

In [1]: `import numpy as np`

```
random_forest_predictions = np.array([2494.12649657, 2549.6525147, 2608.92797354, 2591.26951911, 2587.01802989,
                                     2631.28349363, 2635.42156027, 2727.34553479, 2759.52348873, 2770.7750049, 2751.44550784, 2768.8596, 2733.0322])
ann_predictions = np.array([2475.8757, 2530.8108, 2570.6936, 2603.337, 2587.4731, 2590.8894, 2629.054, 2702.0986, 2739.1794, 2762.2776,
                           2768.8596, 2733.0322])
```

```
# Combine predictions using a simple average
combined_predictions = (random_forest_predictions + ann_predictions) / 2
```

```
print("Combined Predictions:", combined_predictions)
```

```
Combined Predictions: [2485.00109828 2540.23165735 2589.81078677 2597.30325956 2587.24556495
 2611.08644682 2632.23778014 2714.72206739 2749.35144437 2766.52630245
 2760.15255392 2737.12261985]
```

In [3]: `from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score`

```
# Actual stock prices for the same data points
```

```
actual_stock_prices = np.array([2496.449951, 2550.250000, 2615.699951, 2588.750000, 2584.500000, 2638.750000, 2633.600098, 2735.050049, 2768.8596, 2733.0322])
```

```
# Calculate evaluation metrics for the combined predictions
```

```
combined_mse = mean_squared_error(actual_stock_prices, combined_predictions)
```

```
combined_mae = mean_absolute_error(actual_stock_prices, combined_predictions)
```

```
combined_r2 = r2_score(actual_stock_prices, combined_predictions)
```

```
print("Combined Predictions Metrics:")
```

```
print("Mean Squared Error:", combined_mse)
```

```
print("Mean Absolute Error:", combined_mae)
```

```
print("R-squared:", combined_r2)
```

```
Combined Predictions Metrics:
```

```
Mean Squared Error: 225.56875292169136
```

```
Mean Absolute Error: 12.10926050083333
```

```
R-squared: 0.9711977338568019
```

In [4]: `import matplotlib.pyplot as plt`

```
# Data points for x-axis
```

```
data_points = np.arange(1, 13) # Assuming 12 data points for the predictions
```

```
# Plot actual values
```

```
plt.plot(data_points, actual_stock_prices, label='Actual', marker='o')
```

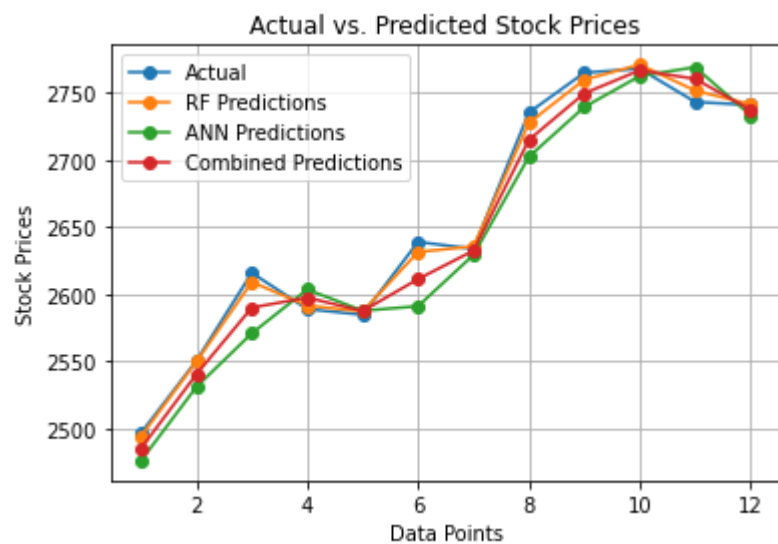
```
# Plot predicted values from Random Forest
plt.plot(data_points, random_forest_predictions, label='RF Predictions', marker='o')

# Plot predicted values from ANN
plt.plot(data_points, ann_predictions, label='ANN Predictions', marker='o')

# Plot combined predicted values
plt.plot(data_points, combined_predictions, label='Combined Predictions', marker='o')

# Add Labels and Legend
plt.xlabel('Data Points')
plt.ylabel('Stock Prices')
plt.title('Actual vs. Predicted Stock Prices')
plt.legend()
plt.grid(True)

# Show the plot
plt.show()
```



In []: