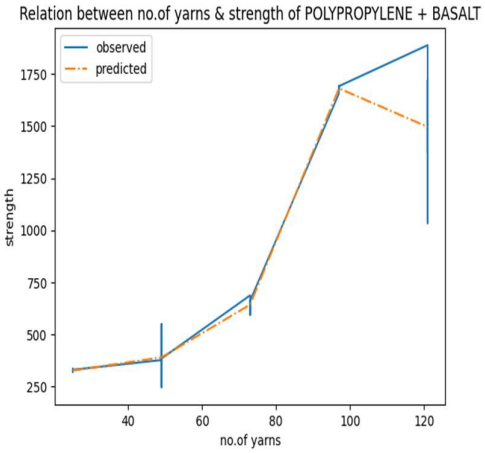


Figure 5. 25 Best fitted curve for polypropylene + basalt rod

5.8.2 Comparative Analysis for Composite Rods

5.8.2.1 Nylon, basalt, polypropylene rod

```
import matplotlib.pyplot as plt
x =
[25,25,25,25,25,49,49,49,49,49,73,73,73,73,73,97,97,97,97
,97,121,121,121,121,121]
n_y=[192.484,176.338,194.113,187.645,190.0645,311.82408,2
85.66756,314.46306,303.9849,307.90449,411.6077856,415.091
2392,377.0811792,401.260068,406.4339268,2525.510204,2360.
204082,2106.122449,2442.857143,2233.163265,2421.428571,25
91.836735,2400,2506.632653,2495.918367]
b_y=
[450.905,520.321,469.257,416.497,597.837,1005.93,986.43,9
```

```
70.35,952.747,1119.46,1652.325,1689.303,1819.069,1857.784
,1739.257,3673.469388,3061.22449,3000,3367.346939,3030.61
2245,3030.612245,3321.428571,2577.55102,3176.020408,2949.
489796]
pp_y=
[576.165,560.095,550.361,562.207,569.186,323.875,377.618,
364.849,355.4473333,339.6611667,532.777,496.276,236.101,4
21.718,477.2475,1555.102041,1605.102041,1481.632653,1580.
102041,1543.367347,1734.693878,1632.653061,1386.734694,16
83.673469,1509.693878]
plt.plot(x,n_y)
plt.plot(x,b_y)
plt.plot(x,pp_y)
plt.xlabel('no. of yarn of layer ')
plt.ylabel('tensile strength')
plt.legend(['nylon rod ','basalt rod ','polypropylene
rod'])
plt.title('comparison of nylon and basalt and
polypropylene rod ')
```

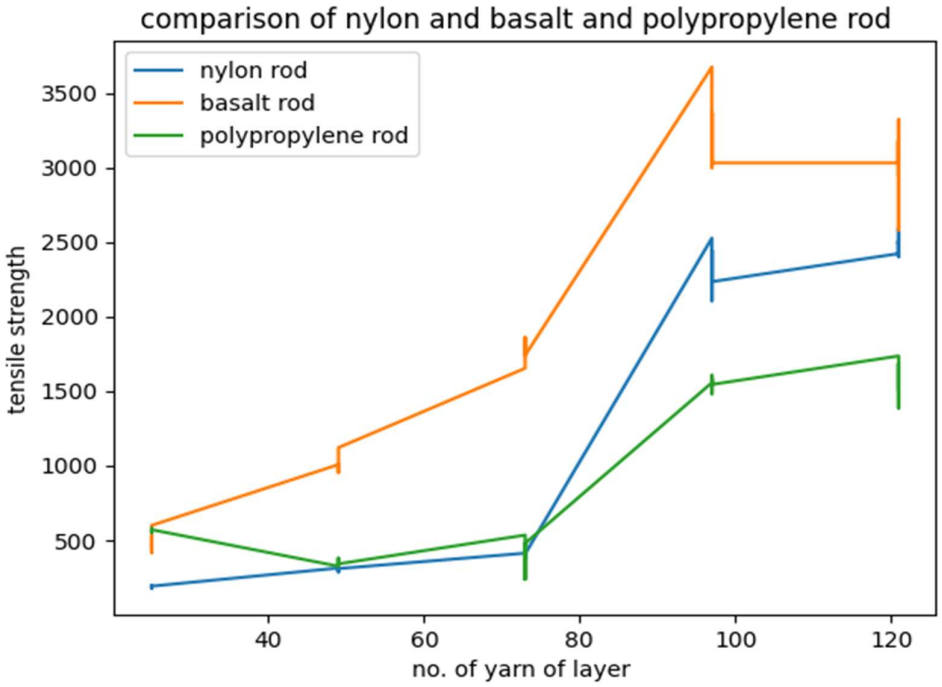


Figure 5. 26 Comparison of nylon, basalt and polypropylene rod

Based on the tensile strength data presented, the basalt rod demonstrates the highest tensile strength across all yarn layers, followed by the nylon and polypropylene rods. Therefore, for applications requiring high tensile strength, the basalt rod would be the preferred choice among the three materials studied.

5.8.2.2 Nylon, basalt, nylon+basalt mix rod

```
import matplotlib.pyplot as plt
x =
[25,25,25,25,25,49,49,49,49,49,73,73,73,73,73,97,97,97,97
,97,121,121,121,121,121]
n_y=[192.484,176.338,194.113,187.645,190.0645,311.82408,2
85.66756,314.46306,303.9849,307.90449,411.6077856,415.091
2392,377.0811792,401.260068,406.4339268,2525.510204,2360.
204082,2106.122449,2442.857143,2233.163265,2421.428571,25
91.836735,2400,2506.632653,2495.918367]

b_y =
[450.905,520.321,469.257,416.497,597.837,1005.93,986.43,9
70.35,952.747,1119.46,1652.325,1689.303,1819.069,1857.784
,1739.257,3673.469388,3061.22449,3000,3367.346939,3030.61
2245,3030.612245,3321.428571,2577.55102,3176.020408,2949.
489796]
```

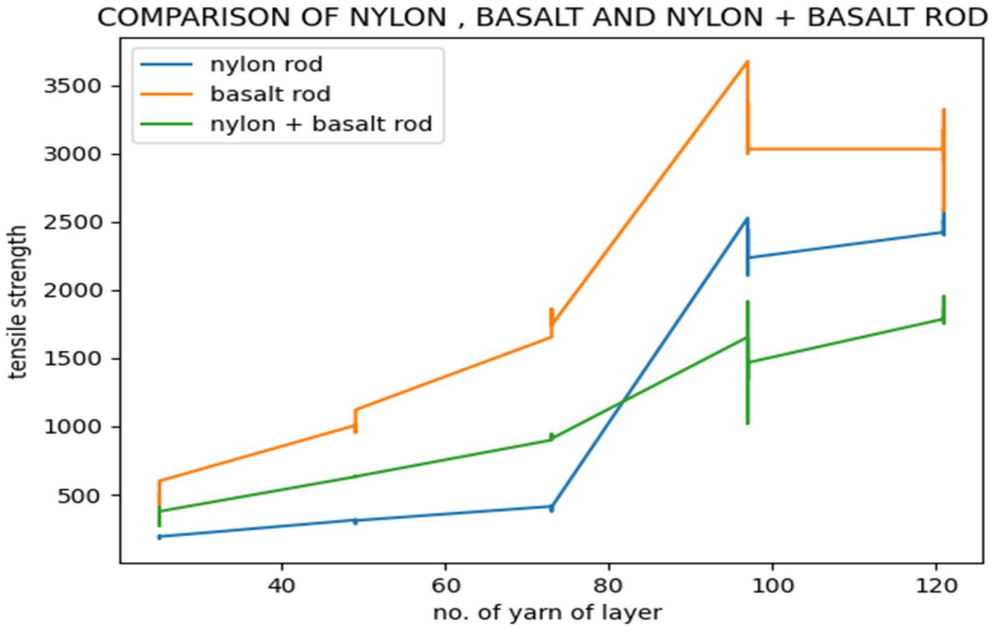


Figure 5. 27 Comparison of nylon, basalt and nylon + basalt rod

```
nb_y =
[406.295,360.147265,268,343.9033333,375.0991667,629.016,6
31.828,635.47,632.1046667,630.5603333,898.0510667,917.908
37,940.4789385,918.8127917,908.4319292,1653.061224,1020.4
08163,1910.204082,1336.734694,1465.306122,1785.714286,175
3.061224,1950,1769.387755,1851.530612]
plt.plot(x,n_y)
plt.plot(x,b_y)
plt.plot(x,nb_y)
plt.xlabel('no. of yarn of layer ')
plt.ylabel('tensile strength')
```

```
plt.legend(['nylon rod ','basalt rod ','nylon + basalt
rod'])
plt.title('COMPARISON OF NYLON AND BASALT AND 'NYLON +
BASALT ROD ')
```

1. ***Strength comparison***:

- Nylon Rod (n_y): The tensile strength of Nylon rod exhibits a moderate increase with the addition of yarn layers.
- Basalt Rod (b_y): The Basalt rod demonstrates a more significant and steady increase in tensile strength with an increase in yarn layers.
- Nylon + Basalt Rod (nb_y): The Nylon + Basalt mix rod shows a substantial increase in tensile strength as the number of yarn layers increases.

2. ***Effect of yarn count:***

- All three types of rods (Nylon, Basalt, and Nylon + Basalt mix) show an overall increase in tensile strength with an increase in yarn layers. However, the rate of increase and the maximum strength achieved vary among the materials.
- The Basalt rod exhibits the highest tensile strength values across all measured yarn counts, followed by the Nylon rod and then the Nylon + Basalt mix rod.

3. ***Material properties:***

- The observed differences in tensile strength among Nylon, Basalt, and Nylon + Basalt mix rods can be attributed to variations in material composition, structural characteristics, and inherent mechanical properties of the respective materials.

4. ***Practical implications:***

- The Basalt rod, with its superior tensile strength values, may be preferred in applications requiring high mechanical strength and load-bearing capacity.
- The Nylon rod could be suitable for applications where a balance between strength and flexibility is desired.
- The Nylon + Basalt mix rod offers an intermediate level of tensile strength and could be utilized in applications that require a combination of mechanical strength and other desirable properties.

In conclusion, the comparison of tensile strength depending on the layer-wise braided yarn count highlights distinct differences among Nylon, Basalt, and Nylon + Basalt mix

rods. The choice of material and the number of yarn layers significantly influence the tensile strength, thereby impacting their suitability for specific applications.

5.8.2.3 Polypropylene, basalt, polypropylene+basalt mix rod

```
import matplotlib.pyplot as plt
x =
[25,25,25,25,25,49,49,49,49,49,73,73,73,73,73,97,97,97,97
,97,121,121,121,121,121]
p_y=[576.165,560.095,550.361,562.207,569.186,323.875,377.
618,364.849,355.4473333,339.6611667,532.777,496.276,236.1
01,421.718,477.2475,1555.102041,1605.102041,1481.632653,1
580.102041,1543.367347,1734.693878,1632.653061,1386.73469
4,16
```

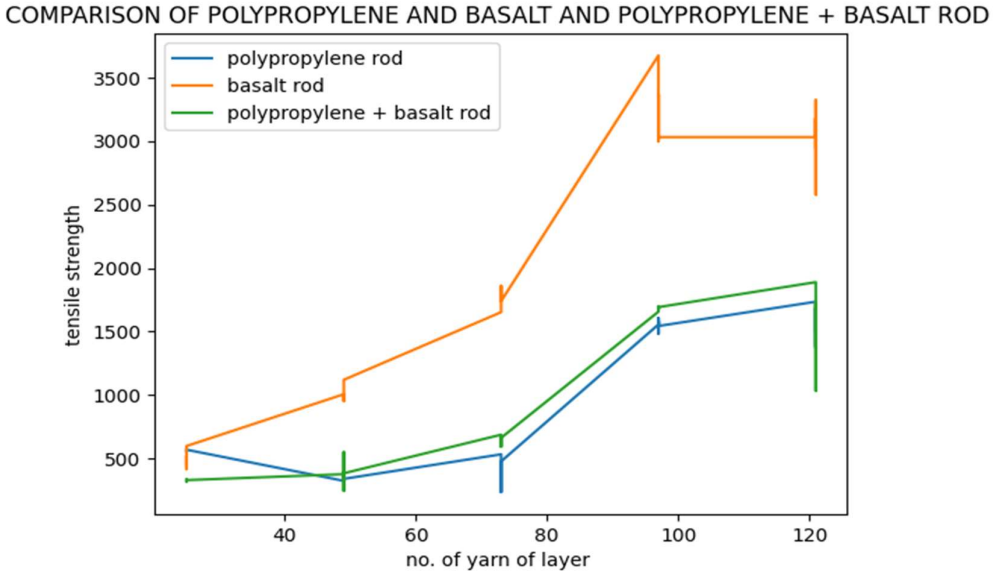


Figure 5. 28 Comparison of polypropylene, basalt and polypropylene + basalt rod

```
83.673469,1509.693878]
b_y =
[450.905,520.321,469.257,416.497,597.837,1005.93,986.43,9
70.35,952.747,1119.46,1652.325,1689.303,1819.069,1857.784
,1739.257,3673.469388,3061.22449,3000,3367.346939,3030.61
2245,3030.612245,3321.428571,2577.55102,3176.020408,2949.
489796]
pb_y=[334.398,319.986,320.584,324.9893333,329.6936667,376
.326,549.205,246.03,390.5203333,383.4231667,687.277,593.5
89,632.124,637.6633333,662.4701667,1656.122449,1693.87755
1,1689.795918,1675,1691.836735,1888.77551,1034.693878,172
0.408163,1461.734694,1377.55102]
plt.plot(x,p_y)
plt.plot(x,b_y)
plt.plot(x,pb_y)
plt.xlabel('no. of yarn of layer ')
plt.ylabel('tensile strength')
```

```
plt.legend(['polypropylene rod ','basalt rod
','polypropylene + basalt rod'])
plt.title('COMPARISON OF POLYPROPYLENE AND BASALT AND
POLYPROPYLENE + BASALT ROD')
```

Based on the plotted data for Polypropylene, Basalt, and Polypropylene + Basalt mix rods of tensile strength depending on the layer-wise braided yarn count:

1. Strength comparison:

- Polypropylene Rod (p_y): The tensile strength of the Polypropylene rod shows a fluctuating pattern with an increase in yarn layers.
- Basalt Rod (b_y): The Basalt rod demonstrates a significant and consistent increase in tensile strength with an increase in yarn layers.
- Polypropylene + Basalt Rod (pb_y): The Polypropylene + Basalt mix rod shows a moderate increase in tensile strength as the number of yarn layers increases.

2. Effect of yarn count:

- All three types of rods (Polypropylene, Basalt, and Polypropylene + Basalt mix) exhibit an overall increase in tensile strength with an increase in yarn layers. However, the rate of increase and the maximum strength achieved vary among the materials.
- The Basalt rod consistently demonstrates the highest tensile strength values across all measured yarn counts, followed by the Polypropylene + Basalt mix rod and then the Polypropylene rod.

3. Material properties:

- The observed differences in tensile strength among Polypropylene, Basalt, and Polypropylene + Basalt mix rods can be attributed to variations in material composition, structural characteristics, and inherent mechanical properties of the respective materials.

4. Practical implications:

- The Basalt rod, with its superior tensile strength values, may be preferred in applications requiring high mechanical strength and load-bearing capacity.
- The Polypropylene rod could be suitable for applications where a balance between strength and flexibility is desired.

- The Polypropylene + Basalt mix rod offers an intermediate level of tensile strength and could be utilized in applications that require a combination of mechanical strength and other desirable properties.

In conclusion, the comparison of tensile strength depending on the layer-wise braided yarn count highlights distinct differences among Polypropylene, Basalt, and Polypropylene + Basalt mix rods. The choice of material and the number of yarn layers significantly influence the tensile strength, thereby impacting their suitability for specific applications.

5.9 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) has gained significant attention in the field of textile engineering due to its potential in predicting material properties and optimizing manufacturing processes. This chapter explores the application of ANN in determining the tensile strength of braided ropes made of different fibers, employing the Levenberg–Marquardt algorithm for nonlinear optimization.

5.9.1 Historical Background of Artificial Neural Network

Artificial Neural Network is a computational model inspired by the structure and functioning of biological neural networks, such as the human brain. The concept of neural networks dates back to the mid-20th century with foundational contributions from researchers such as Warren McCulloch, Walter Pitts, Donald Hebb, Frank Rosenblatt, Bernard Widrow, and others. Over the years, neural networks have found applications in various engineering and economic domains, providing a framework for simulating learning, pattern recognition, and associative memorization.

- ***Optimization techniques in ANN***

For solving nonlinear optimization problems in this study, the Levenberg–Marquardt algorithm was employed due to its efficiency and capability to minimize the Mean Square Error (MSE) (Sanjari et al., 2009; Purwanto et al., 2008).

- ***Determination of optimal ANN architecture***

The optimal architecture of an ANN is a critical factor influencing its performance. In this study, a trial-and-error approach was utilized to determine the optimal number of nodes in the hidden layer, considering various configurations with one or two hidden layers and one to ten neurons (Tu, 1996; Wu et al., 2009). Mean Square Error (MSE)

and Mean Absolute Error (MAE) were used as performance metrics, with simpler architectures preferred over more complex ones (Didier, 2009; Haykin, 1994; Xu and Chen, 2008; Behera and Karthikeyan, 2006).

- ***Application in textile engineering***

Recently, Artificial Neural Networks have become a popular modeling tool in the field of textile engineering. They are commonly used in various applications related to different types of fabrics, including defect detection and strength determination. In this study, ANN was employed to determine the tensile strength of braided ropes made of different fibers, including nylon, basalt, and a mixture of both.

5.9.2 Advantages of ANN

Artificial Neural Networks (ANNs) offer several advantages, making them a powerful tool in various fields such as machine learning, pattern recognition, and artificial intelligence. Some of the key advantages of ANNs include:

1. Nonlinearity and flexibility: ANNs can model highly complex, nonlinear relationships between input and output data. This flexibility allows them to capture intricate patterns and make accurate predictions or classifications in diverse domains.

2. Adaptability and generalization: ANNs can adapt to new data and generalize well to unseen examples, making them robust in handling noisy or incomplete data. They can learn from experience and improve their performance over time, making them suitable for dynamic and changing environments.

3. Parallel processing: ANNs can perform parallel processing of information, mimicking the distributed processing of the human brain. This parallelism enables efficient computation and scalability, making ANNs suitable for large-scale data processing tasks.

4. Feature learning and representation: ANNs can automatically learn relevant features and representations from raw input data, reducing the need for manual feature engineering. This ability to extract meaningful features from data can lead to better performance and insights in tasks such as image recognition, natural language processing, and signal processing.

5. Robustness to noise and fault tolerance: ANNs exhibit robustness to noisy data and are capable of graceful degradation in performance even in the presence of faults or missing information. This fault tolerance property makes them suitable for applications where reliability is critical.

6. Applicability to various data types: ANNs can handle a wide range of data types, including structured data, unstructured data (such as text and images), time-series data, and multimodal data. This versatility allows ANNs to be applied to diverse tasks across different domains.

7. Scalability and parallelization: ANNs can be scaled to handle large datasets and complex models by leveraging parallel computing architectures such as GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units). This scalability enables efficient training and inference on big data platforms.

8. Interpretability and explainability: While interpretability can be a challenge in deep and complex neural networks, efforts are being made to develop techniques for interpreting and explaining the decisions made by ANNs. This is particularly important in applications such as healthcare and finance, where transparency and trustworthiness are crucial.

Overall, the advantages of ANNs contribute to their widespread use and applicability across a wide range of domains, driving advancements in artificial intelligence and machine learning research.

5.9.3 Application of Artificial Neural Network in Predicting Tensile Strength

In this section, an Artificial Neural Network (ANN) model is developed to predict the tensile strength of Fiber based on the provided dataset. The dataset consists of the variable 'x', which represents a feature related to the material, and 'y', which represents the tensile strength.

```
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import
```

```
mean_squared_error,mean_absolute_error,r2_score
from sklearn.preprocessing import StandardScaler

data = pd.read_excel('/content/nylon.xlsx')
xx=data[['x']]
X = data[['x']]
y = data["y"]

scaler = StandardScaler()
X = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.1, random_state=84)

mlp_regressor = MLPRegressor(hidden_layer_sizes=(10,10),
activation='relu', solver='adam', max_iter=500000,
random_state=84)
mlp_regressor.fit(X_train, y_train)
print("Random State:", mlp_regressor.random_state)
print("Activation Function:", mlp_regressor.activation)
print("Hidden Layers:", mlp_regressor.n_layers_ - 2)
print("Neurons in Each Hidden Layer:",
mlp_regressor.hidden_layer_sizes)
print("Solver:", mlp_regressor.solver)
print("Iterations:", mlp_regressor.n_iter_)
print("Training Set Size:", len(X_train))
print("Test Set Size:", len(X_test))
predicted_values = mlp_regressor.predict(X)
mse = mean_squared_error(y, predicted_values)
mae = mean_absolute_error(y, predicted_values)
r2=r2_score(y,predicted_values)
print('Mean Squared Error:', mse)
print('Mean absolute Error:', mae)
print('R2 score: ',r2)
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
true_predicted_df = pd.DataFrame({'True': y, 'Predicted':
predicted_values})
print("\nTrue and Predicted Values for the Entire
Dataset:")
print(true_predicted_df)
plt.plot(xx,y)
plt.scatter(xx, predicted_values)
plt.plot(xx, predicted_values,ls='-.')
```

5.9.3.1 Nylon rope

```
Random State: 84
Activation Function: relu
Hidden Layers: 2
Neurons in Each Hidden Layer: (10, 10)
Solver: adam
Iterations: 12690
Training Set Size: 5
Test Set Size: 1
Mean Squared Error: 49.29636758237897
```

Mean absolute Error: 2.9714831545444316

R2 score: 0.9997702086358171

```
True and Predicted Values for the Entire Dataset:
         True    Predicted
0    25.5320    25.518948
1   657.8194   657.907652
2  1395.3740  1395.159551
3  1073.1090  1073.343357
4  1152.8450  1152.761044
5  1259.1590  1276.353833
```

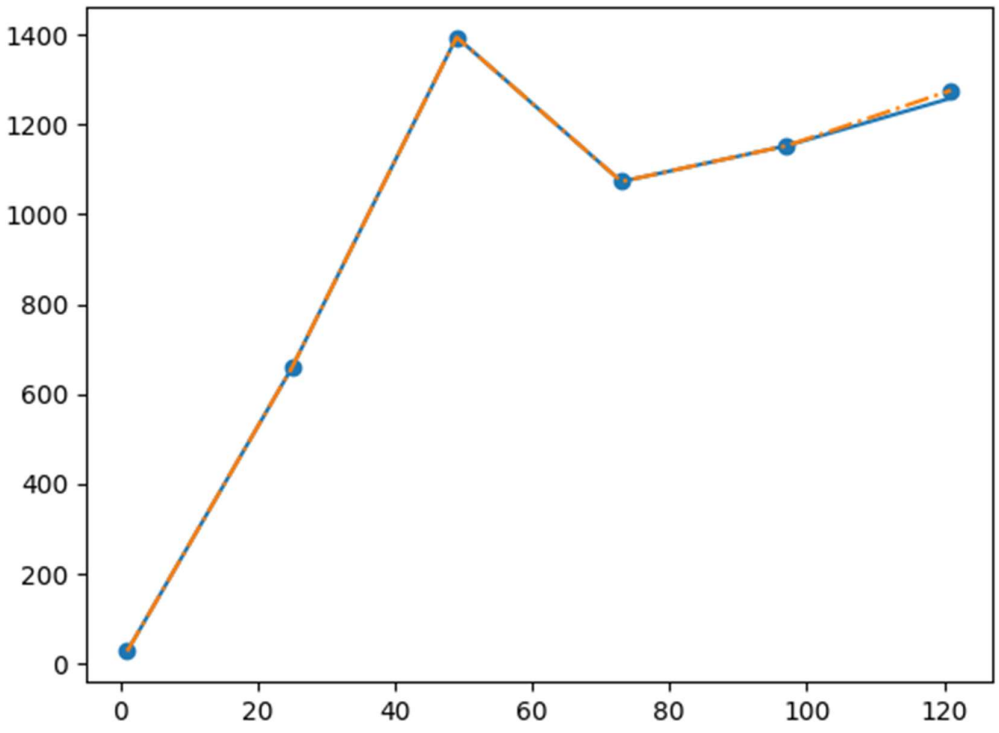


Figure 5. 29 MLP regressor for nylon rope

In the same manner for other ropes

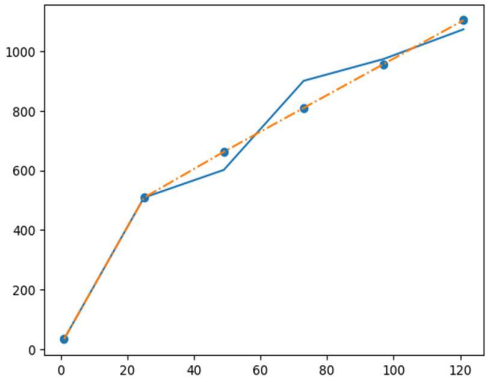


Figure 5. 30 MLP regressor for basalt rope

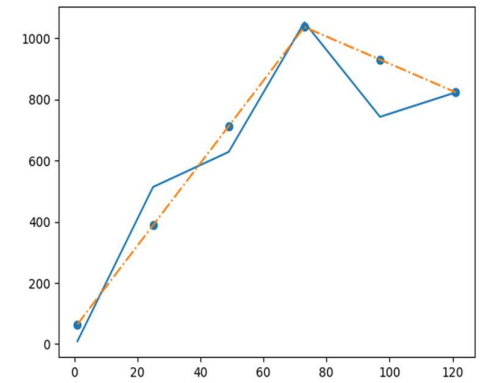


Figure 5. 31 MLP regressor for polypropylene rope

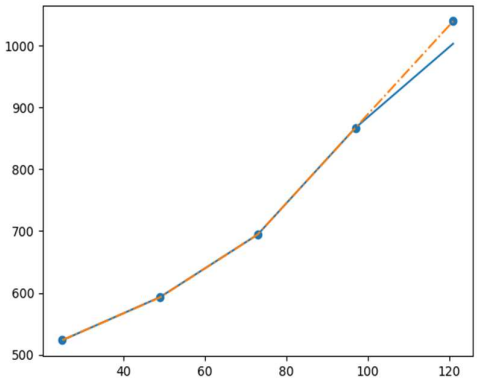


Figure 5. 32 MLP regressor for nylon + basalt rope

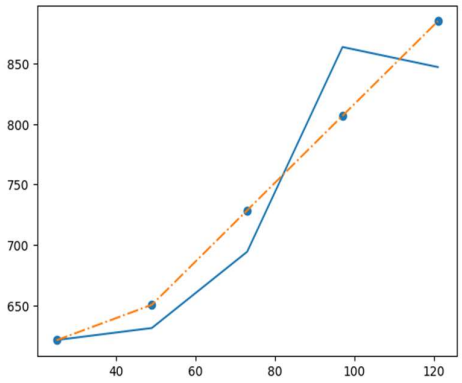


Figure 5. 33 MLP regressor for basalt + polypropylene rope

5.9.3.2 Nylon rod

```
Random State: 11
Activation Function: relu
Hidden Layers: 2
Neurons in Each Hidden Layer: (10, 10)
Solver: adam
Iterations: 15758
Training Set Size: 22
Test Set Size: 3
Mean Squared Error: 5580.922197911891
Mean absolute Error: 42.0233149677146
R2 score:  0.9948361003708848

True and Predicted Values for the Entire Dataset:
```

	True	Predicted
0	192.484000	192.344600
1	176.338000	192.344600
2	194.113000	192.344600
3	187.645000	192.344600
4	190.064500	192.344600
5	311.824080	298.932479
6	285.667560	298.932479
7	314.463060	298.932479
8	303.984900	298.932479
9	307.904490	298.932479
10	411.607786	405.520359
11	415.091239	405.520359
12	377.081179	405.520359
13	401.260068	405.520359
14	406.433927	405.520359
15	2525.510204	2358.195480
16	2360.204082	2358.195480
17	2106.122449	2358.195480
18	2442.857143	2358.195480
19	2233.163265	2358.195480
20	2421.428571	2483.343521
21	2591.836735	2483.343521
22	2400.000000	2483.343521
23	2506.632653	2483.343521
24	24 95.918367	2483.343521

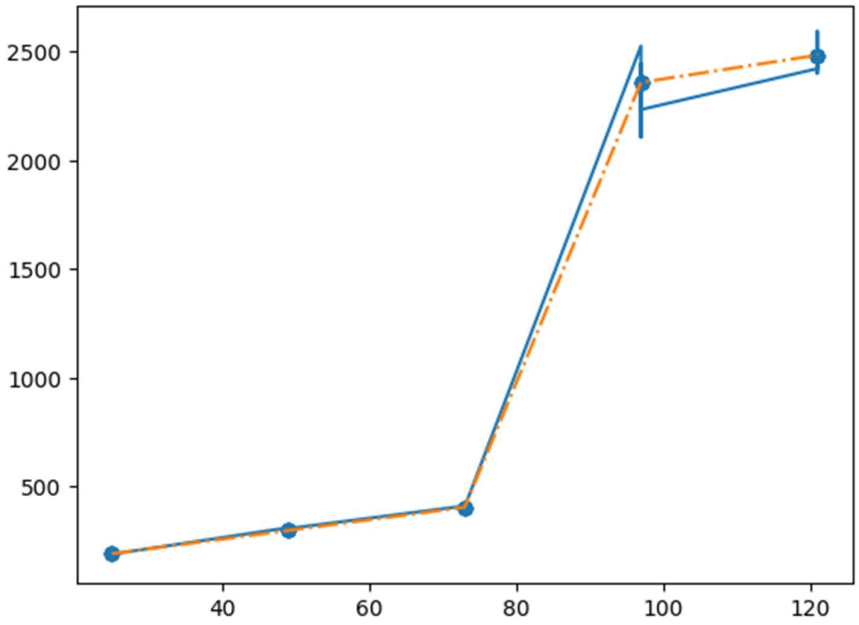


Figure 5. 34 MLP regressor for nylon rod

In the same manner for other rods

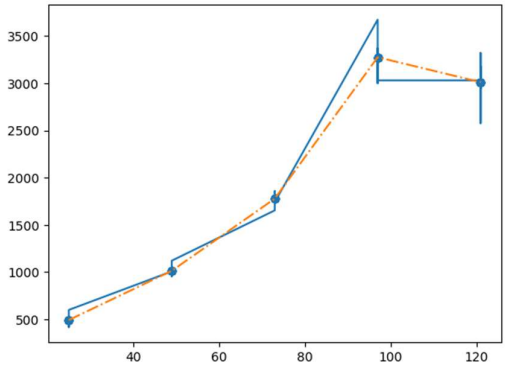


Figure 5. 35MLP regressor for basalt rod

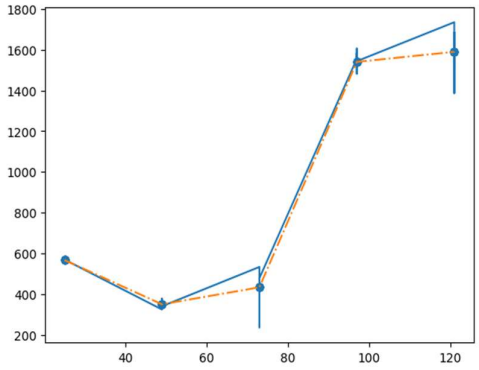


Figure 5. 36 MLP regressor for PP rod

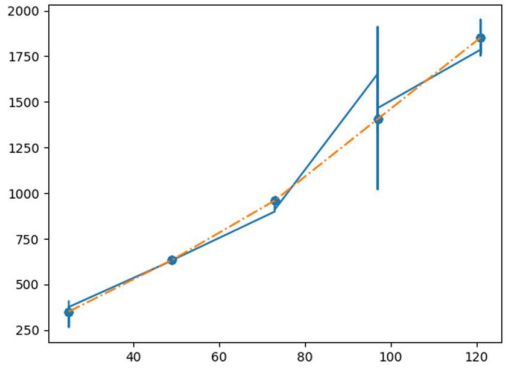


Figure 5. 37 MLP regressor for nylon + basalt rod

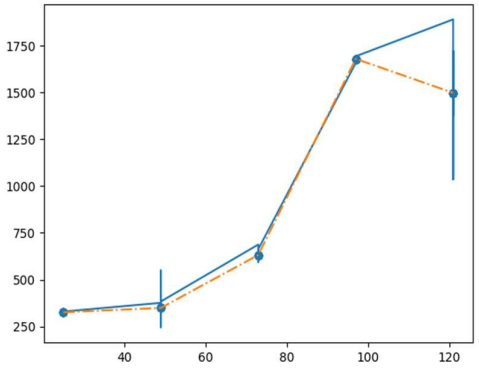


Figure 5. 38 MLP regressor for polypropylene + basalt rod

The proposed mathematical model offers a cost-effective and efficient alternative to traditional experiment testing methods, accurately predicting rope strength by considering key parameters like yarn strength and braiding configuration. Further refinement of the model could enhance its accuracy and applicability in real world scenario contributing to advancements in material science and engineering. Additionally, the analysis of the relationship between yarns and tensile strength in braided ropes, alongside investigations into composite rods, underscores the significance of accurate mathematical modelling in optimizing composite material properties. Leveraging curve fitting techniques and machine learning, these studies provide practical guidance for enhancing the design and performance of braided ropes and composite materials across various applications, fostering continued advancements in the field.