In [45]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, r2_score
from scipy.stats import pearsonr
```

In [46]:
```
df = pd.read_excel("C:/Users/91974/Desktop/Yokogawa/Model dataset/training/TRAIN-2.xlsx"
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1353 entries, 0 to 1352
Data columns (total 17 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   MFI gm/10 min           1353 non-null   float64
 1   PL25_HDMFI_XI1400A      1353 non-null   float64
 2   PL25_HDMFI_XI1401A      1353 non-null   float64
 3   PL25_HDMFI_XI1403A      1353 non-null   float64
 4   PL25_HDMFI_XI1430       1353 non-null   float64
 5   PL25_HDMFI_TIC1090A     1353 non-null   float64
 6   PL25_HDMFI_PI1111A      1353 non-null   float64
 7   PL25_HDMFI_KPI_DP_ADS   1353 non-null   float64
 8   PL25_HDMFI_TI1112       1353 non-null   float64
 9   PL25_HDMFI_XI1440B      1353 non-null   float64
 10  PL25_HDMFI_XI1440F      1353 non-null   float64
 11  PL25_HDMFI_TDI1129      1353 non-null   float64
 12  PL25_HDMFI_TDI1108C     1353 non-null   float64
 13  PL25_HDMFI_XI1428       1353 non-null   float64
 14  PL25_HDMFI_XI1405A      1353 non-null   float64
 15  PL25_HDMFI_XI1406A      1353 non-null   float64
 16  PL25_HDMFI_KPI_FB1_FLW  1353 non-null   float64
dtypes: float64(17)
memory usage: 179.8 KB
```

In [47]:
```
X = df.drop('MFI gm/10 min', axis=1)  # Features
y = df['MFI gm/10 min']  # Target variable
```

In [48]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=60
```

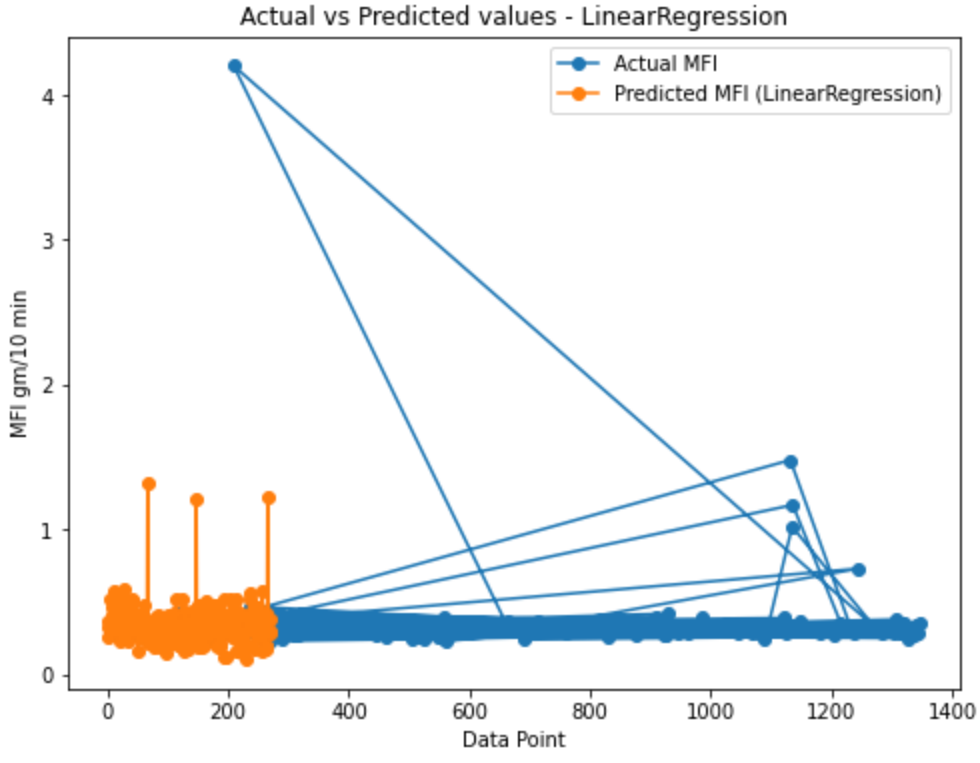
In [49]:
```
models = [
    LinearRegression(),
    DecisionTreeRegressor(),
    RandomForestRegressor(n_estimators=200)
]
```

In [50]:
```
comparison_df = pd.DataFrame()
```

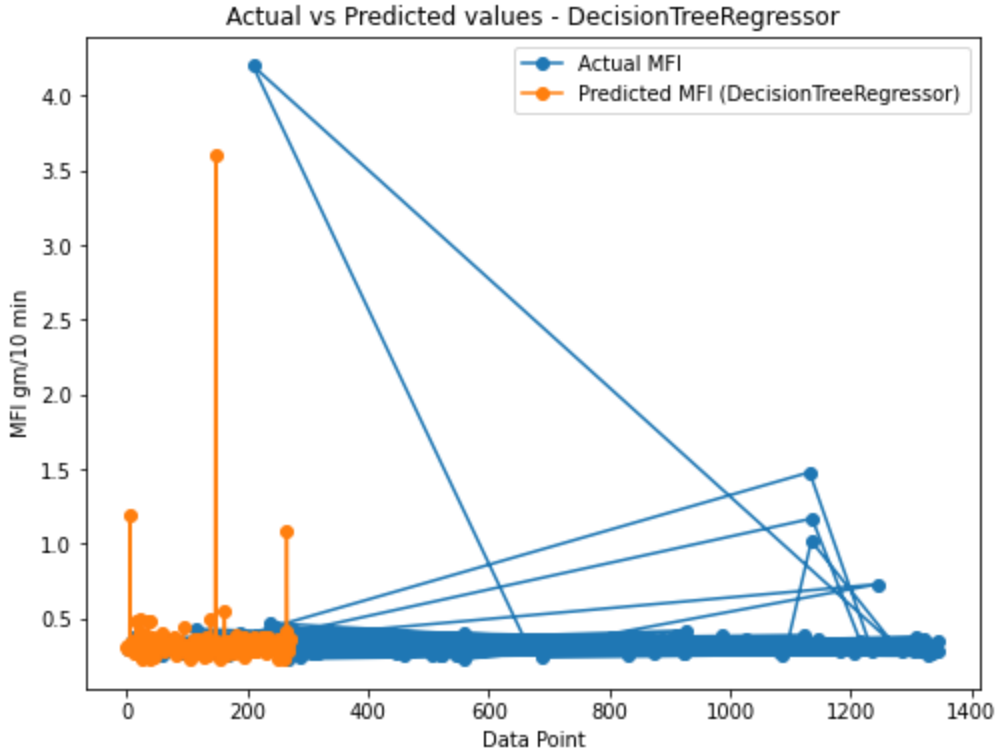
In [51]:
```
for model in models:
    # Train the model
    model.fit(X_train, y_train)

    # Make predictions for training set
    y_train_pred = model.predict(X_train)

    # Make predictions for testing set
    y_test_pred = model.predict(X_test)

    # Evaluate the model on training set
    mae_train = mean_absolute_error(y_train, y_train_pred)
```

```
    r2_train = r2_score(y_train, y_train_pred)

    # Evaluate the model on testing set
    mae_test = mean_absolute_error(y_test, y_test_pred)
    r2_test = r2_score(y_test, y_test_pred)

    # Print model performance
    print(f"\nModel: {type(model).__name__}")
    print(f"MAE (Training): {mae_train}")
    print(f"R-squared (Training): {r2_train}")
    print(f"MAE (Testing): {mae_test}")
    print(f"R-squared (Testing): {r2_test}")

    # Add actual and predicted values to the comparison dataframe
    comparison_df[f'Actual {type(model).__name__}'] = y_test
    comparison_df[f'Predicted {type(model).__name__}'] = y_test_pred

    # Displaying the DataFrame
    print(comparison_df.head())

    # Plotting the comparison graph as a line plot
    plt.figure(figsize=(8, 6))
    plt.plot(y_test, label='Actual MFI', marker='o')
    plt.plot(y_test_pred, label=f'Predicted MFI ({type(model).__name__})', marker='o')
    plt.title(f"Actual vs Predicted values - {type(model).__name__}")
    plt.xlabel("Data Point")
    plt.ylabel("MFI gm/10 min")
    plt.legend()
    plt.show()
```

```
Model: LinearRegression
MAE (Training): 0.11246282476897734
R-squared (Training): 0.03485235426741573
MAE (Testing): 0.0936323185461513
R-squared (Testing): 0.34339185281647344
      Actual LinearRegression  Predicted LinearRegression
987                      0.39                    0.371226
976                      0.31                    0.327010
1145                     0.30                    0.256618
787                      0.32                    0.286951
919                      0.30                    0.516561
```

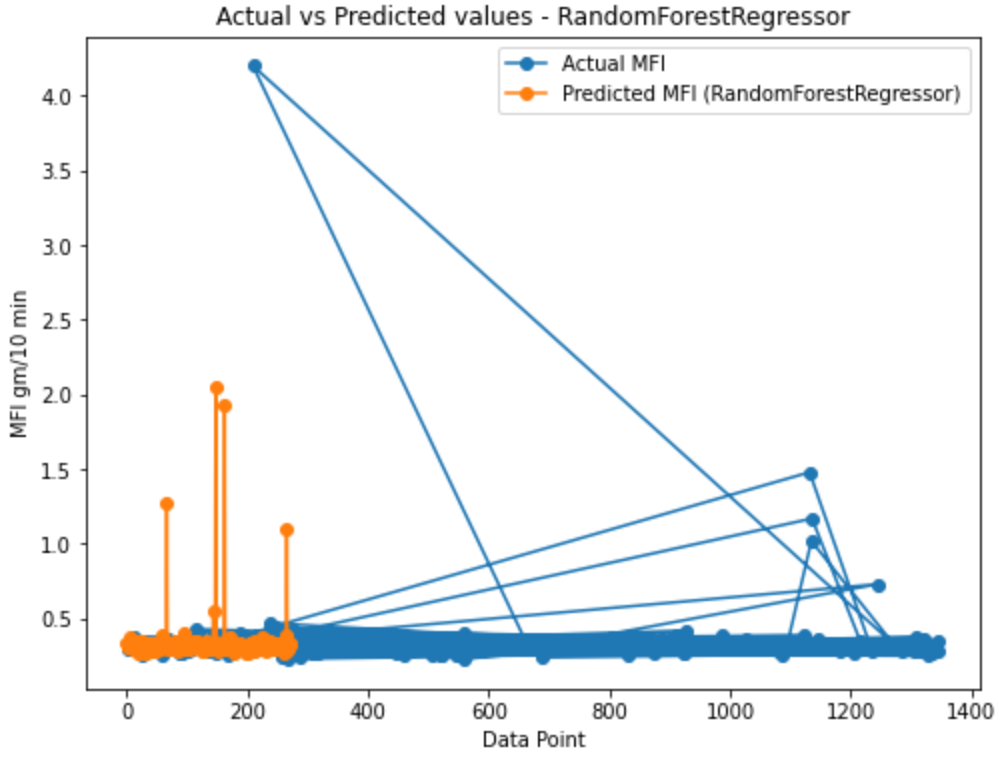


```
Model: DecisionTreeRegressor
MAE (Training): 1.0773883325844864e-18
R-squared (Training): 1.0
MAE (Testing): 0.06638376383763839
R-squared (Testing): -0.19918288006929208
      Actual LinearRegression  Predicted LinearRegression  \
987                      0.39                    0.371226
976                      0.31                    0.327010
1145                     0.30                    0.256618
787                      0.32                    0.286951
919                      0.30                    0.516561

      Actual DecisionTreeRegressor  Predicted DecisionTreeRegressor
987                           0.39                             0.31
976                           0.31                             0.29
1145                          0.30                             0.30
787                           0.32                             0.33
919                           0.30                             0.30
```




```
Model: RandomForestRegressor
MAE (Training): 0.03987236598890939
R-squared (Training): 0.7568196282594618
MAE (Testing): 0.047963284132841365
R-squared (Testing): 0.32050508146418677
      Actual LinearRegression  Predicted LinearRegression  \
987                      0.39                    0.371226
976                      0.31                    0.327010
1145                     0.30                    0.256618
787                      0.32                    0.286951
919                      0.30                    0.516561

      Actual DecisionTreeRegressor  Predicted DecisionTreeRegressor  \
987                           0.39                             0.31
976                           0.31                             0.29
1145                          0.30                             0.30
787                           0.32                             0.33
919                           0.30                             0.30

      Actual RandomForestRegressor  Predicted RandomForestRegressor
987                           0.39                          0.33640
976                           0.31                          0.31565
1145                          0.30                          0.30805
```

```
787                    0.32                    0.32745
919                    0.30                    0.33080
```



In [37]:
```
# Save the comparison dataframe to an Excel sheet
comparison_df.to_excel("C:/Users/91974/Desktop/Yokogawa/Model dataset/comparison_results
# Calculate and print the correlation coefficients
for model in models:
    correlation_coefficient, _ = pearsonr(comparison_df[f'Actual {type(model).__name__}'
    print(f"\nCorrelation coefficient for {type(model).__name__}: {correlation_coefficie
```

```
Correlation coefficient for LinearRegression: 0.5905702218234441

Correlation coefficient for DecisionTreeRegressor: 0.34169443052560994

Correlation coefficient for RandomForestRegressor: 0.5413125862584128
```

In [24]:
```
#model.fit(X_train, y_train)
```

In [25]:
```
#y_train_pred = model.predict(X_train)
#y_test_pred = model.predict(X_test)
```

In [26]:
```
#mse_train = mean_squared_error(y_train, y_train_pred)
#r2_train = r2_score(y_train, y_train_pred)
#mae_train = mean_absolute_error(y_train, y_train_pred)

#mse_test = mean_squared_error(y_test, y_test_pred)
#r2_test = r2_score(y_test, y_test_pred)
#mae_test = mean_absolute_error(y_test, y_test_pred)
```

In [27]:
```
#print("\nTraining Set Metrics:")
#print(f'Mean Squared Error: {mse_train}')
#print(f'R-squared: {r2_train}')
#print(f'Mean Absolute Error: {mae_train}')

#print("\nTesting Set Metrics:")
#print(f'Mean Squared Error: {mse_test}')
#print(f'R-squared: {r2_test}')
#print(f'Mean Absolute Error: {mae_test}')
```

In [28]:
```
#comparison_df = pd.DataFrame({'Actual MFI': y_test, 'Predicted MFI': y_test_pred})
```

```
#print("\nActual vs Predicted MFI for Testing Set:")
#print(comparison_df.head())
```