1. Description

1.1 Basic Information

The motivation behind this project lies in the pursuit of developing robust predictive models for stock price movements, specifically focusing on companies such as AMD, Electronic Arts, and Apple. The objective is to leverage machine learning techniques to forecast stock prices over different time frames (1, 3, 5, 10 years, and full range), aiding investors and financial analysts in making informed decisions.

In financial markets, predicting stock prices accurately is a challenging task due to the myriad factors influencing market dynamics. The need for reliable forecasting tools is crucial for optimizing investment strategies. This project employs diverse models, including Linear Regression, Decision Trees, and Long Short-Term Memory (LSTM) networks, to capture varying patterns and complexities inherent in financial time series data.

The choice of companies, AMD, Electronic Arts, and Apple, represents a mix of technology and entertainment sectors, offering a comprehensive analysis across different industries. The inclusion of multiple time frames allows for a nuanced understanding of model performance over varying investment horizons.

Through this project, I aimed to not only develop accurate prediction models but also conduct a comprehensive analysis of model performance, providing insights into the strengths and weaknesses of each model type. This information is crucial for investors seeking to align their investment strategies with the dynamic nature of financial markets.

1.1 Project Objectives

The primary objectives of the project are centered around predicting stock prices for the selected companies (AMD, ELECTRONIC ARTS, APPLE) over different time ranges (1, 3, 5, 10 years, and the full available range).

The main questions the project aims to address include:

Prediction Accuracy Across Models and Time Ranges:

1. Assessing and comparing the prediction accuracy of three different models: Linear Regression, Decision Trees, and LSTM (Long Short-Term Memory) neural networks.

2. Evaluating how well these models perform over varying time ranges to understand their suitability for short-term and long-term stock price predictions.

Feature Importance Analysis:

Conducting feature importance analysis, particularly for Decision Trees, to understand which stock features (Open, High, Low, Volume) contribute significantly to the prediction accuracy.

Overall Best Model and Time Frame:

Identifying the best-performing model and time range for each company, as well as an overall best-performing model and time range across all companies.

Variability Analysis:

Analyzing the variability in prediction accuracy across different models and time frames to understand the robustness and consistency of each model under varying market conditions.

1.1 Description of the Data Set

The process of obtaining this dataset involved leveraging the yfinance library to download stock data for AMD, ELECTRONIC ARTS, and APPLE. Subsequently, the individual datasets were merged into a unified DataFrame, providing a consolidated view for comparative analyses and predictions.The dataset offers a comprehensive exploration of historical stock price data for three key technology companies: Advanced Micro Devices (AMD), ELECTRONIC ARTS, and APPLE. Spanning an extensive time frame from January 2, 2001, to November 29, 2023, the dataset allows for an in-depth analysis of these companies' stock performance over more than two decades.

Each entry in the dataset corresponds to a specific date, forming a time-series structure that captures the day-to-day fluctuations in stock prices.

Columns:

Open: Represents the opening price of the stock at the beginning of a trading day.
High: Indicates the highest price reached by the stock during the trading day.
Low: Signifies the lowest price recorded by the stock in the course of the trading day.
Close: Represents the closing price of the stock at the end of the trading day.
Adj Close: Denotes the adjusted closing price, accounting for actions like dividends, stock splits.
Volume: Reflects the total number of shares traded on a given day.
company_name: Identifies the specific company to which the stock data belongs.

Each of the three companies contributes 5764 entries to the dataset, creating a rich temporal profile that facilitates the exploration of various market conditions, economic events, and technological advancements over the extended timeframe.

This detailed dataset forms the cornerstone for the project's objectives, enabling analyses related to stock price prediction and the evaluation of diverse modeling techniques. It serves as a valuable resource for gaining insights into the intricate dynamics of the stock market and understanding the factors influencing the stock performance of the selected technology companies over the years.

Head()

```
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
AMD: 5764
ELECTRONIC ARTS: 5764
APPLE: 5764
Date Range: 2001-01-02 to 2023-11-29
```

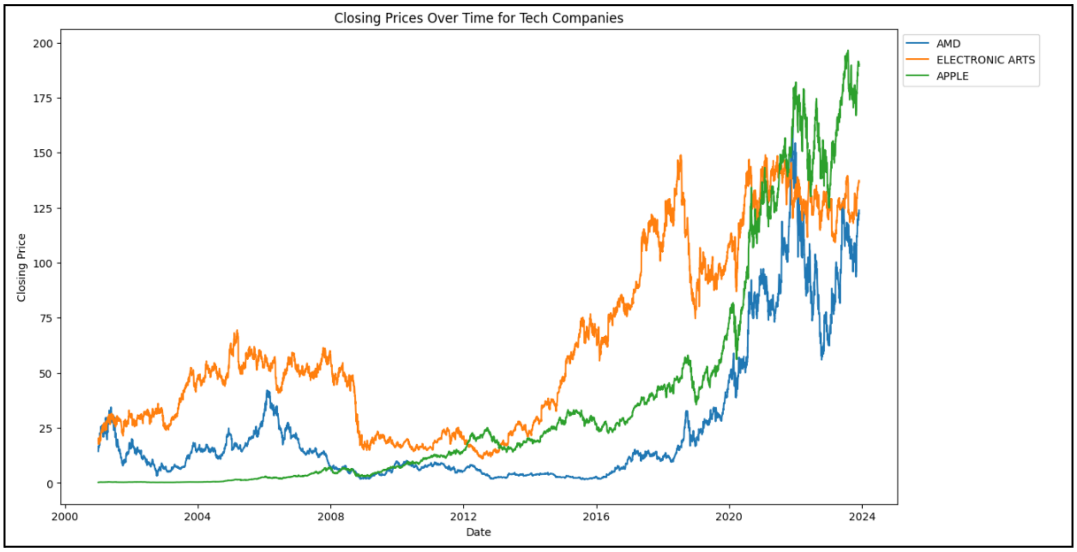
Date	Open	High	Low	Close	Adj Close	Volume	company_name
2001-01-02	14.1250	14.7500	14.1250	14.3750	14.3750	4863600	AMD
2001-01-03	14.5000	16.3750	14.4375	16.2500	16.2500	8808100	AMD
2001-01-04	16.1250	17.3750	15.7500	16.6875	16.6875	7045000	AMD
2001-01-05	17.1875	17.1875	15.6875	16.0000	16.0000	5230600	AMD
2001-01-08	15.9375	16.2500	15.4375	16.1875	16.1875	3365700	AMD

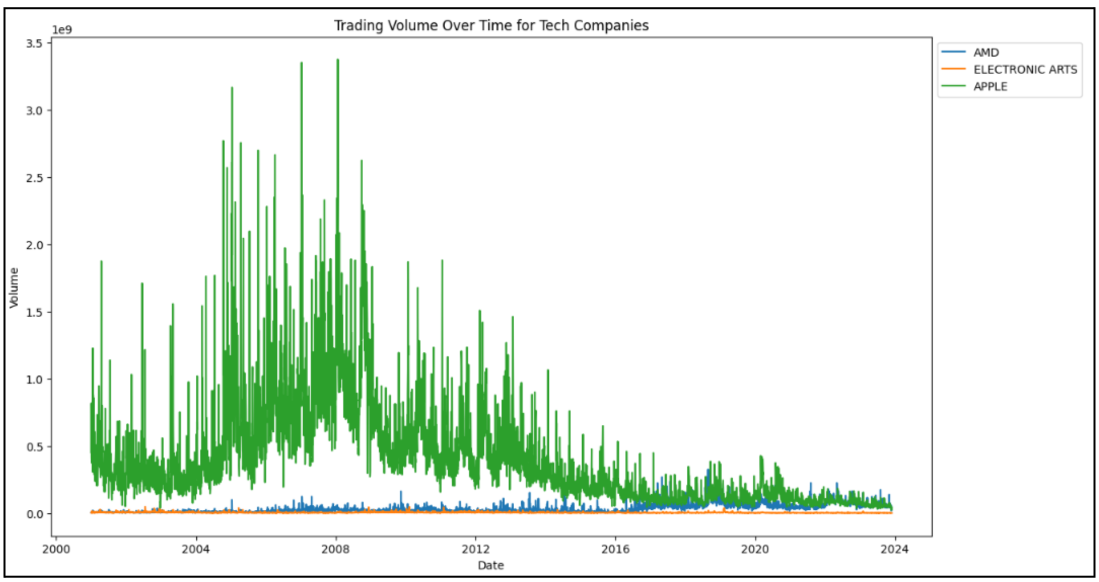
Tail()

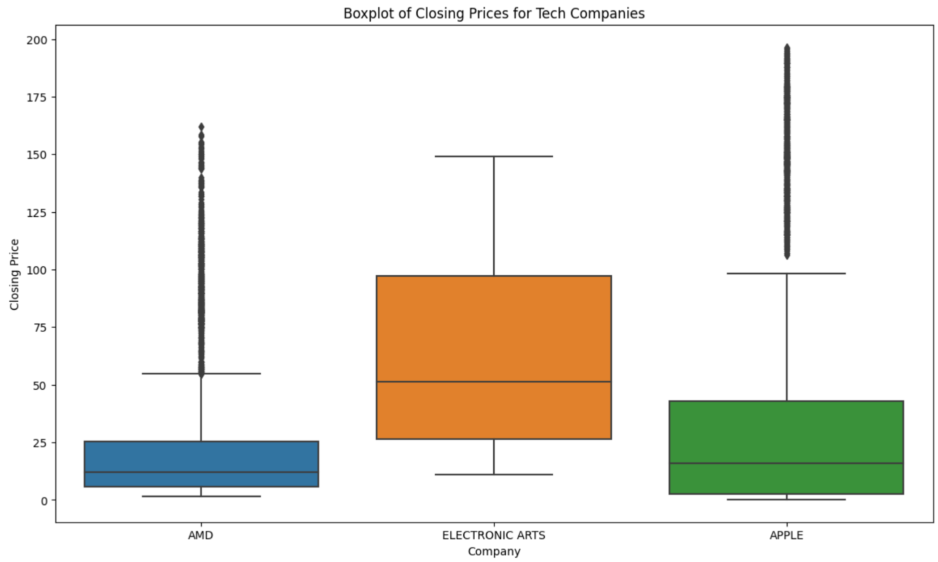
```
df.tail()
```

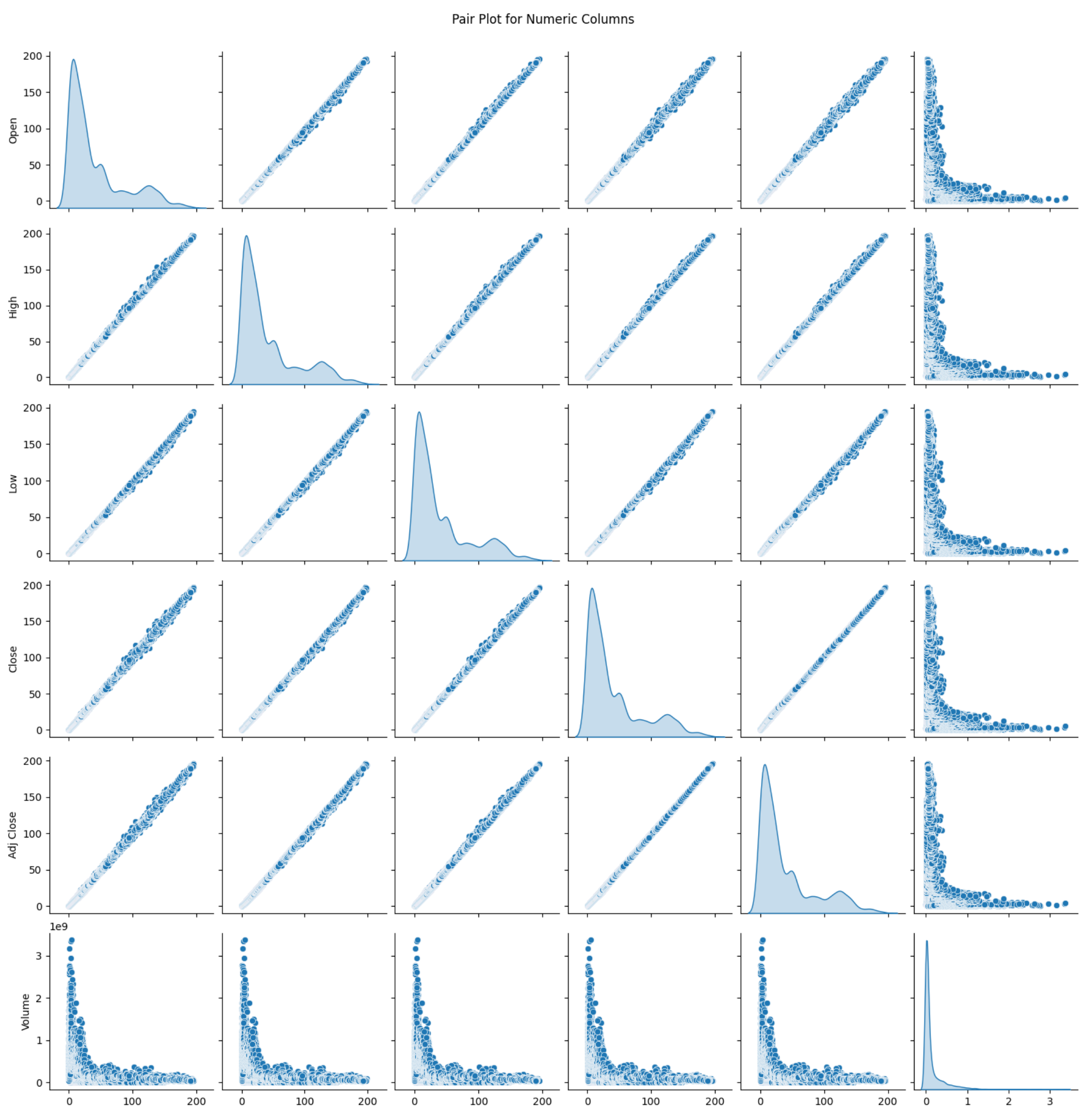
Date	Open	High	Low	Close	Adj Close	Volume	company_name
2023-11-22	191.490005	192.929993	190.830002	191.309998	191.309998	39617700	APPLE
2023-11-24	190.869995	190.899994	189.250000	189.970001	189.970001	24048300	APPLE
2023-11-27	189.919998	190.669998	188.899994	189.789993	189.789993	40552600	APPLE
2023-11-28	189.779999	191.080002	189.399994	190.399994	190.399994	38415400	APPLE
2023-11-29	190.899994	192.089996	188.970001	189.369995	189.369995	43014200	APPLE

Viewing the Data






Boxplot of Closing Prices for Tech Companies
Closing Price
200
175
150
125
100
75
50
25
0
AMD
ELECTRONIC ARTS
APPLE
Company


Pair Plot for Numeric Columns
Open
High
Low
Close
Adj Close
Volume
1e9

2. Design of the Project

The project's design revolves around the task of predicting stock prices using a multifaceted approach. This involves the integration of diverse techniques, each contributing a unique perspective to the challenge. The project's motivation stems from the need to explore and compare the effectiveness of traditional statistical methods, like Linear Regression, against more complex machine learning models, such as Decision Trees, and deep learning techniques, like LSTM networks.

The project starts with a meticulous examination of historical stock data, focusing on key variables like Open, High, Low, and Volume. These variables serve as the foundation for training and testing the predictive models. The choice of multiple models facilitates a holistic understanding of their respective strengths and limitations in capturing the intricate patterns inherent in stock price movements.

2.1 Technique Methodology

The project methodology seamlessly blends traditional statistical methods with advanced machine learning and deep learning techniques. It starts with Linear Regression, a fundamental tool leveraging historical price trends to establish a baseline for predicting stock prices, assuming a linear relationship.This method capitalizes on historical price trends, assuming a linear relationship between predictor variables and stock closing prices.

The Decision Trees model introduces a layer of complexity, chosen for its interpretability and ability to capture non-linear relationships within the data. In this context, I have implemented the Decision Tree Regressor, leveraging its inherent capacity to decipher complex patterns. What sets this apart is the incorporation of SHAP (SHapley Additive exPlanations) values, a cutting-edge technique that sheds light on feature importance. This not only enhances the model's interpretability but also contributes to a deeper understanding of how specific factors influence stock price predictions.

Then I explored how deep learning works through the inclusion of Long Short-Term Memory (LSTM) networks. Unlike traditional models, LSTM networks excel at capturing temporal dependencies in sequential data, making them particularly suited for time-series forecasting. This part of the methodology involves data preprocessing, including scaling and reshaping to meet the input requirements of the LSTM model. The preprocessing steps involve crucial tasks like splitting the data into training and testing sets, a pivotal aspect in ensuring the models are evaluated on unseen data to gauge their real-world predictive capabilities.

The model training process is tailored to the specific time ranges (1, 3, 5, 10 years, and full range) designated for evaluation. Each model undergoes training with its unique algorithm, adapting to the temporal nuances and patterns within the data. The Decision Trees model, in particular, stands out with its insightful feature importance analysis, driven by SHAP values. This comprehensive methodology not only delves into the technical intricacies of model training but also underscores the significance of interpretability, a crucial factor in making informed decisions based on predictive analytics in financial markets.

2.2 Implementation of the Project

Starting with data preprocessing, model training, and concluding with model evaluation. For the LSTM model, an additional layer of complexity involves data scaling and reshaping to meet the network's input requirements. The generated plots play a pivotal role in understanding the models' behavior, depicting the training loss curve for LSTM, and visually comparing predicted versus actual stock prices. Important implementations include the Decision Trees model's feature importance analysis and the visualization of its structure. SHAP values contribute to understanding the impact of different features on predictions.

2.3 Evaluation of the Project

The evaluation phase is where the project's models are rigorously tested and compared. Performance metrics, ranging from Mean Squared Error to R2 Score, offer quantitative insights into the efficacy of each model. The experiments are conducted across different time ranges, allowing for a nuanced understanding of how models perform under varying forecasting horizons.

Comparative analyses are conducted, shedding light on the unique contributions of each technique. This evaluation is not confined to a singular metric but extends to a comprehensive view, considering the models' performance from multiple angles. The project's strength lies not only in predicting stock prices but also in its ability to discern the most suitable model for diverse scenarios, contributing to the broader discourse on predictive analytics in financial markets.

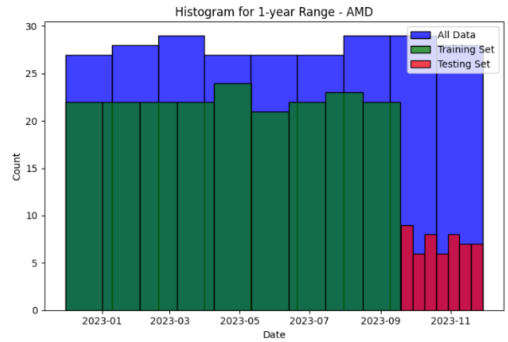
Data Preparation

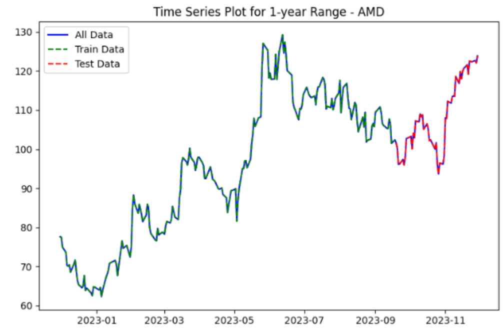
```
[3] import matplotlib.pyplot as plt
    import seaborn as sns
    import pandas as pd
    from sklearn.model_selection import train_test_split

    # Function to create train-test splits and perform analysis for Step 2
    def create_splits_and_visualizations_step2(data, company_name, time_ranges):
        # Creating a figure with subplots for visualizations
        fig, axes = plt.subplots(len(time_ranges), 2, figsize=(15, 5 * len(time_ranges)))
        for i, time_range in enumerate(time_ranges):
            # Extracting data for the specified time range
            subset_data = data[data.index >= (end - pd.DateOffset(years=time_range))]
            # Train-test split (80% train, 20% test)
            train_data, test_data = train_test_split(subset_data, test_size=0.2, shuffle=False)
            # Visualization 1: Histogram for all data, training set, and testing set
            sns.histplot(data=subset_data, x='Date', ax=axes[i, 0], color='blue', label='All Data')
            sns.histplot(data=train_data, x='Date', ax=axes[i, 0], color='green', label='Training Set', alpha=0.7)
            sns.histplot(data=test_data, x='Date', ax=axes[i, 0], color='red', label='Testing Set', alpha=0.7)
            axes[i, 0].set_title(f'Histogram for {time_range}-year Range - {company_name}')
            axes[i, 0].legend()
            # Visualization 2: Time series plot
            axes[i, 1].plot(subset_data.index, subset_data['Close'], label='All Data', color='blue')
            axes[i, 1].plot(train_data.index, train_data['Close'], label='Train Data', linestyle='--', color='green')
            axes[i, 1].plot(test_data.index, test_data['Close'], label='Test Data', linestyle='--', color='red')
            axes[i, 1].set_title(f'Time Series Plot for {time_range}-year Range - {company_name}')
            axes[i, 1].legend()
            # Printing information about the train-test split
            print(f"{company_name} - {time_range}-year Range:")
            print(f"  Training data size: {len(train_data)}")
            print(f"  Testing data size: {len(test_data)}")
            print(f"  Start date: {train_data.index.min().date()} - End date: {test_data.index.max().date()}")
        # Adjusting layout
        plt.tight_layout()
        plt.show()

    # Applying the function for each company
    for company, com_name in zip(company_list, ["AMD", "ELECTRONIC ARTS", "APPLE"]):
        full_range_years = (end - start).days // 365
        create_splits_and_visualizations_step2(company, com_name, [1, 3, 5, 10, full_range_years])
```

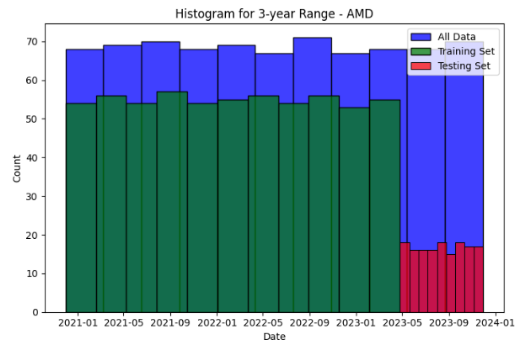
```
AMD - 1-year Range:
  Training data size: 200
  Testing data size: 51
  Start date: 2022-11-30 - End date: 2023-11-29
AMD - 3-year Range:
  Training data size: 604
  Testing data size: 151
  Start date: 2020-11-30 - End date: 2023-11-29
AMD - 5-year Range:
  Training data size: 1005
  Testing data size: 252
  Start date: 2018-11-30 - End date: 2023-11-29
AMD - 10-year Range:
  Training data size: 2012
  Testing data size: 504
  Start date: 2013-12-02 - End date: 2023-11-29
AMD - 22-year Range:
  Training data size: 4429
  Testing data size: 1108
  Start date: 2001-11-30 - End date: 2023-11-29
```

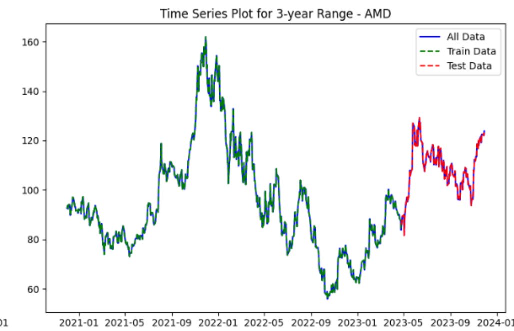
```
ELECTRONIC ARTS - 1-year Range:
  Training data size: 200
  Testing data size: 51
  Start date: 2022-11-30 - End date: 2023-11-29
ELECTRONIC ARTS - 3-year Range:
  Training data size: 604
  Testing data size: 151
  Start date: 2020-11-30 - End date: 2023-11-29
ELECTRONIC ARTS - 5-year Range:
  Training data size: 1005
  Testing data size: 252
  Start date: 2018-11-30 - End date: 2023-11-29
ELECTRONIC ARTS - 10-year Range:
  Training data size: 2012
  Testing data size: 504
  Start date: 2013-12-02 - End date: 2023-11-29
ELECTRONIC ARTS - 22-year Range:
  Training data size: 4429
  Testing data size: 1108
  Start date: 2001-11-30 - End date: 2023-11-29
```

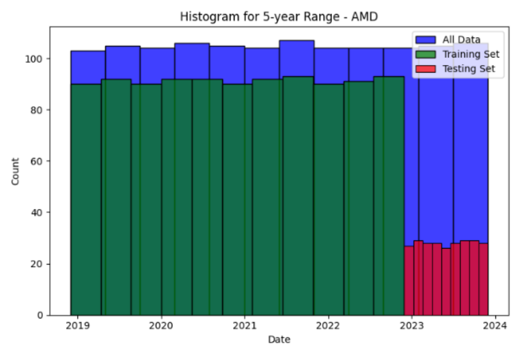
```
APPLE - 1-year Range:
  Training data size: 200
  Testing data size: 51
  Start date: 2022-11-30 - End date: 2023-11-29
APPLE - 3-year Range:
  Training data size: 604
  Testing data size: 151
  Start date: 2020-11-30 - End date: 2023-11-29
APPLE - 5-year Range:
  Training data size: 1005
  Testing data size: 252
  Start date: 2018-11-30 - End date: 2023-11-29
APPLE - 10-year Range:
  Training data size: 2012
  Testing data size: 504
  Start date: 2013-12-02 - End date: 2023-11-29
APPLE - 22-year Range:
  Training data size: 4429
  Testing data size: 1108
  Start date: 2001-11-30 - End date: 2023-11-29
```

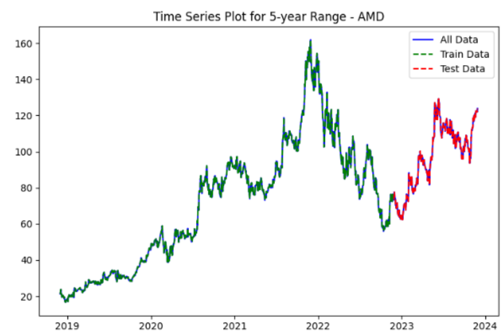

Histogram for 1-year Range - AMD
All Data
Training Set
Testing Set
Count
Date

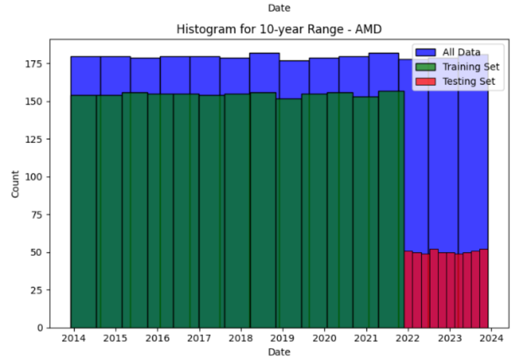

Time Series Plot for 1-year Range - AMD
All Data
Train Data
Test Data


Histogram for 3-year Range - AMD
All Data
Training Set
Testing Set
Count
Date

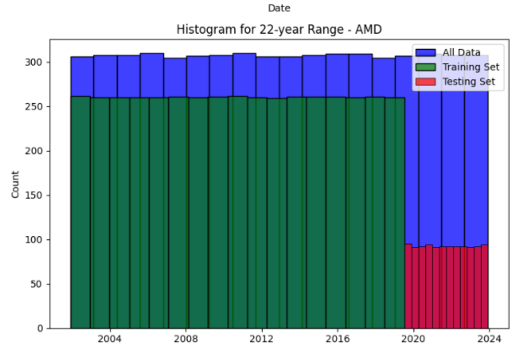

Time Series Plot for 3-year Range - AMD
All Data
Train Data
Test Data

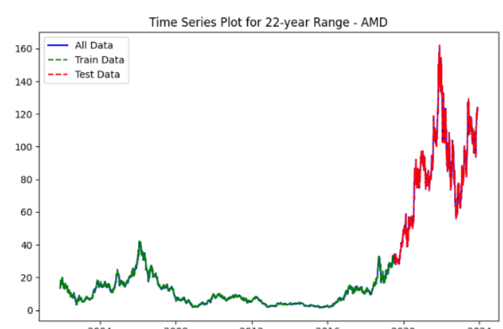

Histogram for 5-year Range - AMD
All Data
Training Set
Testing Set
Count
Date

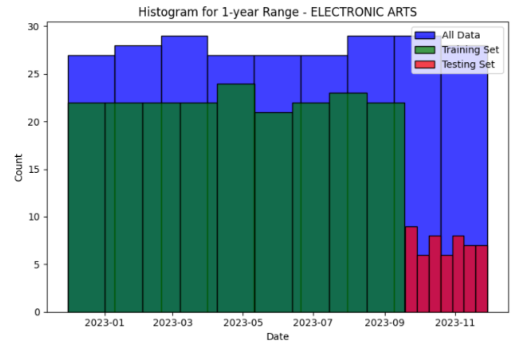

Time Series Plot for 5-year Range - AMD
All Data
Train Data
Test Data

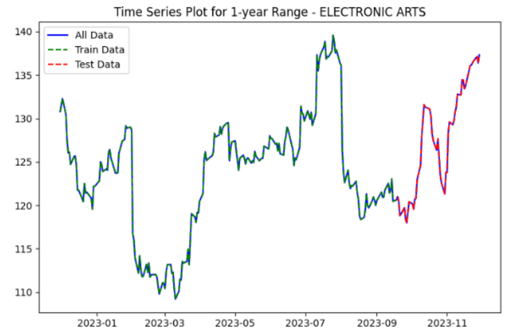

Histogram for 10-year Range - AMD
All Data
Training Set
Testing Set
Count
Date

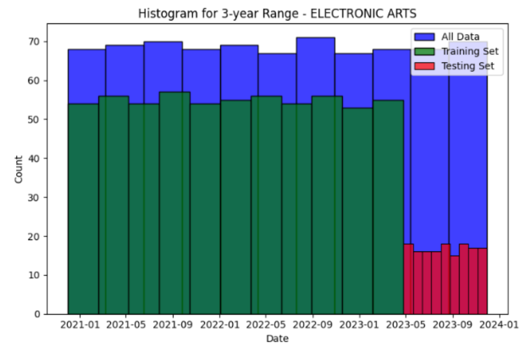

Time Series Plot for 10-year Range - AMD
All Data
Train Data
Test Data

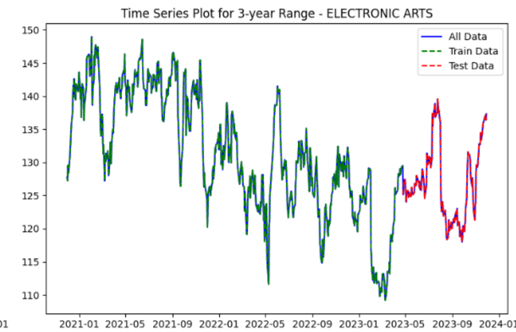

Histogram for 22-year Range - AMD
All Data
Training Set
Testing Set
Count
Date

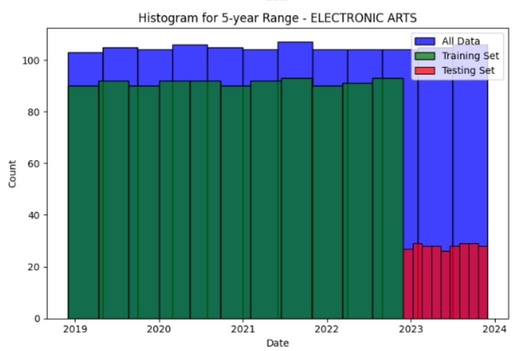

Time Series Plot for 22-year Range - AMD
All Data
Train Data
Test Data

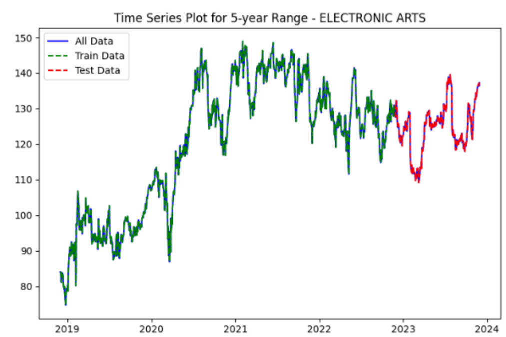

Histogram for 1-year Range - ELECTRONIC ARTS
All Data
Training Set
Testing Set
Count
Date

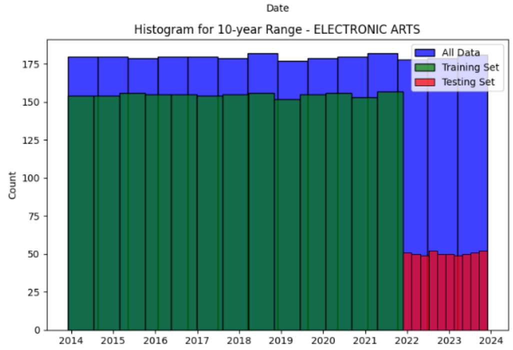

Time Series Plot for 1-year Range - ELECTRONIC ARTS
All Data
Train Data
Test Data

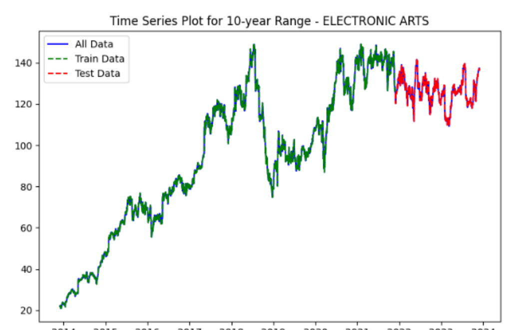

Histogram for 3-year Range - ELECTRONIC ARTS
All Data
Training Set
Testing Set
Count
Date

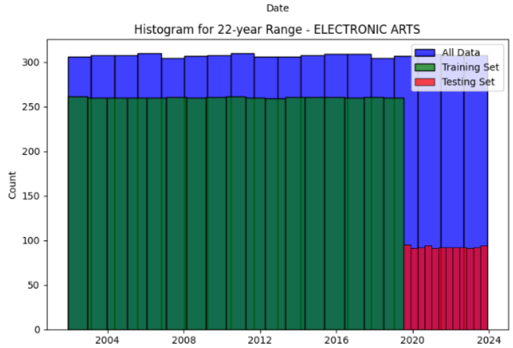

Time Series Plot for 3-year Range - ELECTRONIC ARTS
All Data
Train Data
Test Data

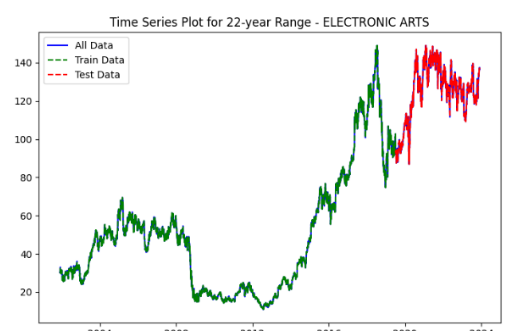

Histogram for 5-year Range - ELECTRONIC ARTS
All Data
Training Set
Testing Set
Count
Date

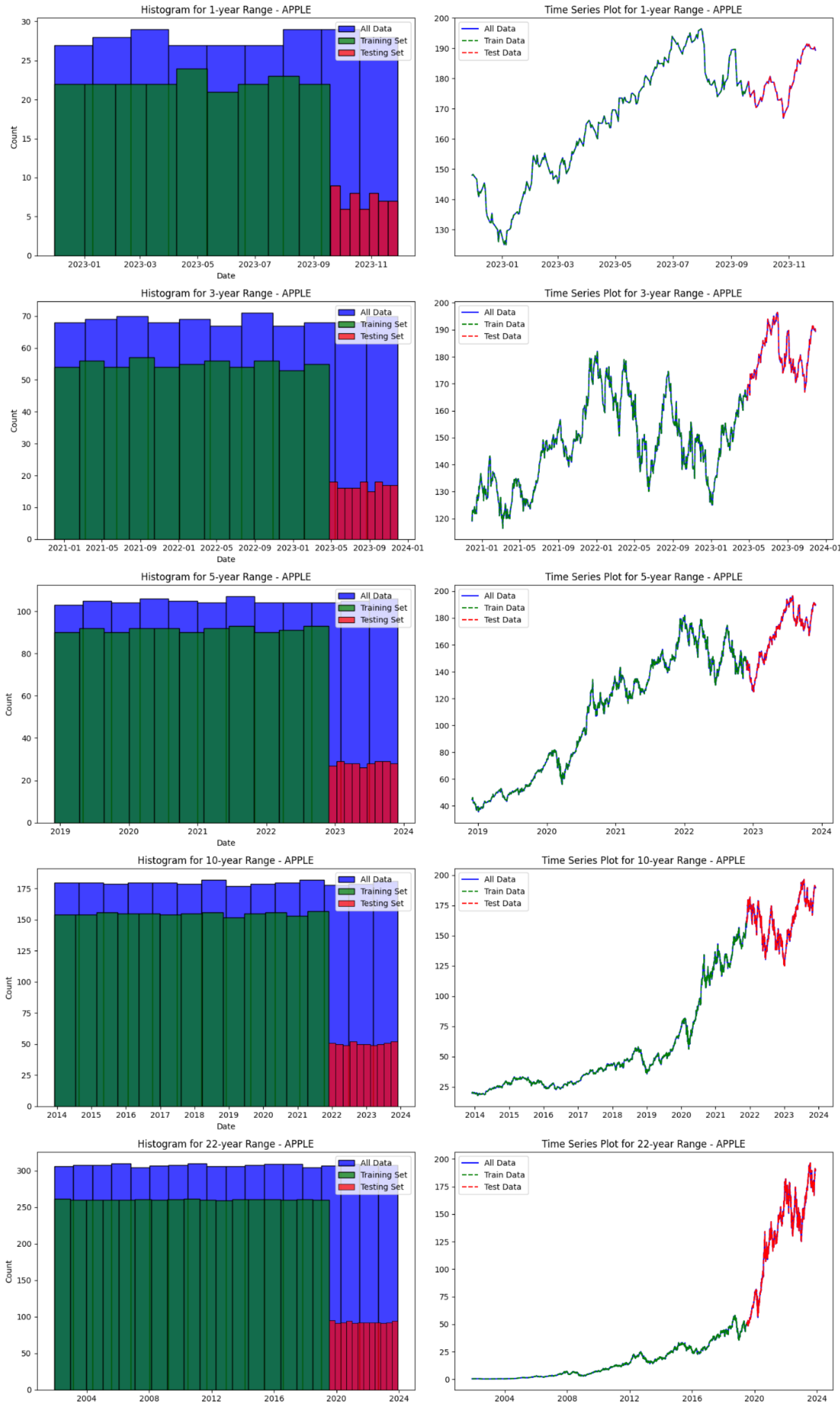

Time Series Plot for 5-year Range - ELECTRONIC ARTS
All Data
Train Data
Test Data


Histogram for 10-year Range - ELECTRONIC ARTS
All Data
Training Set
Testing Set
Count
Date


Time Series Plot for 10-year Range - ELECTRONIC ARTS
All Data
Train Data
Test Data


Histogram for 22-year Range - ELECTRONIC ARTS
All Data
Training Set
Testing Set
Count
Date


Time Series Plot for 22-year Range - ELECTRONIC ARTS
All Data
Train Data
Test Data



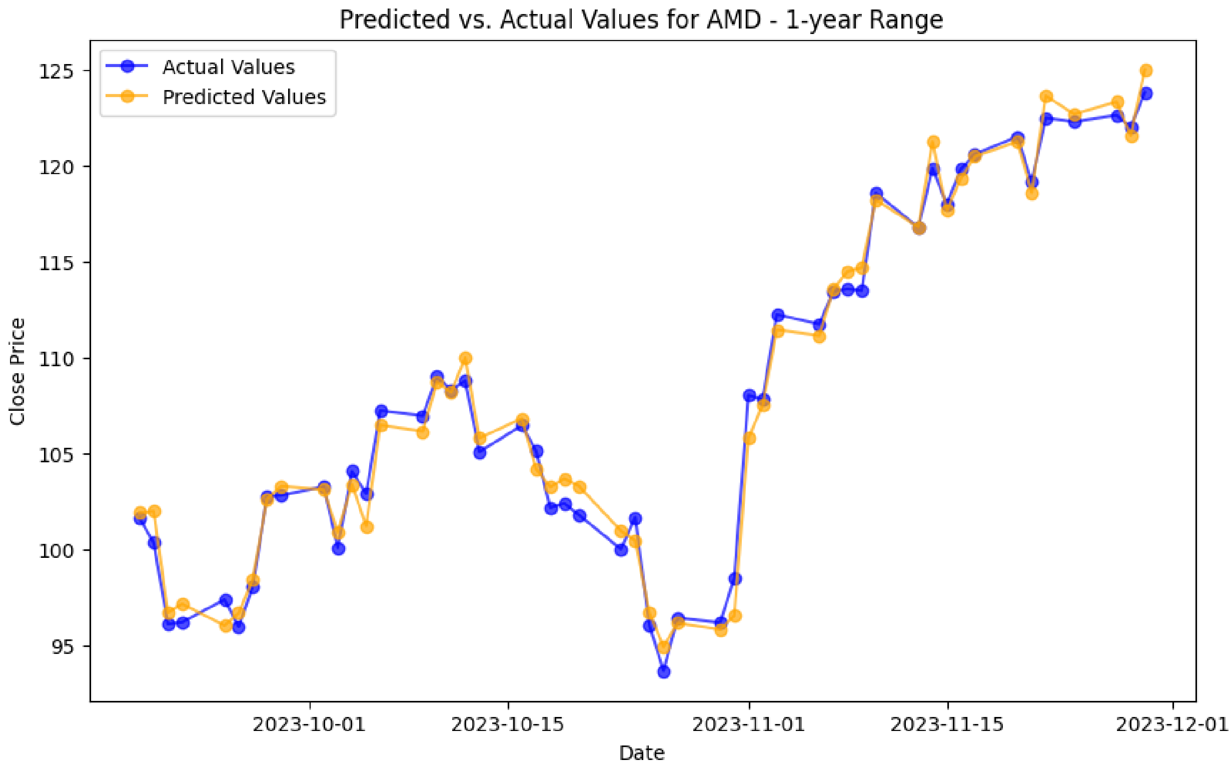
Histogram for 1-year Range - APPLE
Time Series Plot for 1-year Range - APPLE
Histogram for 3-year Range - APPLE
Time Series Plot for 3-year Range - APPLE
Histogram for 5-year Range - APPLE
Time Series Plot for 5-year Range - APPLE
Histogram for 10-year Range - APPLE
Time Series Plot for 10-year Range - APPLE
Histogram for 22-year Range - APPLE
Time Series Plot for 22-year Range - APPLE
Count
Date
All Data
Training Set
Testing Set
All Data
Train Data
Test Data

Model, Training & Prediction

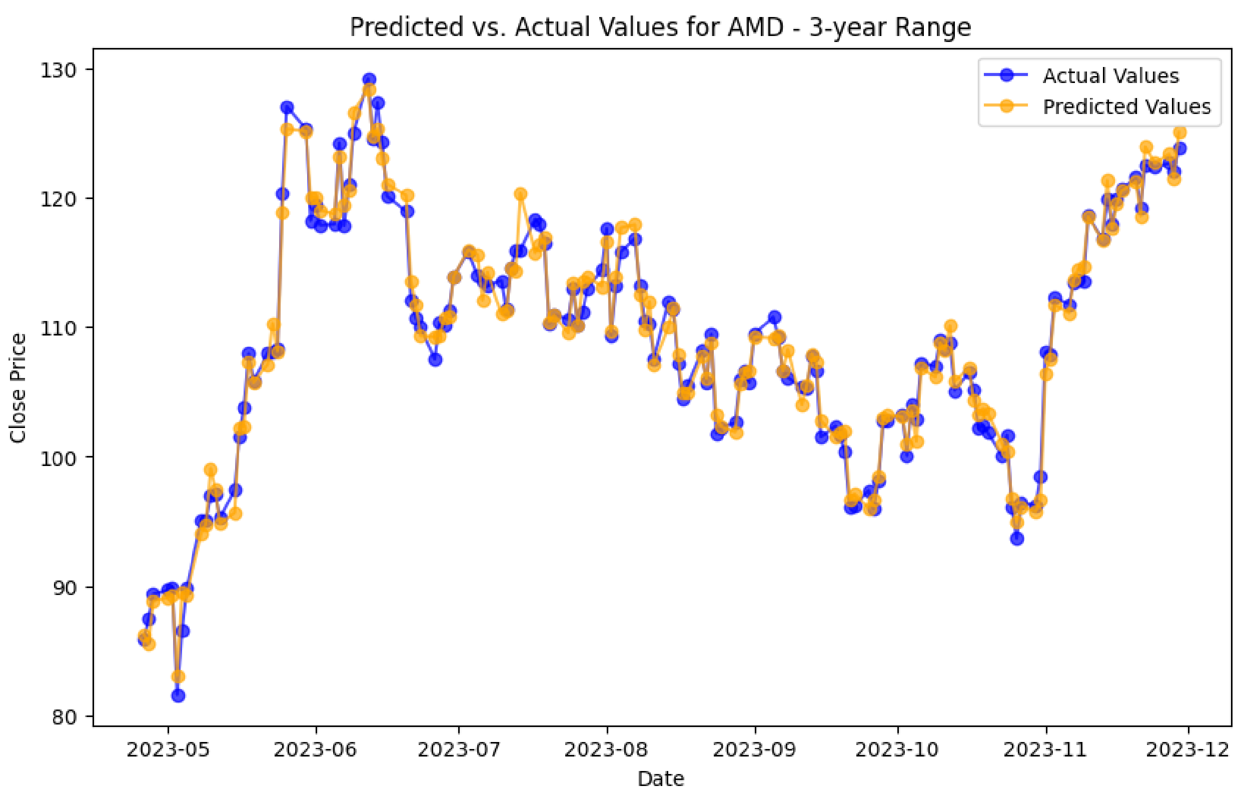
```python
# Importing necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor, plot_tree
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import LSTM, Dense
import shap
import matplotlib.pyplot as plt
# Function to calculate mean absolute percentage error
def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
# Function for model training and evaluation
def train_and_evaluate_models(train_data, test_data, model_type, company_name, time_range):
    X_train, y_train = train_data[['Open', 'High', 'Low', 'Volume']], train_data['Close']
    X_test, y_test = test_data[['Open', 'High', 'Low', 'Volume']], test_data['Close']
    # Model selection
    if model_type == 'Linear Regression':
        model = LinearRegression()
    elif model_type == 'Decision Trees':
        model = DecisionTreeRegressor(random_state=42)
    elif model_type == 'LSTM':
        # LSTM requires scaled input data
        scaler = MinMaxScaler(feature_range=(0, 1))
        X_train_scaled = scaler.fit_transform(X_train)
        X_test_scaled = scaler.transform(X_test)
        # Reshaping the input data for LSTM
        X_train_reshaped = X_train_scaled.reshape((X_train_scaled.shape[0], 1, X_train_scaled.shape[1]))
        X_test_reshaped = X_test_scaled.reshape((X_test_scaled.shape[0], 1, X_test_scaled.shape[1]))
        # Defining the LSTM model
        model = Sequential()
        model.add(LSTM(50, activation='relu', input_shape=(1, X_train_scaled.shape[1])))
        model.add(Dense(1))
        model.compile(optimizer='adam', loss='mse')
    # Training the model
    if model_type != 'LSTM':
        model.fit(X_train, y_train)
    else:
        history = model.fit(X_train_reshaped, y_train, epochs=50, batch_size=32, verbose=0, validation_split=0.1)
        # Plotting training loss curve
        plt.figure(figsize=(10, 6))
        plt.plot(history.history['loss'], label='Training Loss', marker='o', linestyle='-', color='blue')
        plt.plot(history.history['val_loss'], label='Validation Loss', marker='o', linestyle='-', color='orange')
        plt.title(f'Training Loss Curve for {company_name} - {time_range}-year Range')
        plt.xlabel('Epoch')
        plt.ylabel('Loss')
        plt.legend()
        plt.show()
    # Making predictions
    y_pred = model.predict(X_test) if model_type != 'LSTM' else model.predict(X_test_reshaped)
```

Linear Regression Evaluations

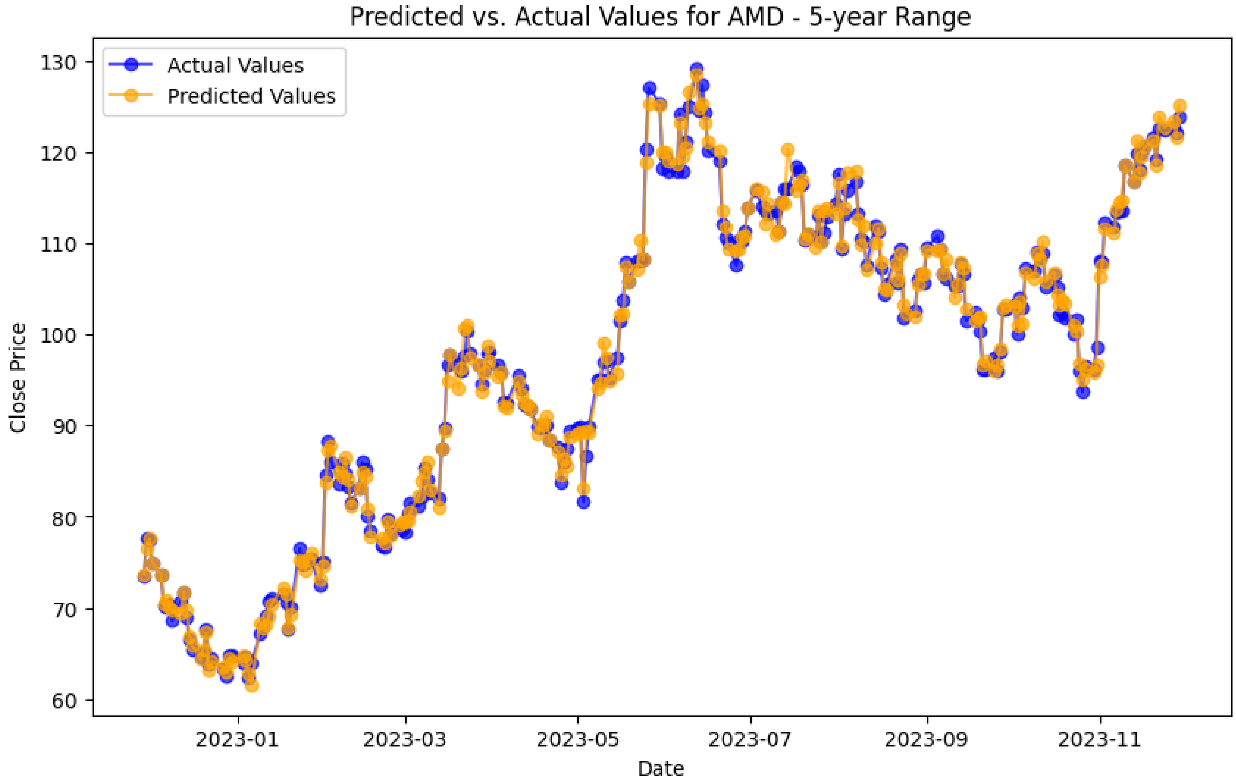
AMD - 1 -year Range - Linear Regression Evaluation:



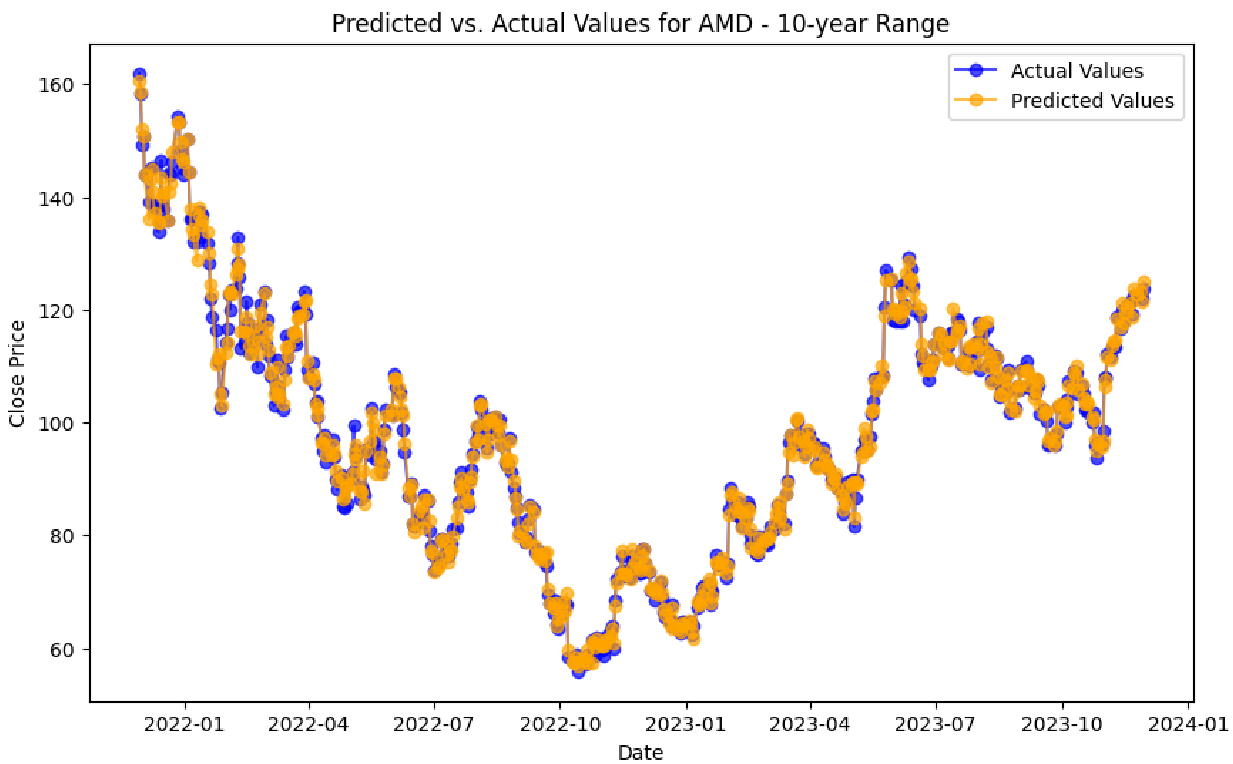
AMD - 3 -year Range - Linear Regression Evaluation:



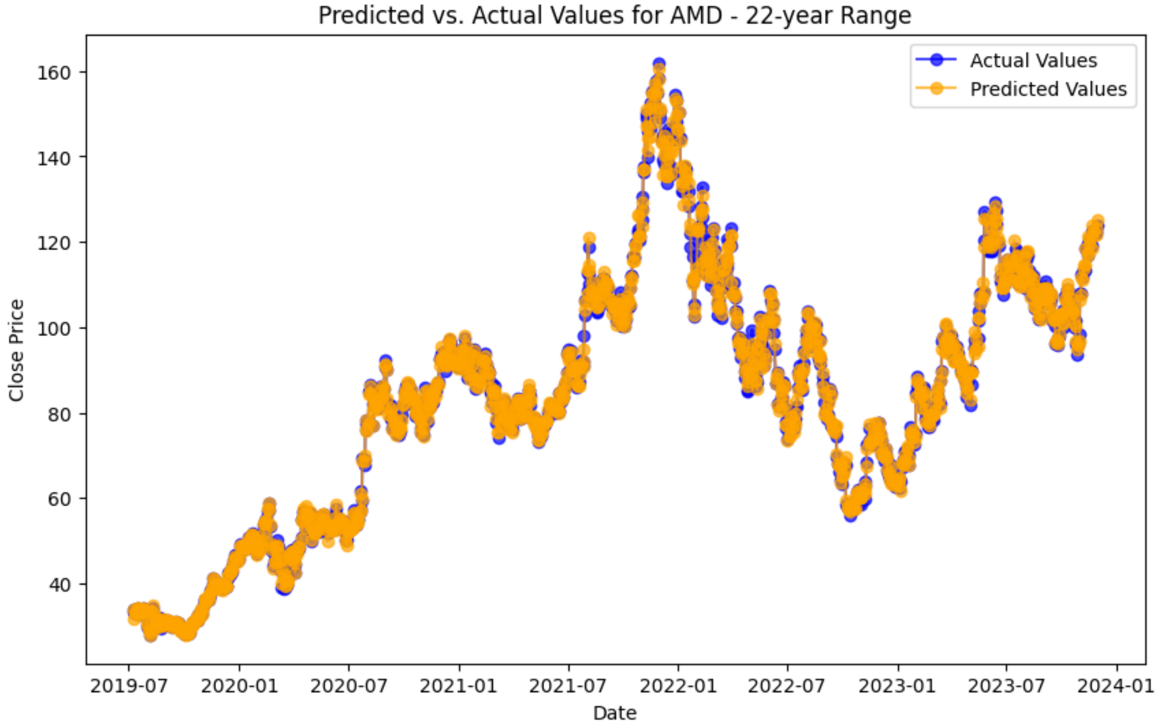
AMD - 5 -year Range - Linear Regression Evaluation:



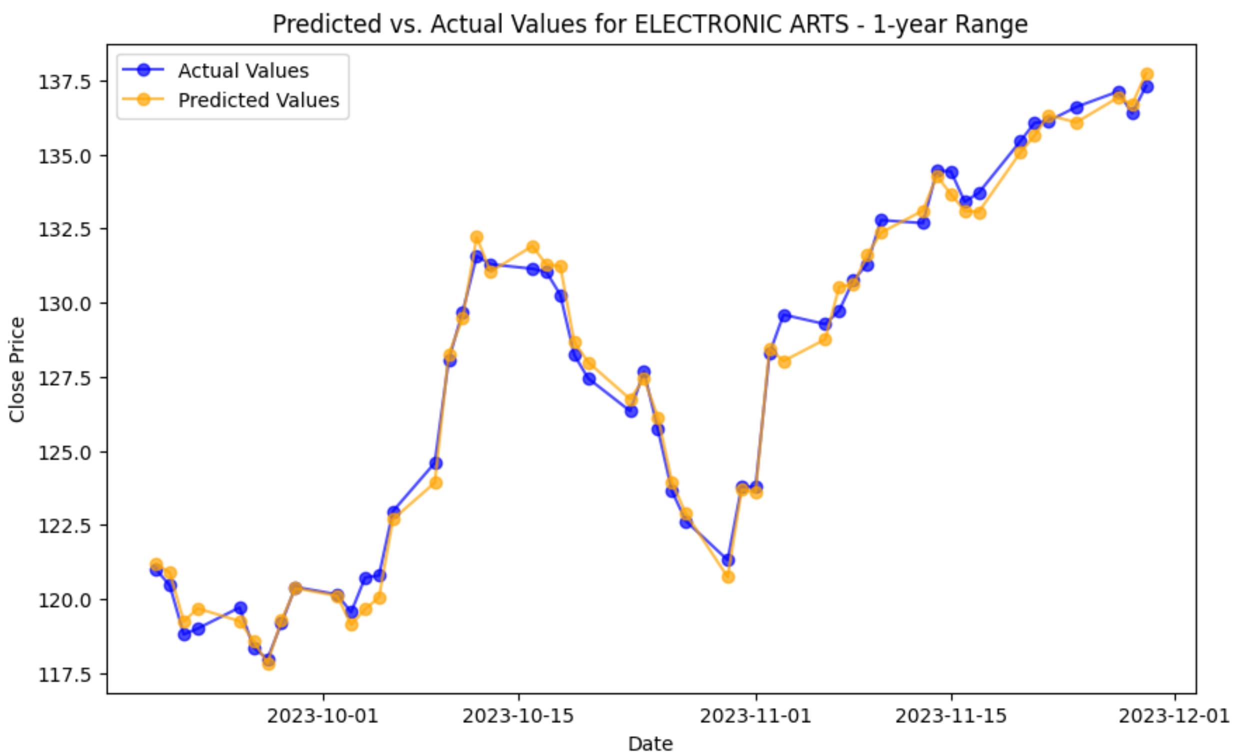
AMD - 10 -year Range - Linear Regression Evaluation:



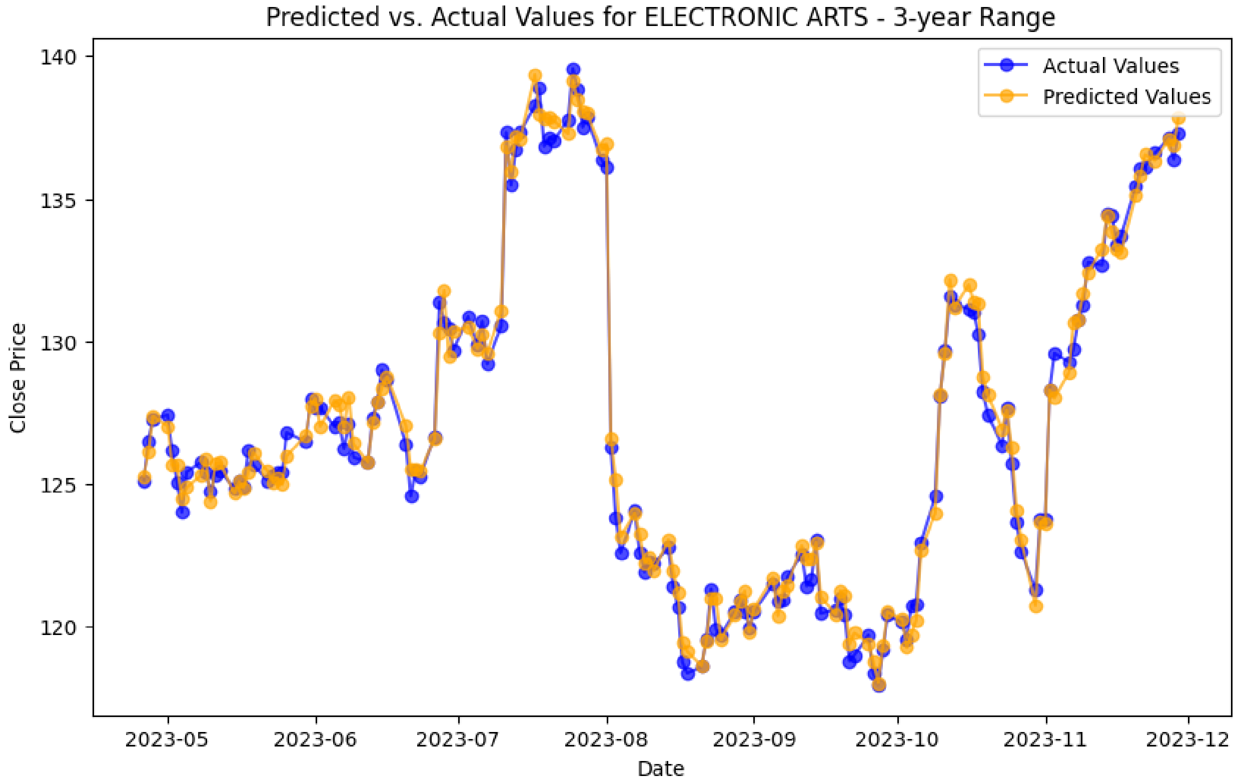
AMD - 22 -year Range - Linear Regression Evaluation:



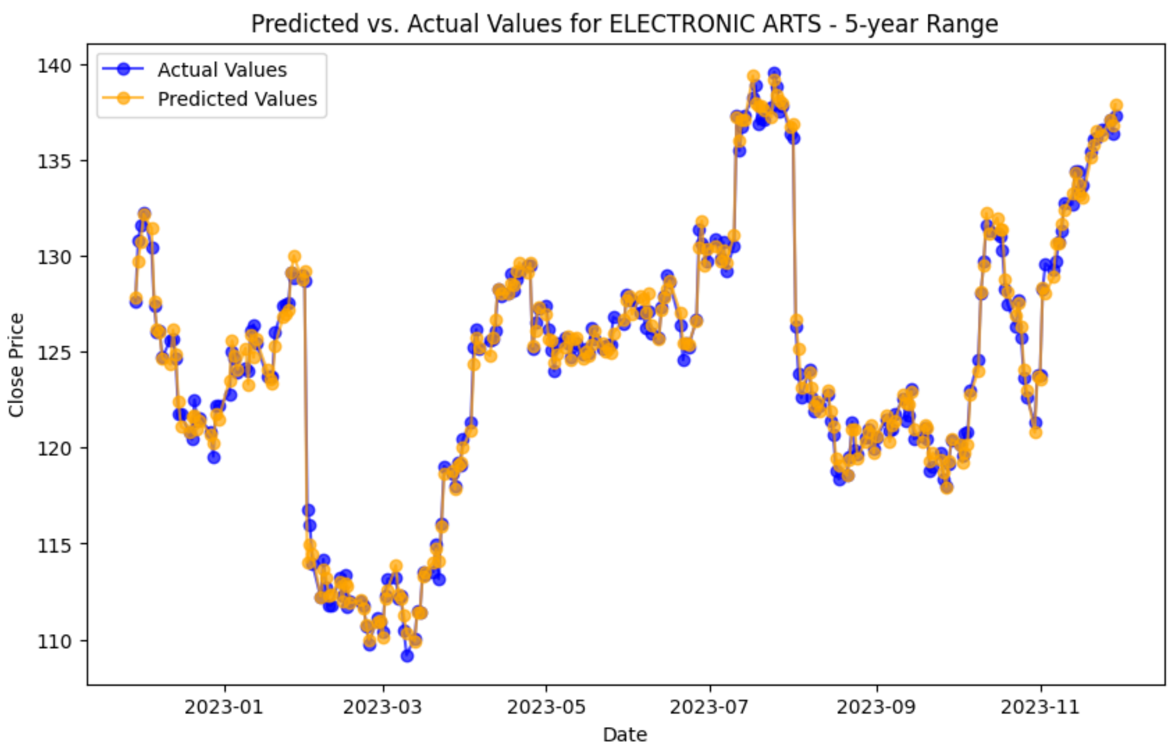
EA - 1 -year Range - Linear Regression Evaluation:



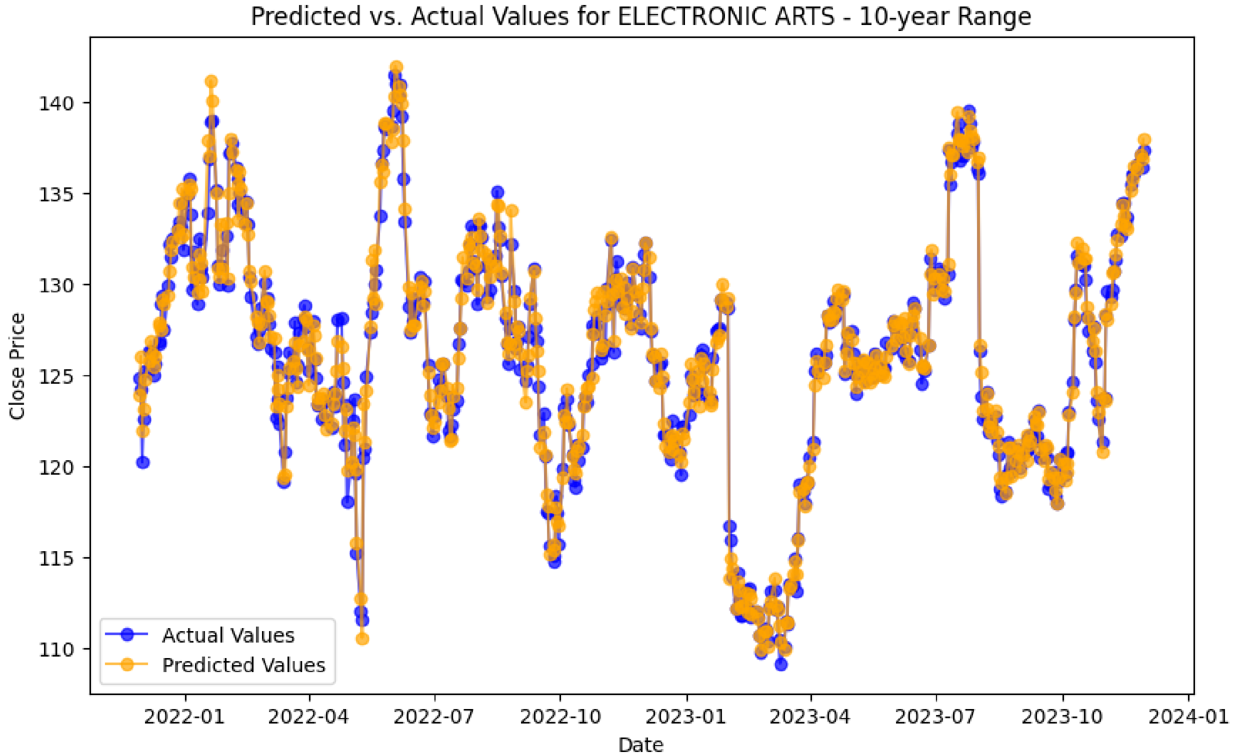
EA - 3 -year Range - Linear Regression Evaluation:



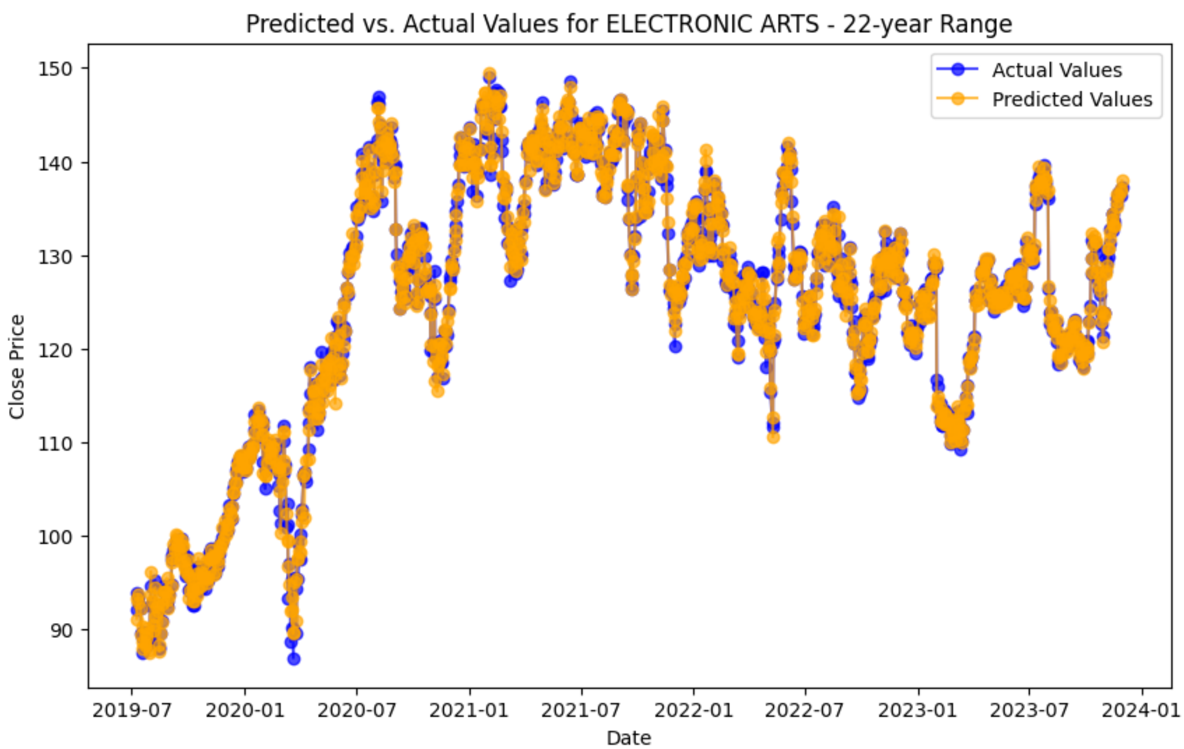
EA - 5 -year Range - Linear Regression Evaluation:



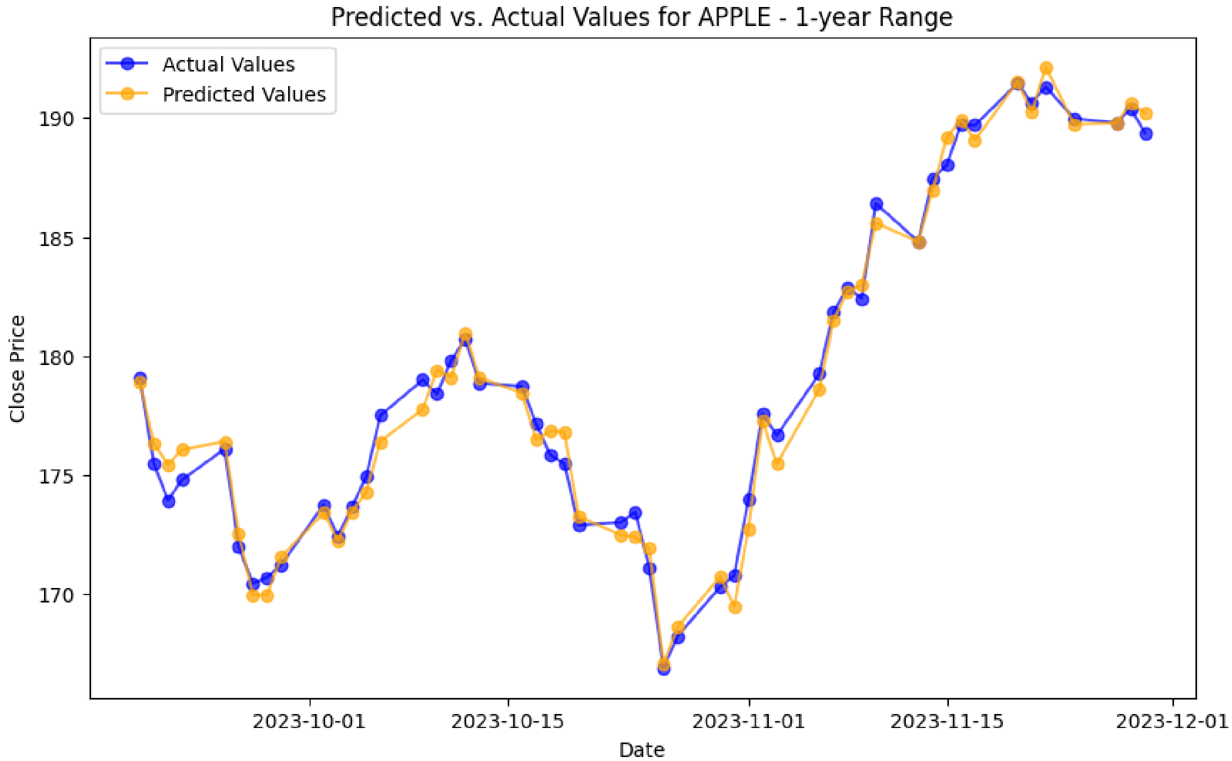
EA - 10 -year Range - Linear Regression Evaluation:



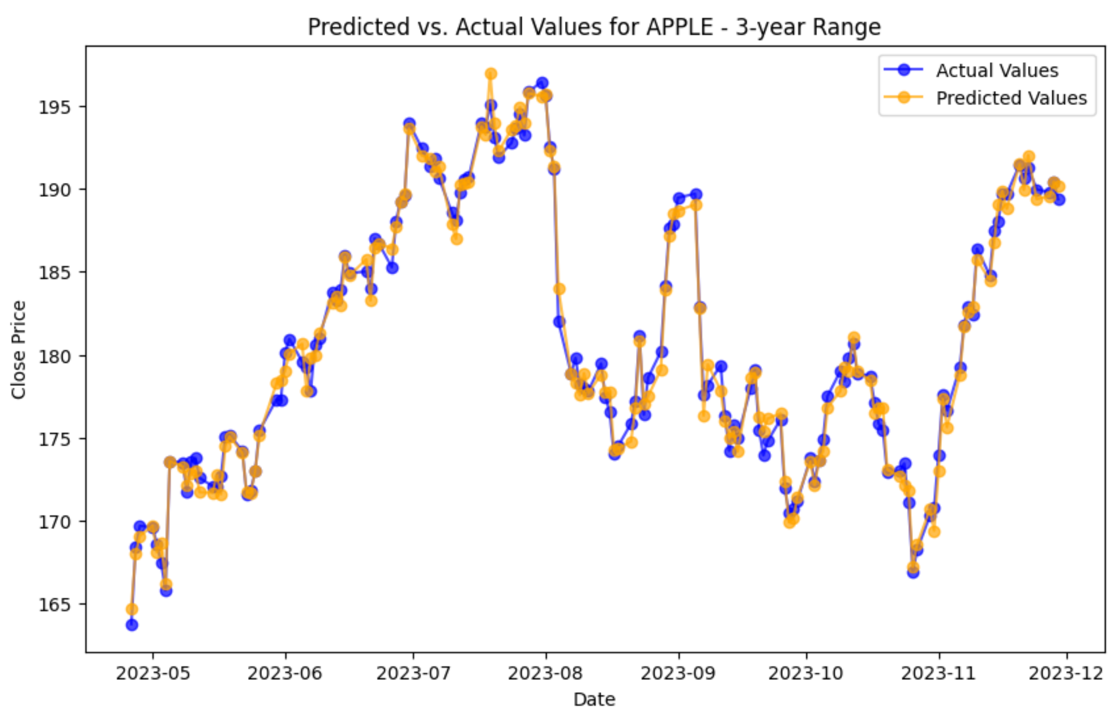
EA - 22 -year Range - Linear Regression Evaluation:



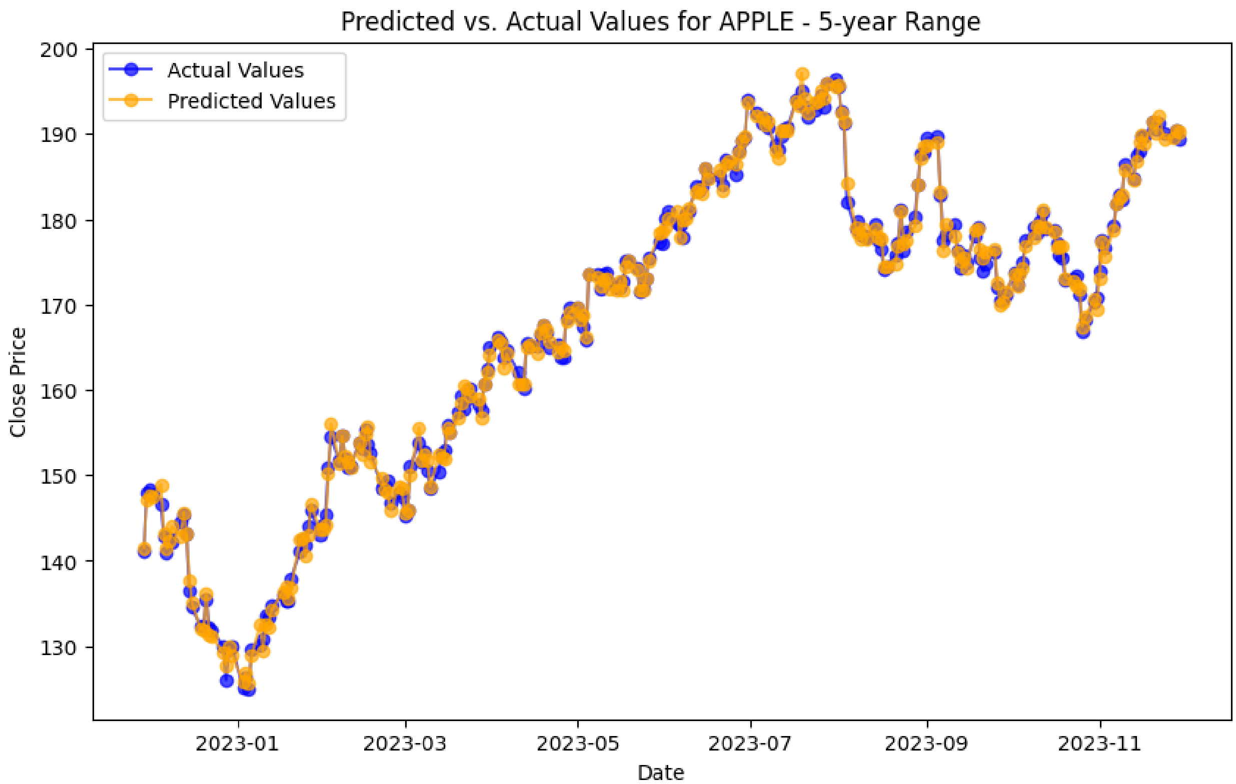
Apple - 1 -year Range - Linear Regression Evaluation:



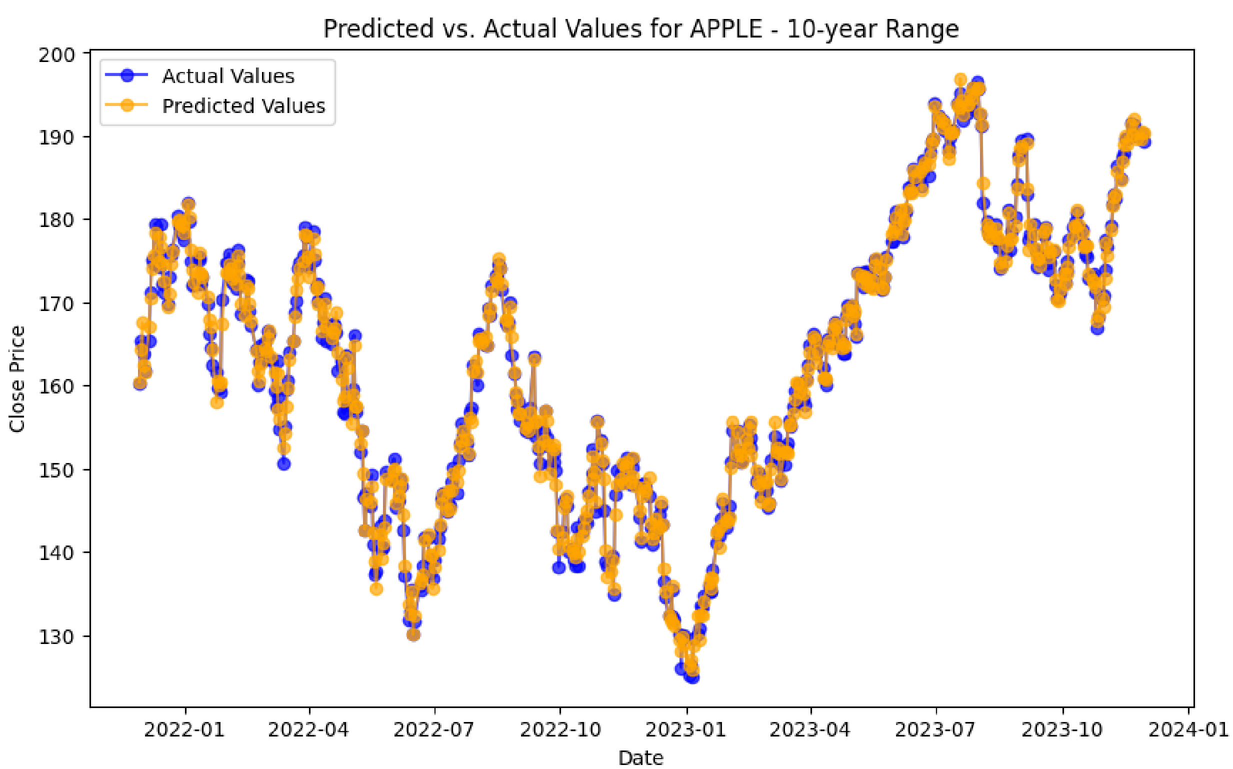
Apple - 3 -year Range - Linear Regression Evaluation:



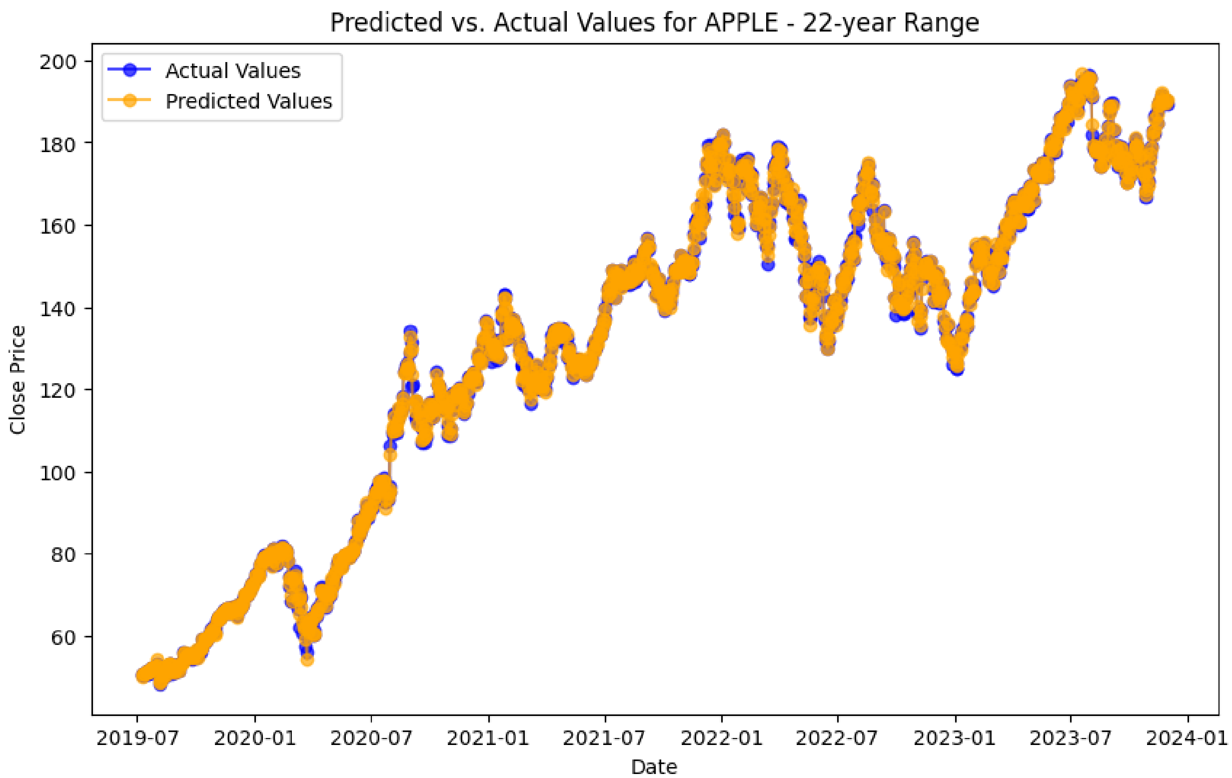
Apple - 5 -year Range - Linear Regression Evaluation:



Apple - 10 -year Range - Linear Regression Evaluation:

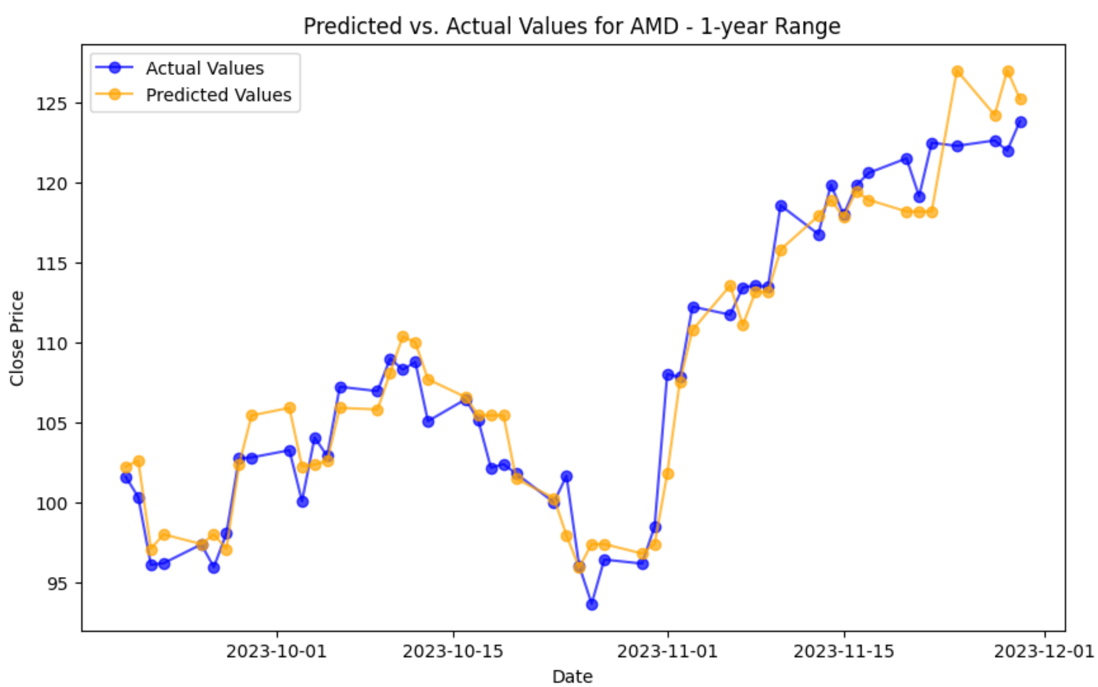


Apple - 22 -year Range - Linear Regression Evaluation:

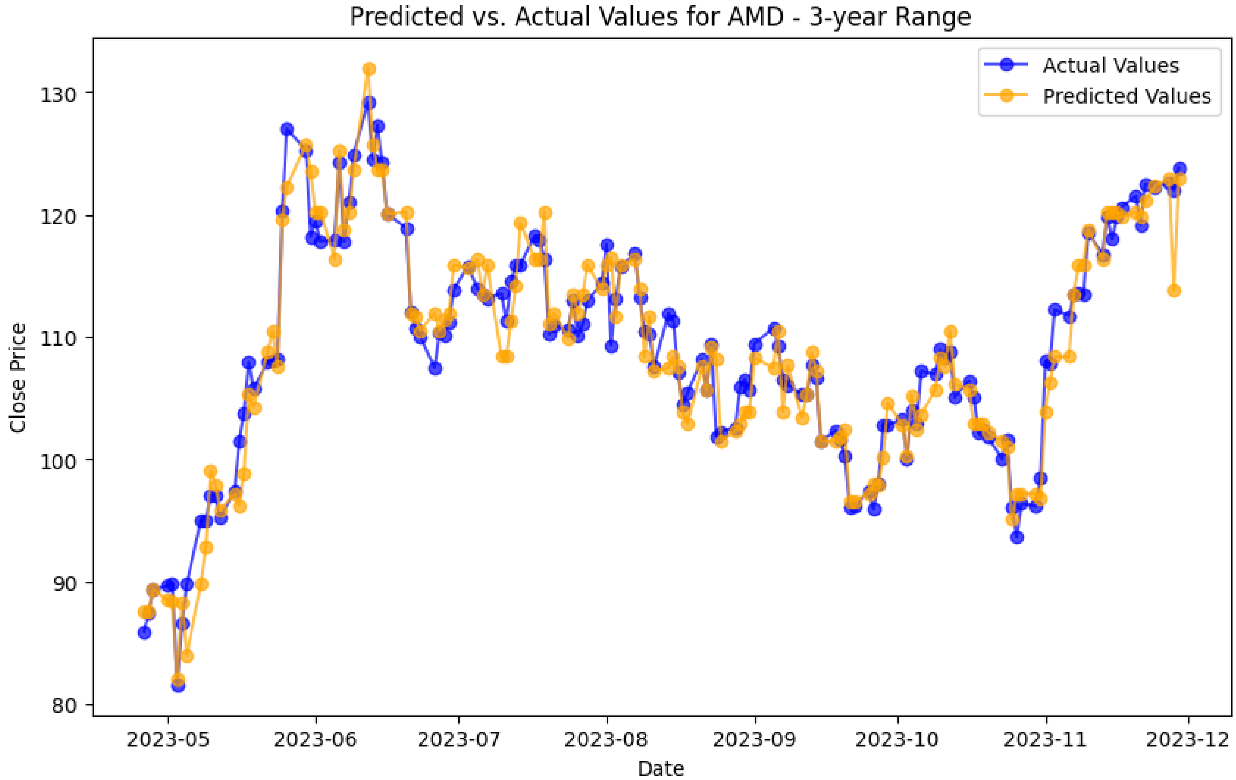


Decision Trees Evaluation

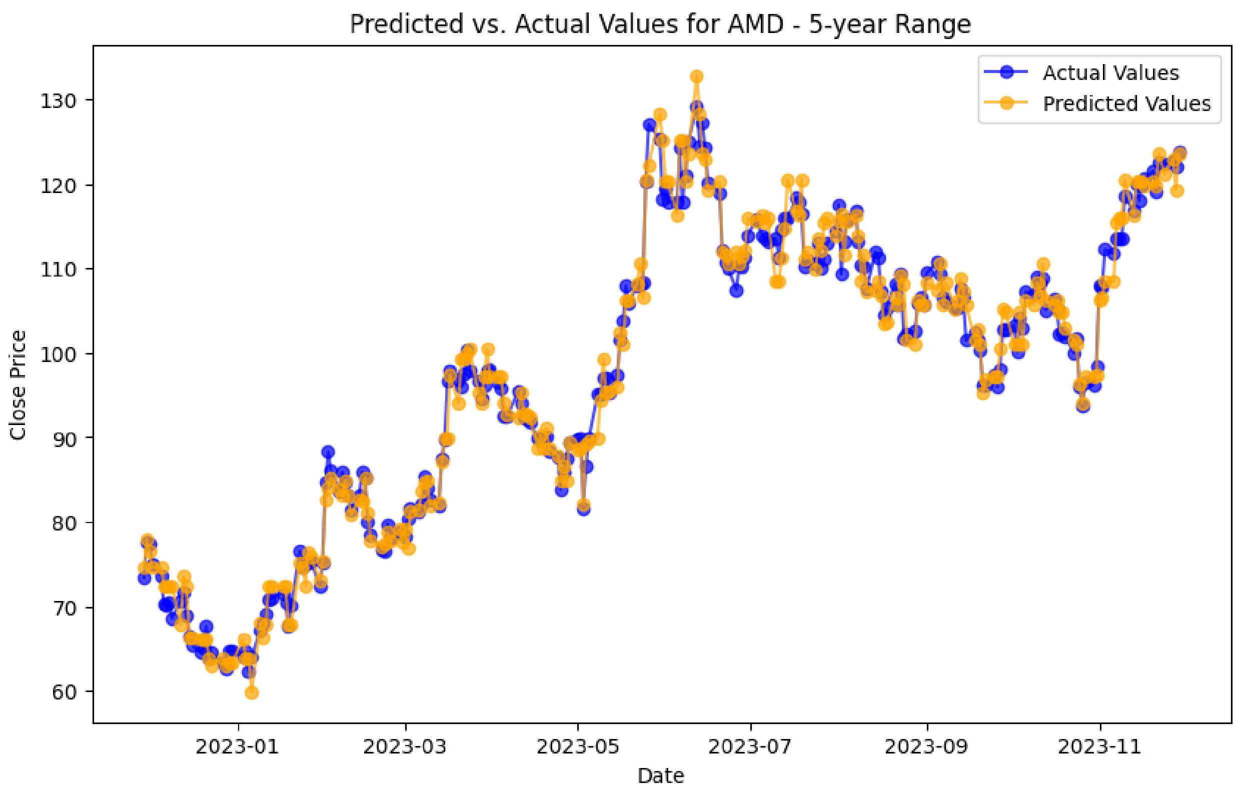
AMD - 1 -year Range - Decision Trees Evaluation:



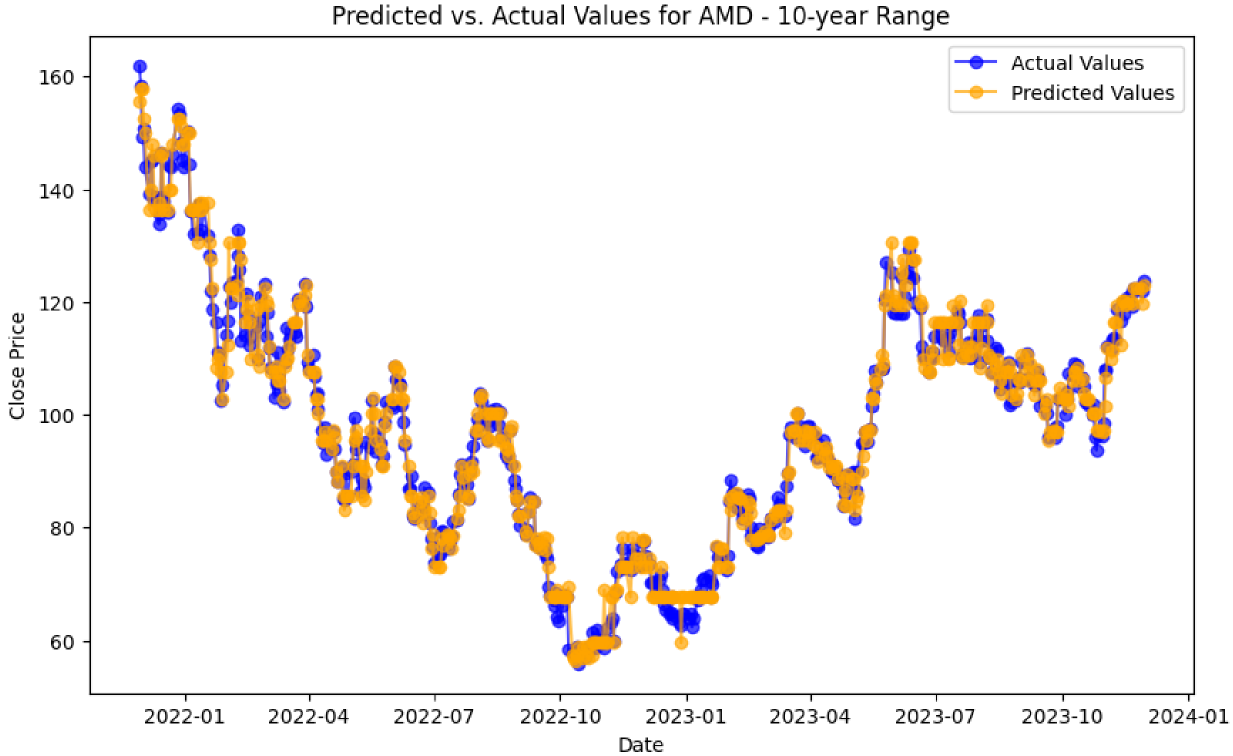
AMD - 3 -year Range - Decision Trees Evaluation:



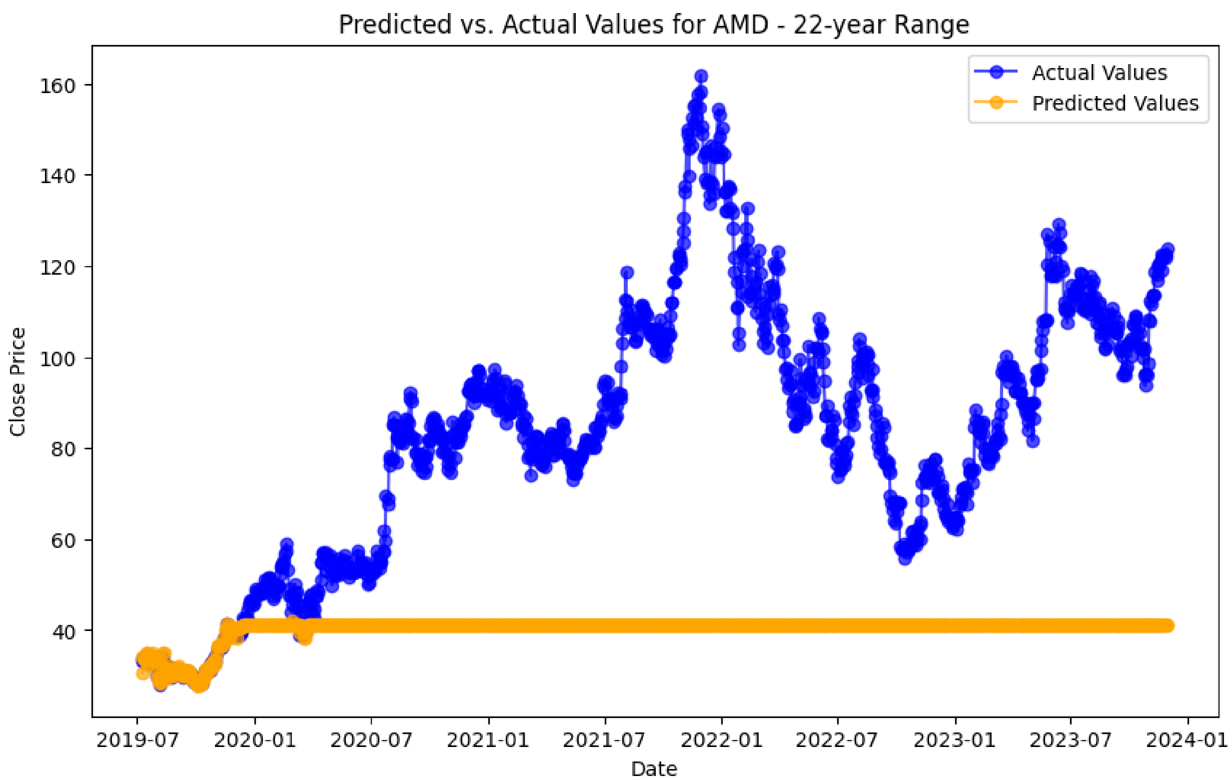
AMD - 5 -year Range - Decision Trees Evaluation:



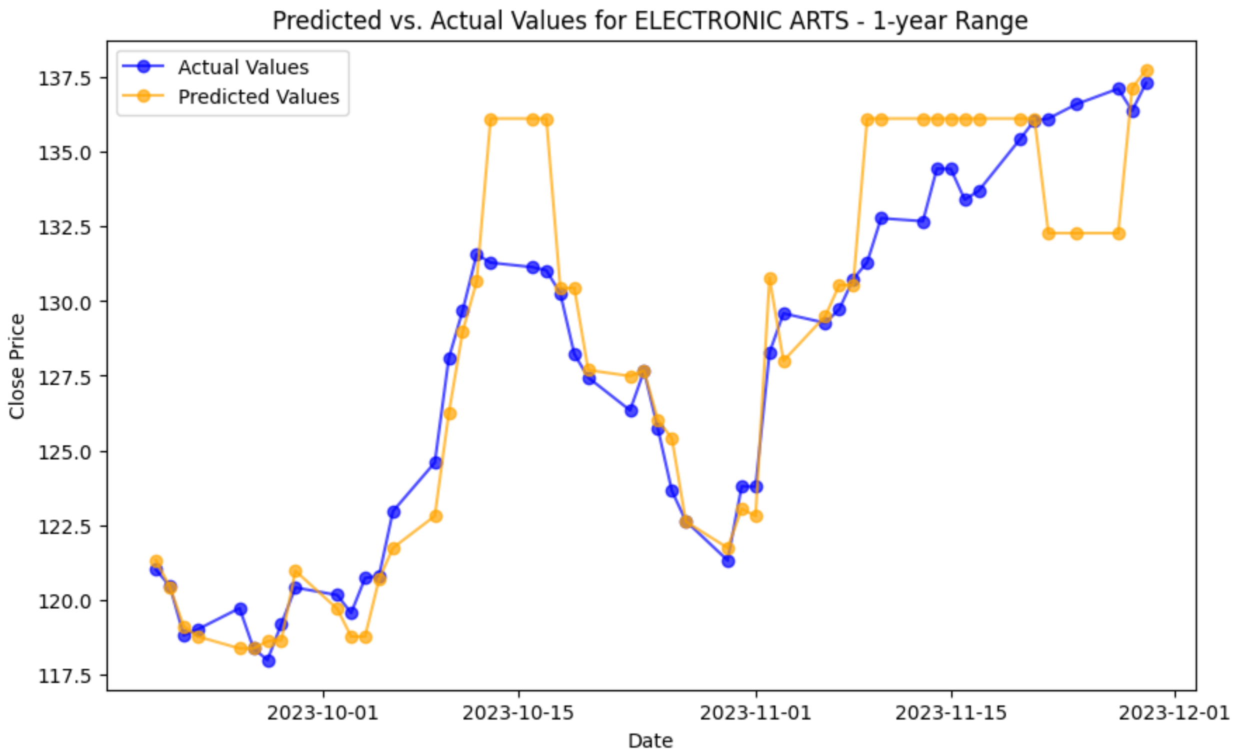
AMD - 10 -year Range - Decision Trees Evaluation:



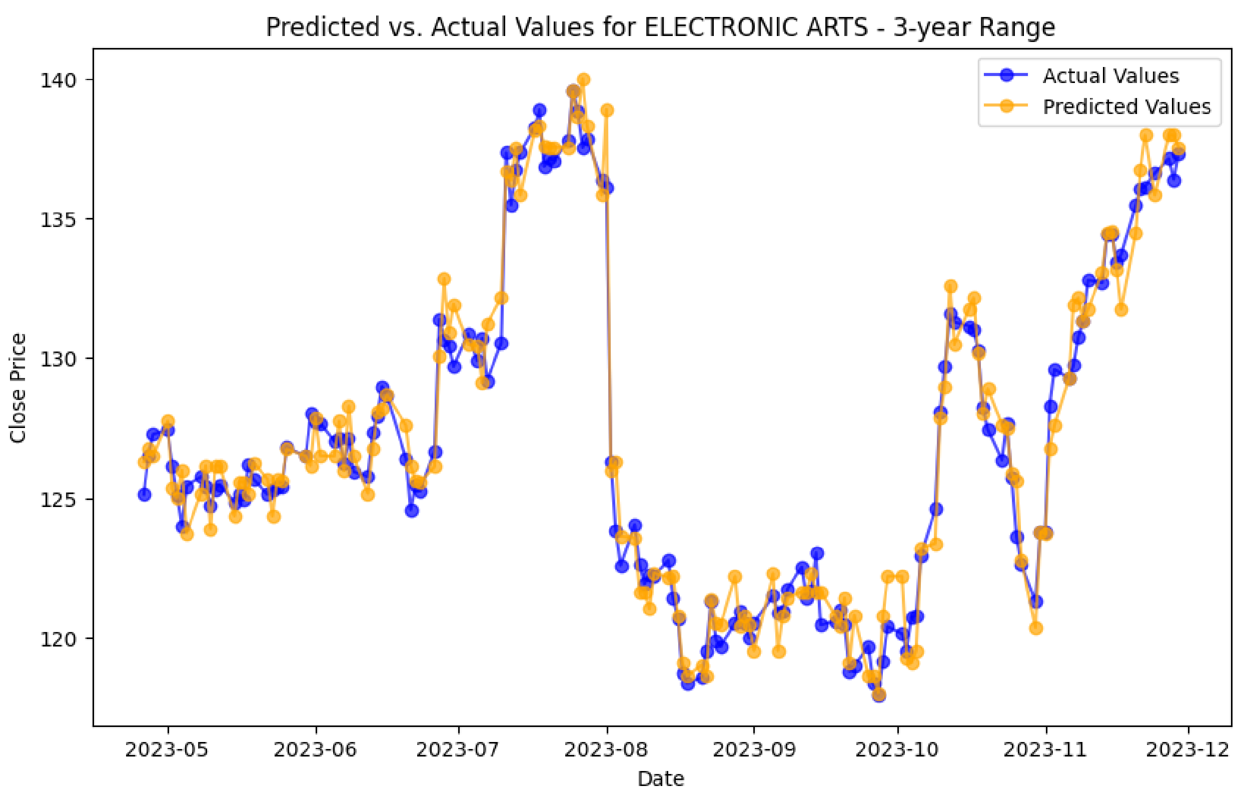
AMD - 22 -year Range - Decision Trees Evaluation:



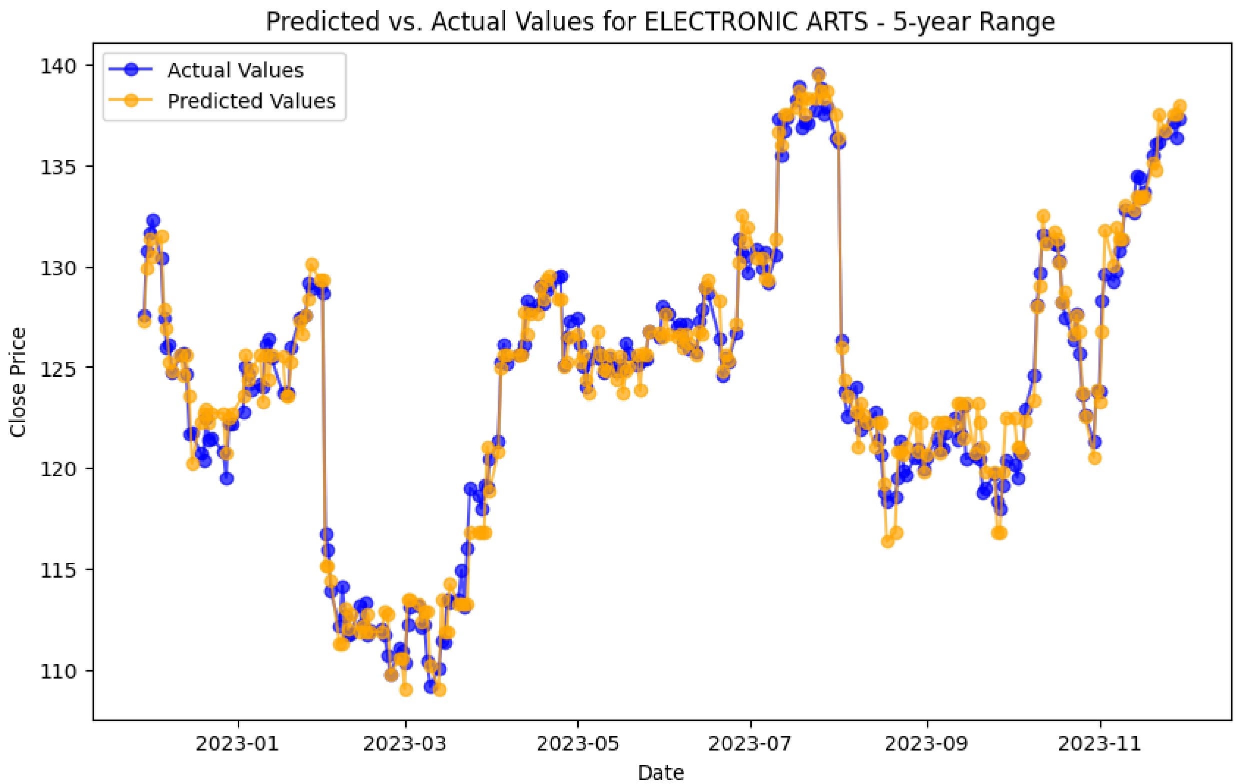
EA - 1 -year Range - Decision Trees Evaluation:



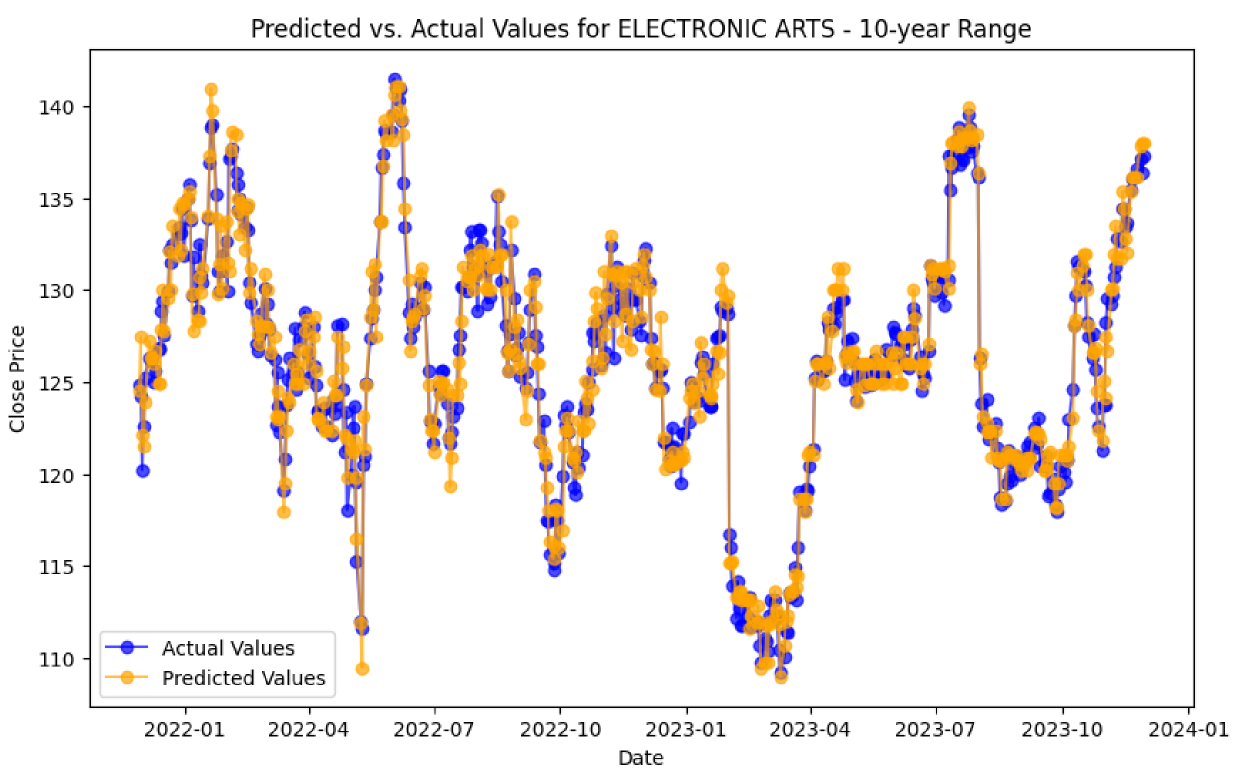
EA - 3 -year Range - Decision Trees Evaluation:



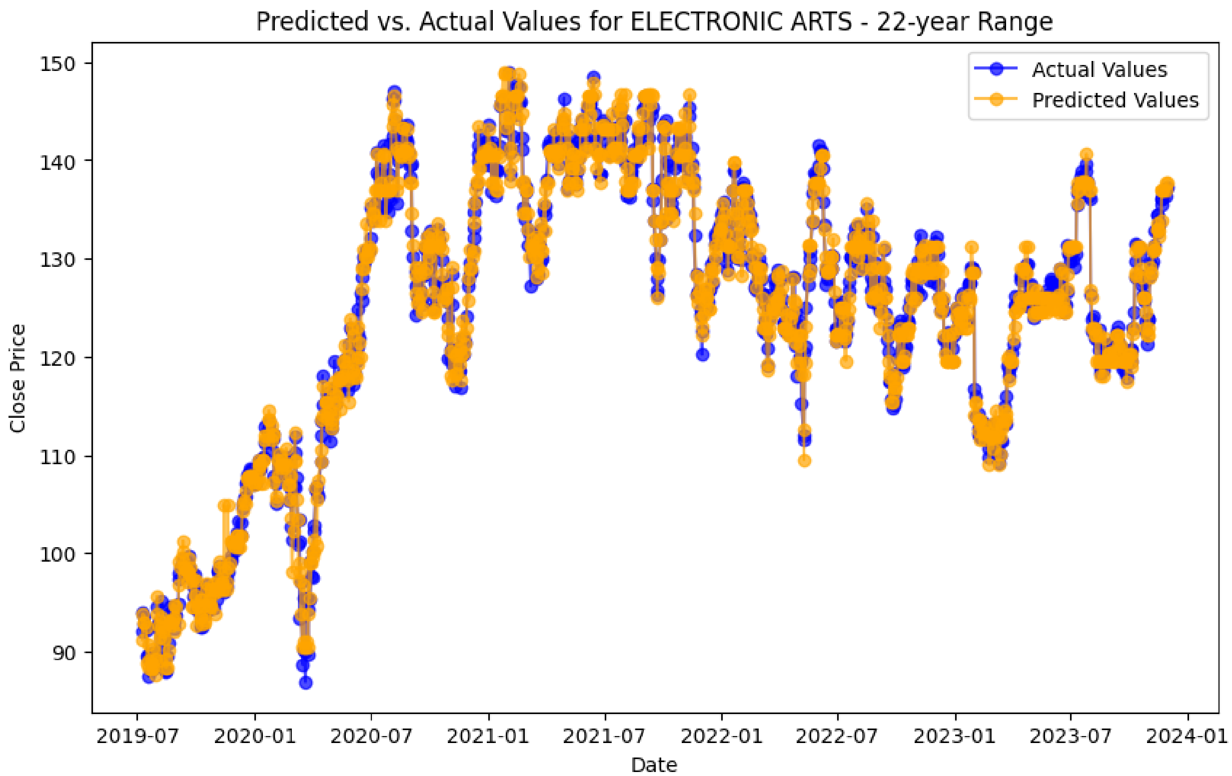
EA - 5 -year Range - Decision Trees Evaluation:



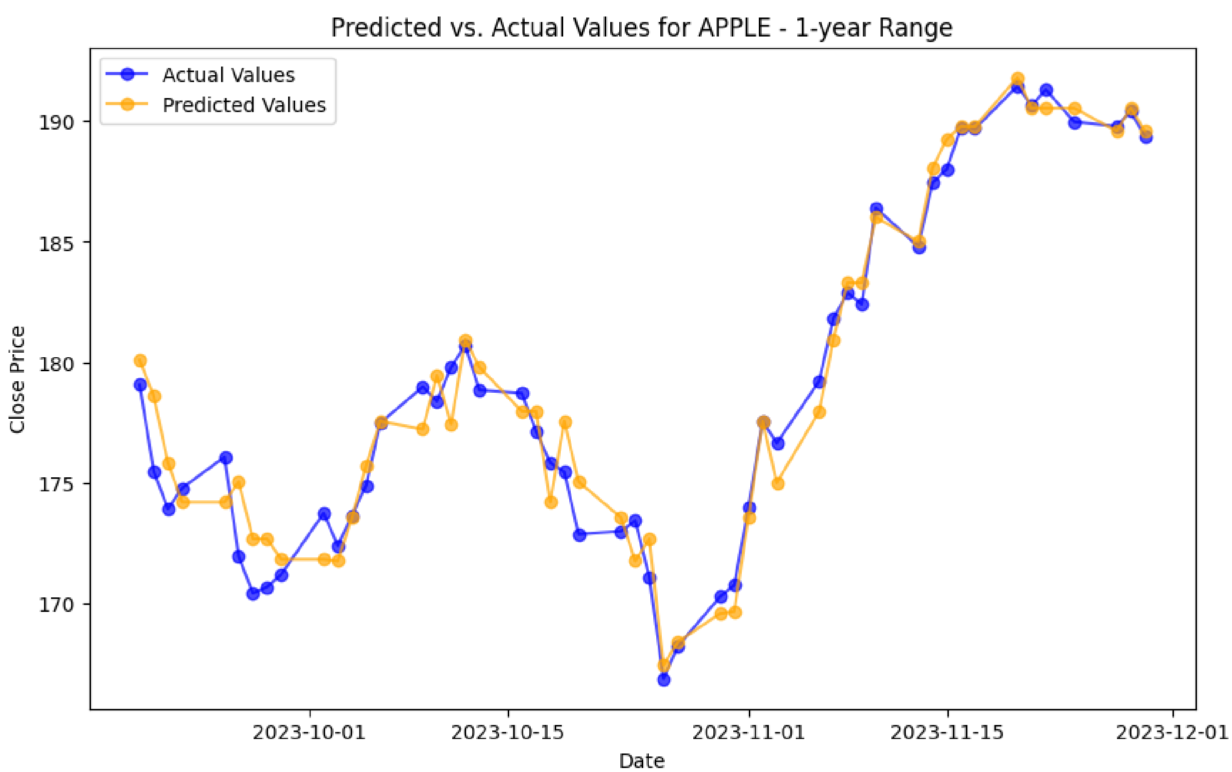
EA - 10 -year Range - Decision Trees Evaluation:



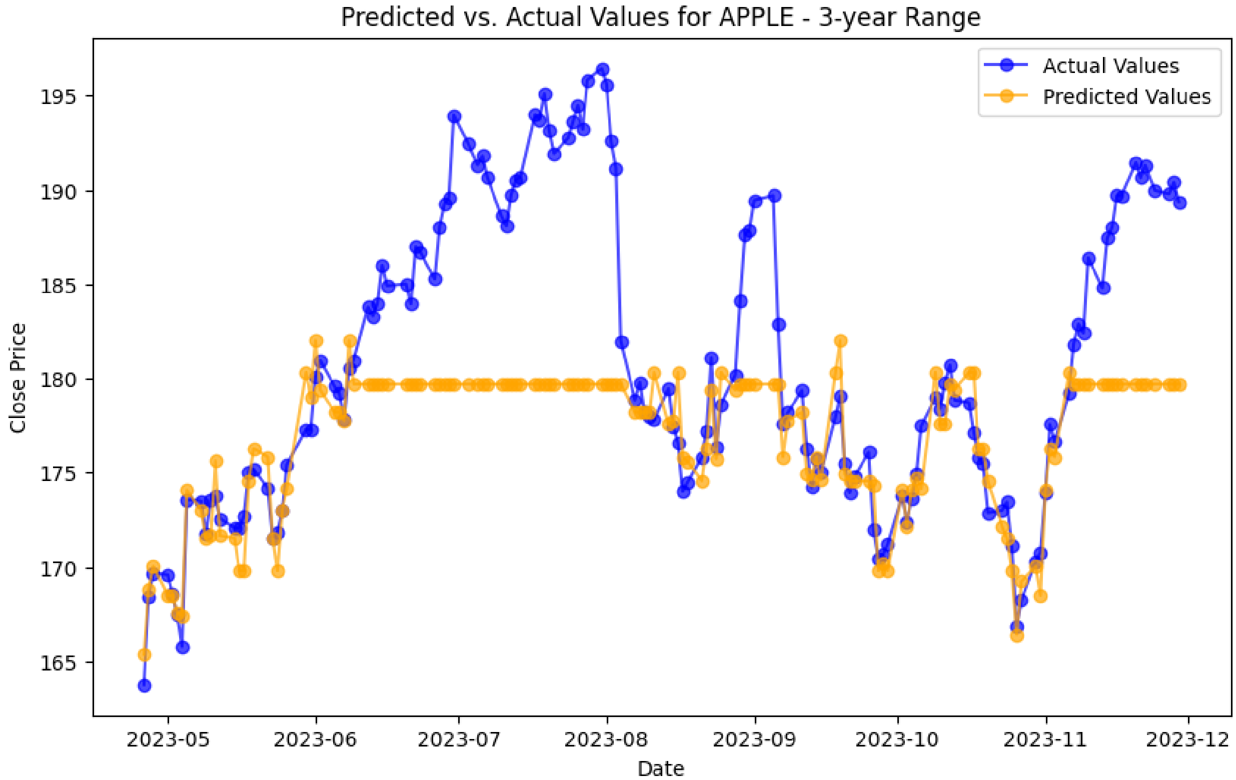
EA - 22 -year Range - Decision Trees Evaluation:



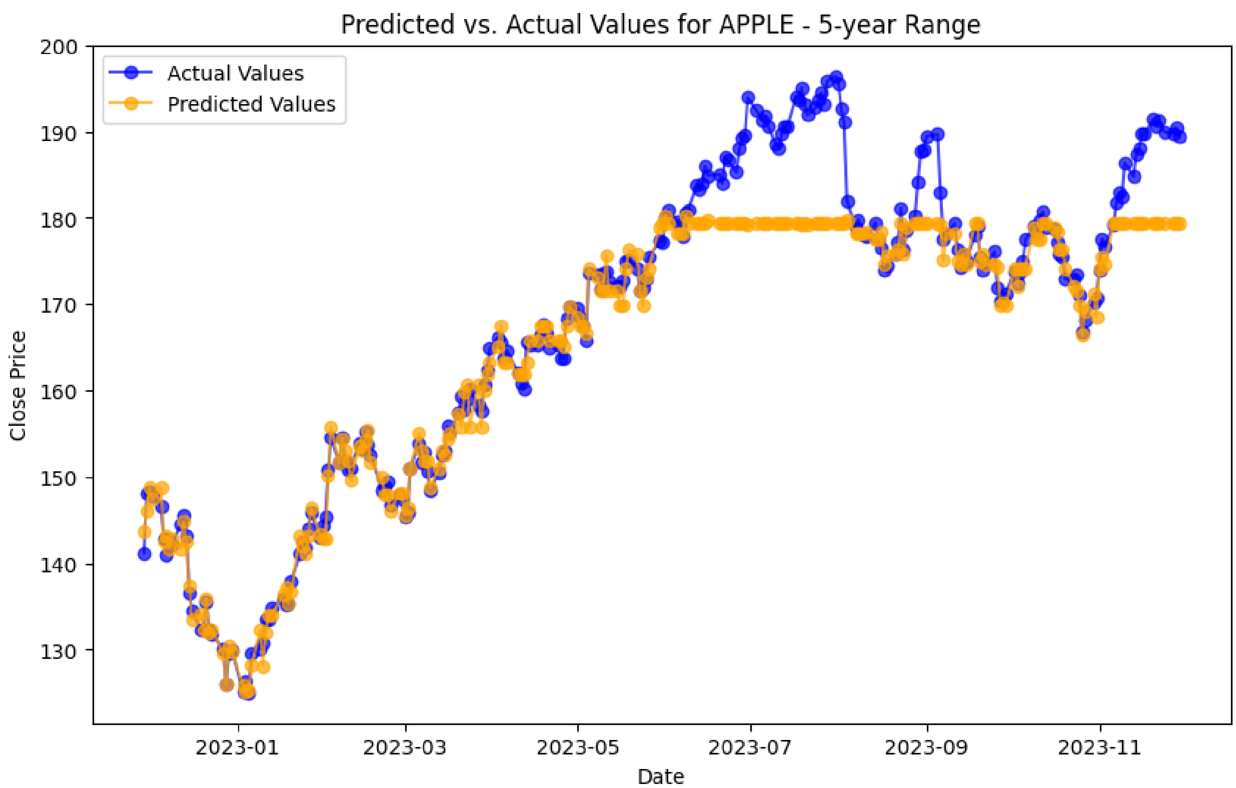
Apple - 1 -year Range - Decision Trees Evaluation:



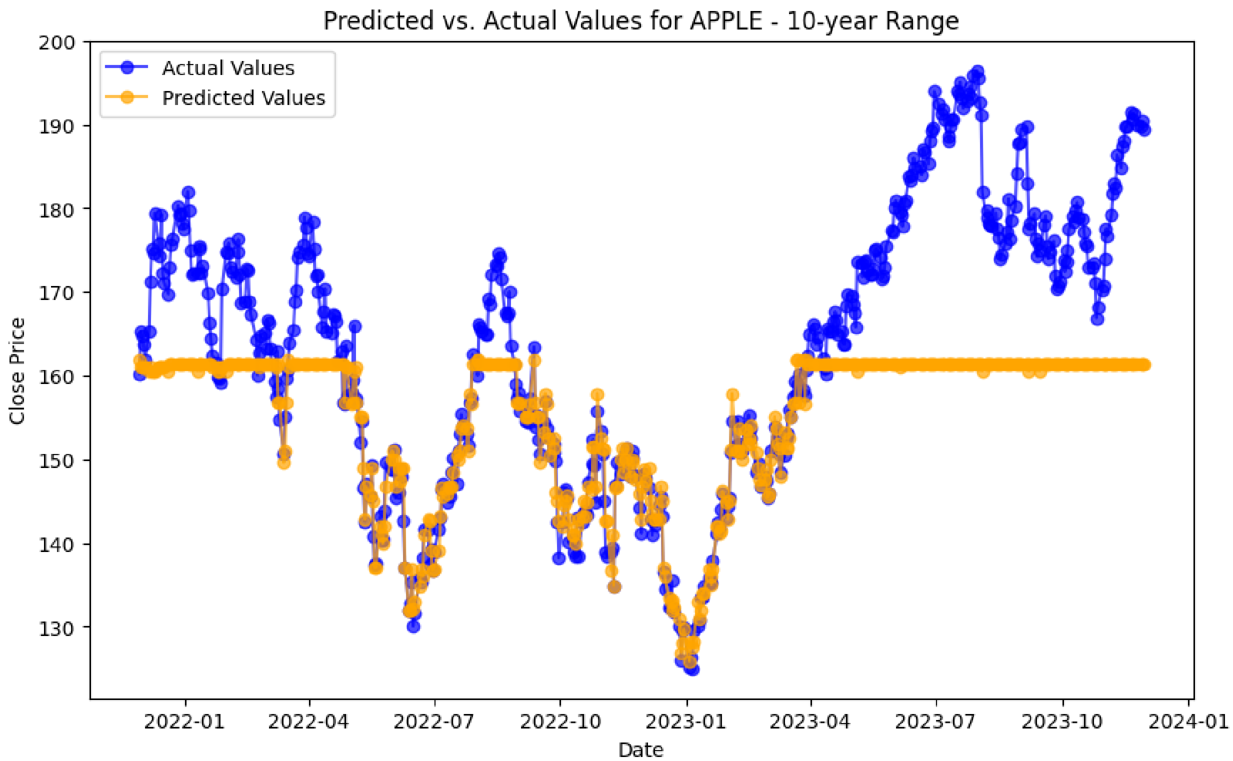
Apple - 3 -year Range - Decision Trees Evaluation:



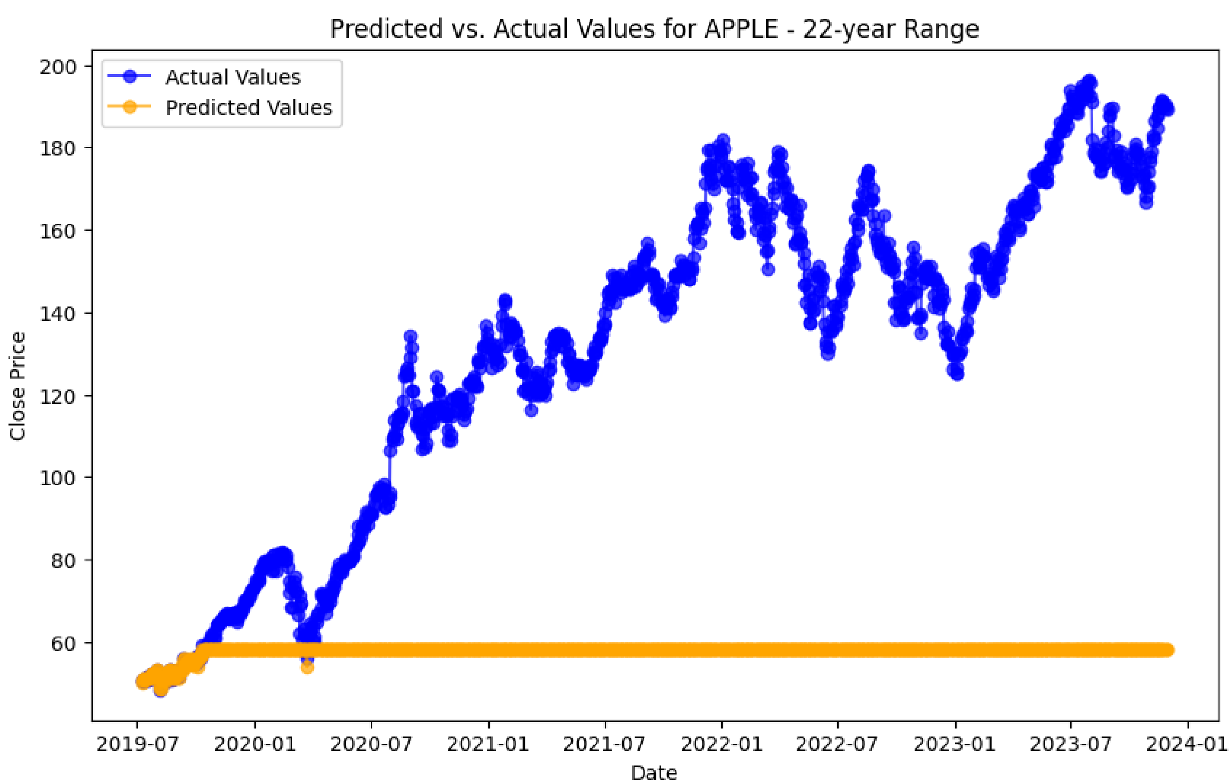
Apple - 5 -year Range - Decision Trees Evaluation:



Apple - 10 -year Range - Decision Trees Evaluation:

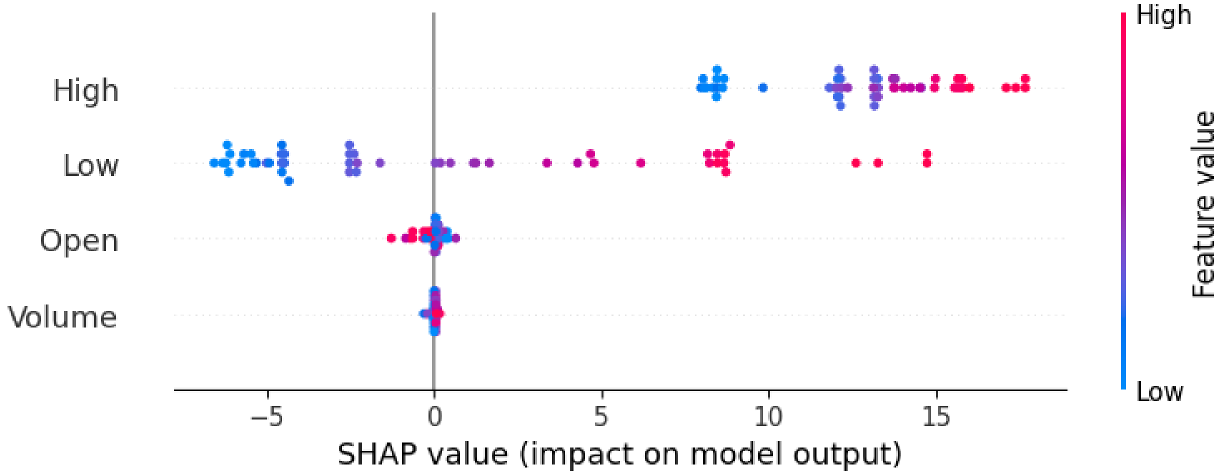


Apple - 22 -year Range - Decision Trees Evaluation:

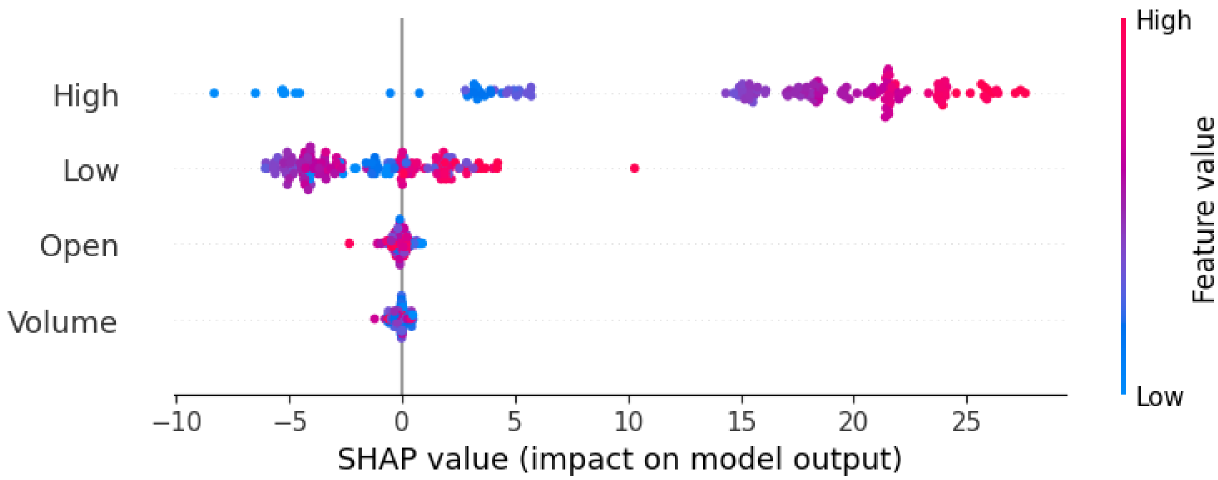


SHAP Impact - Decision Tree

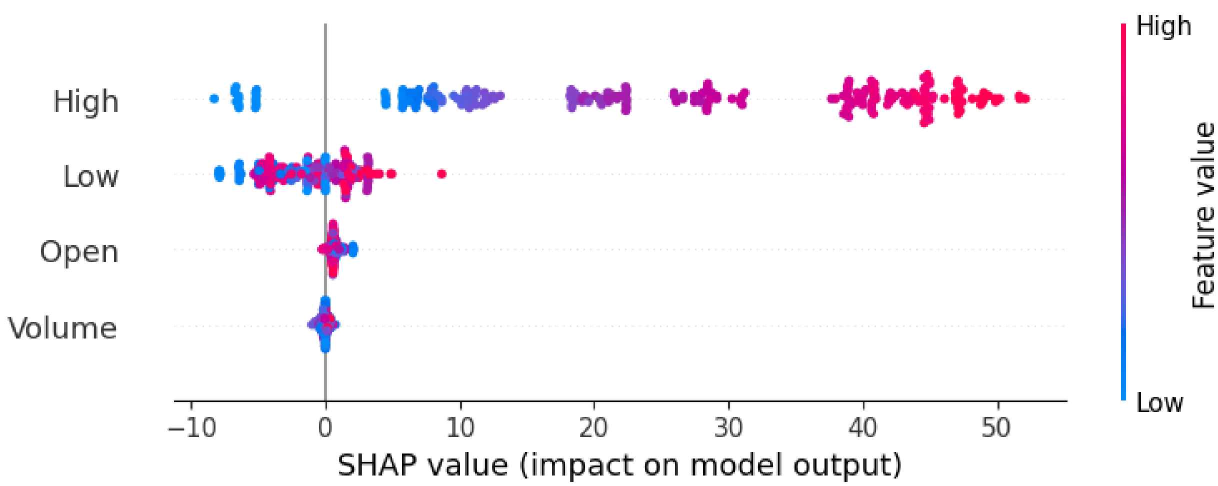
AMD - 1 -year Range - Decision Trees Evaluation:



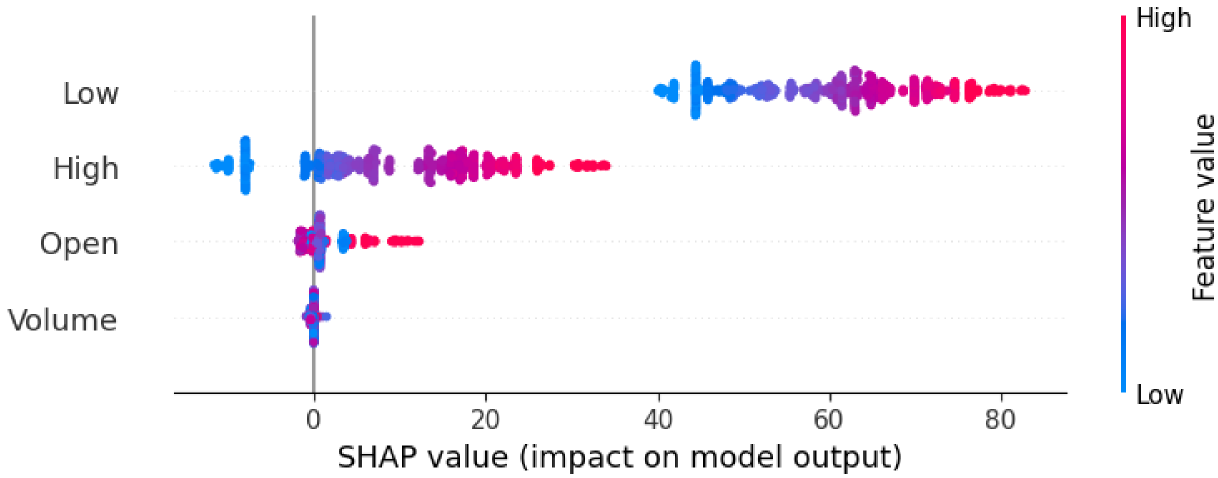
AMD - 3 -year Range - Decision Trees Evaluation:



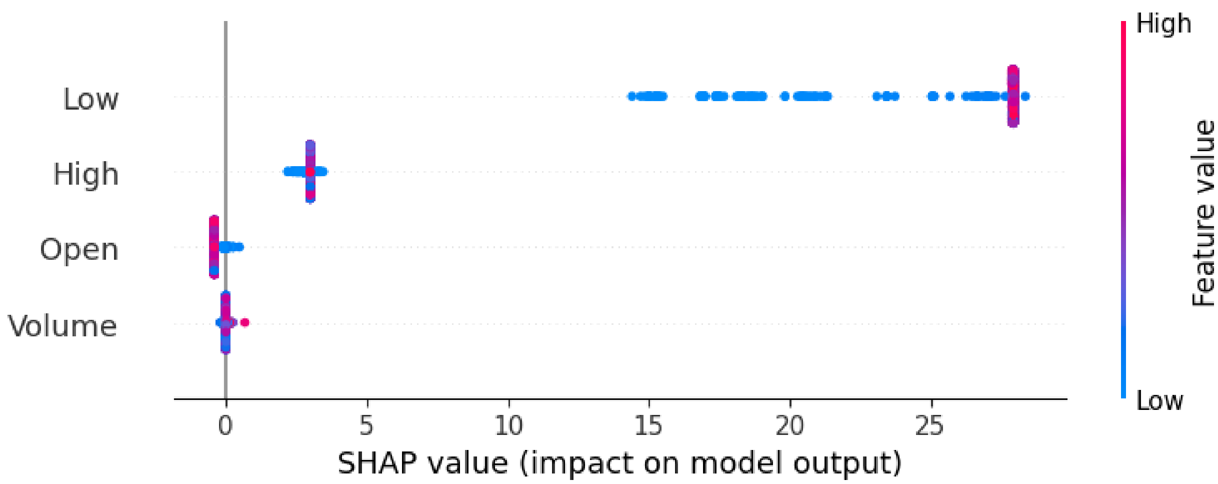
AMD - 5 -year Range - Decision Trees Evaluation:



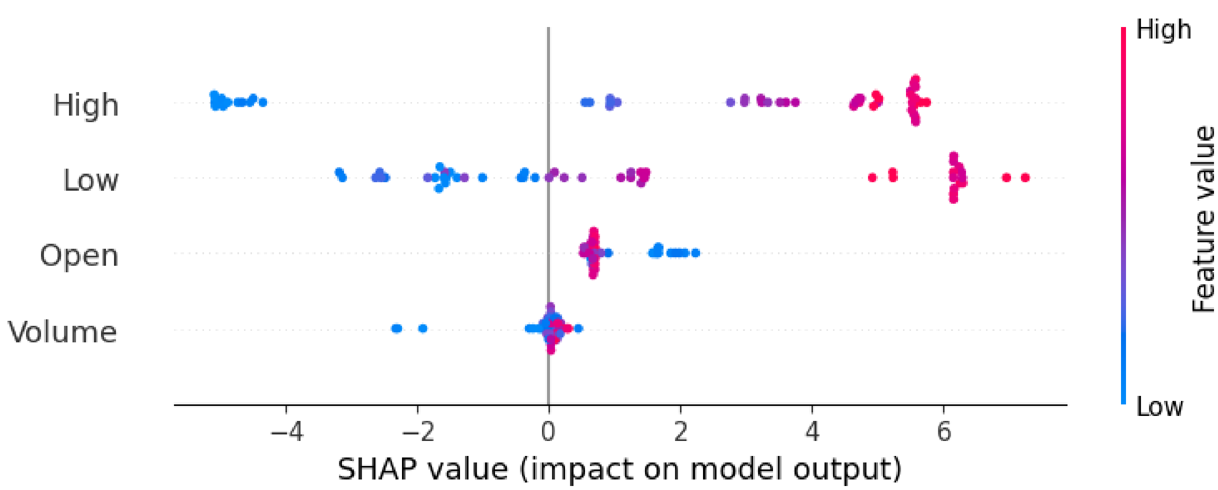
AMD - 10 -year Range - Decision Trees Evaluation:



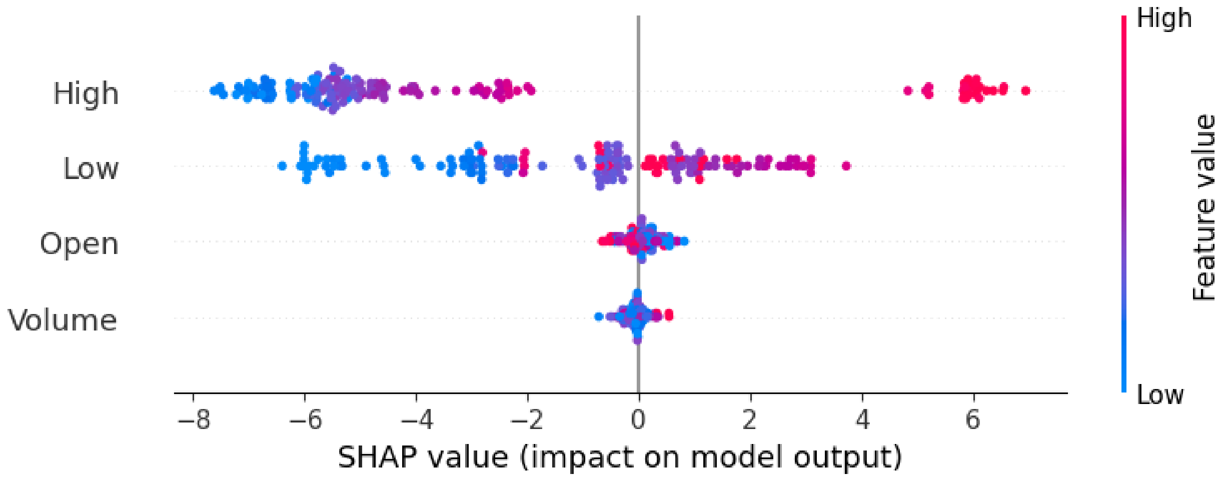
AMD - 22 -year Range - Decision Trees Evaluation:



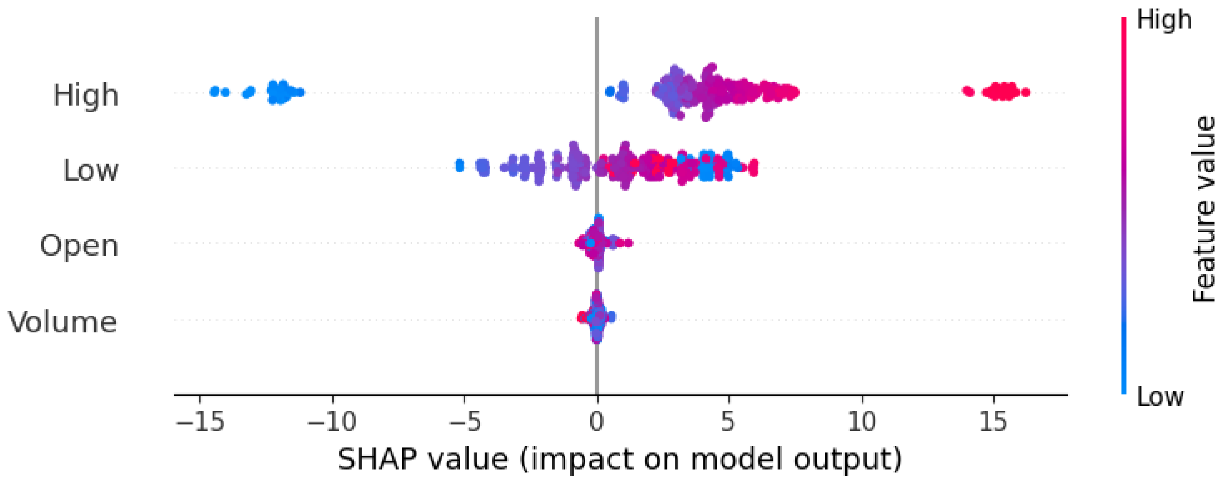
EA - 1 -year Range - Decision Trees Evaluation:



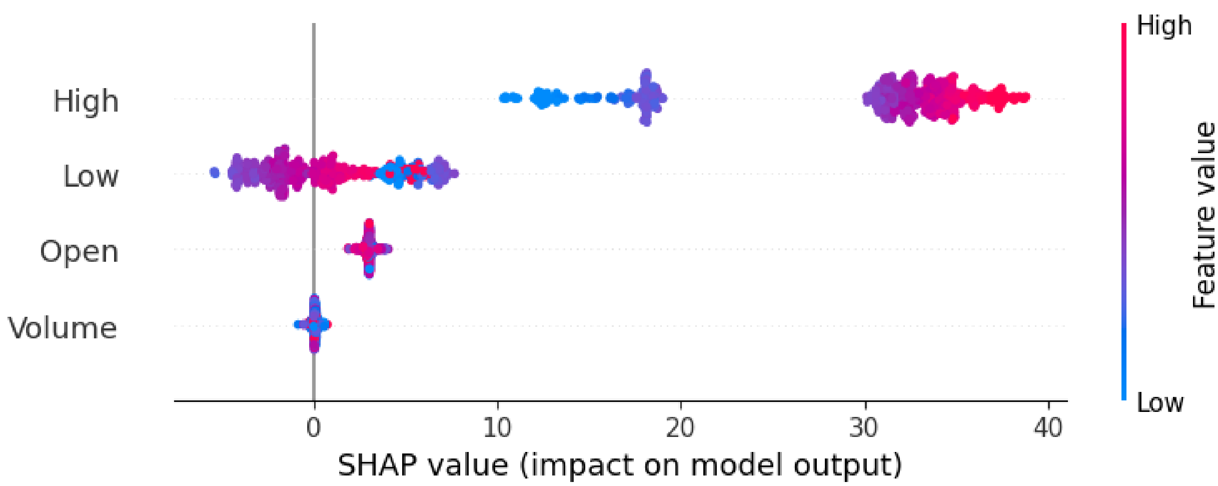
EA - 3 -year Range - Decision Trees Evaluation:



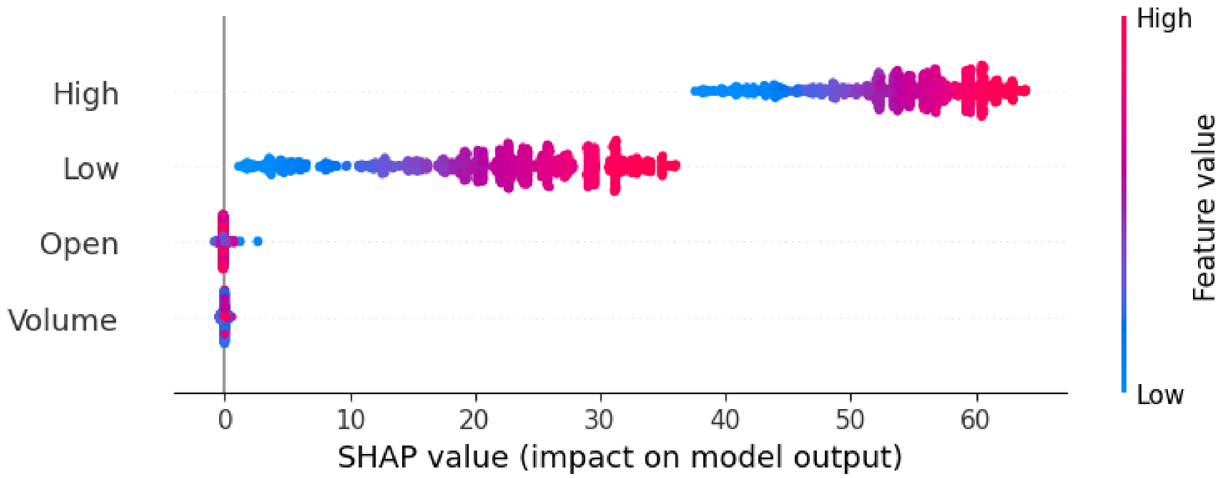
EA - 5 -year Range - Decision Trees Evaluation:



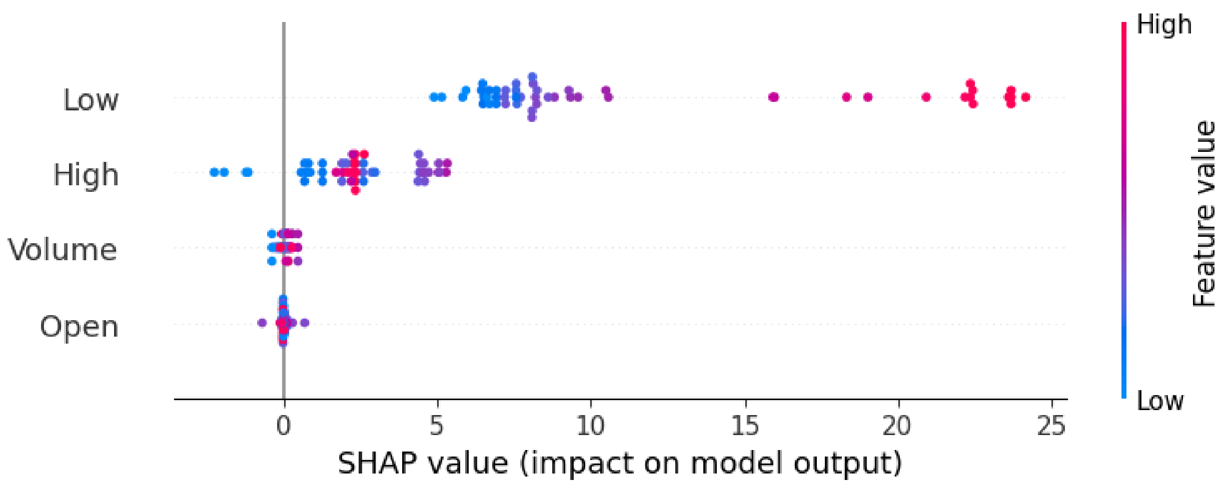
EA - 10 -year Range - Decision Trees Evaluation:



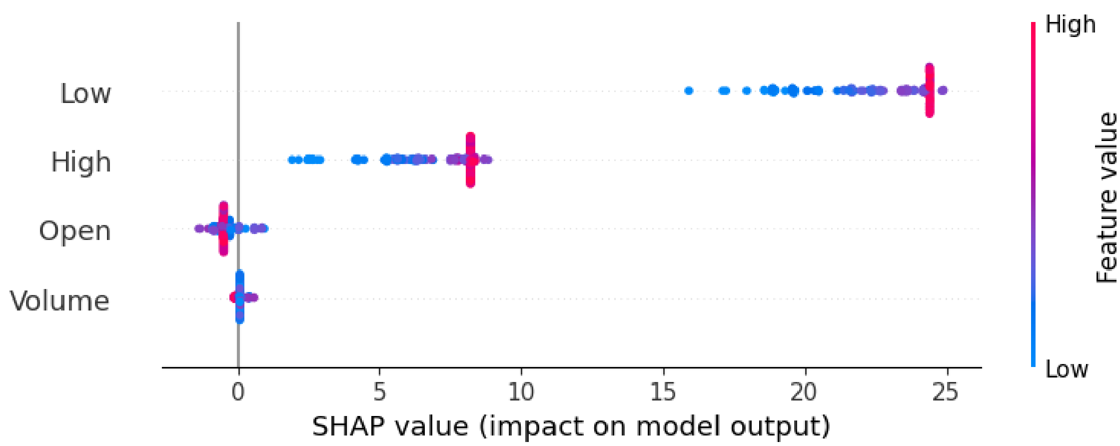
EA - 22 -year Range - Decision Trees Evaluation:



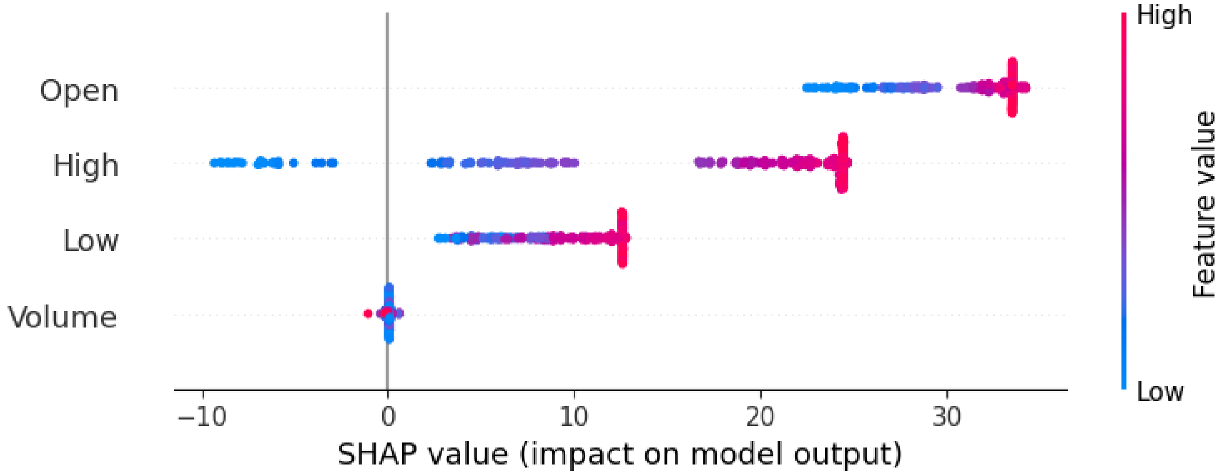
Apple - 1 -year Range - Decision Trees Evaluation:



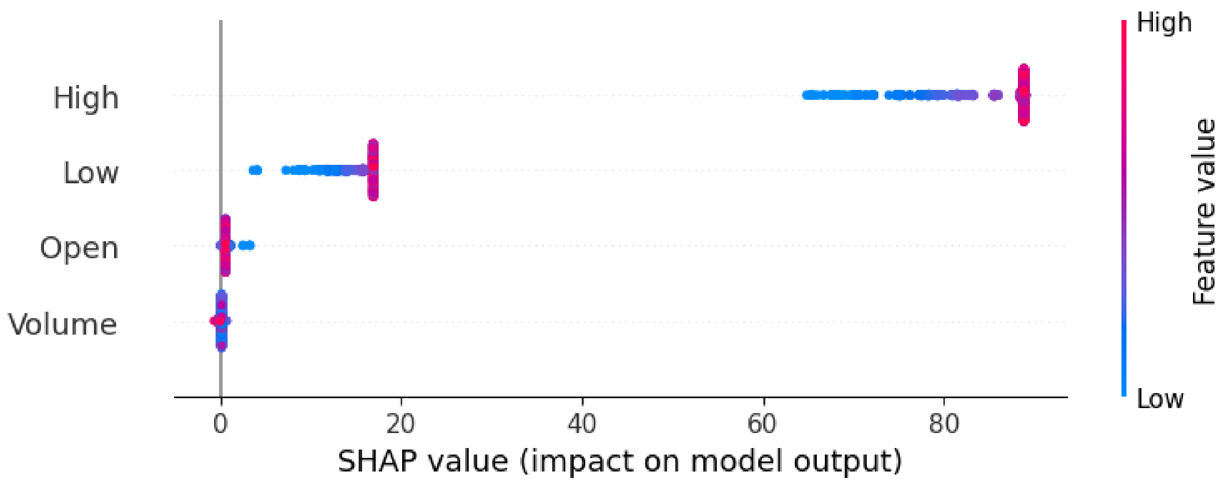
Apple - 3 -year Range - Decision Trees Evaluation:



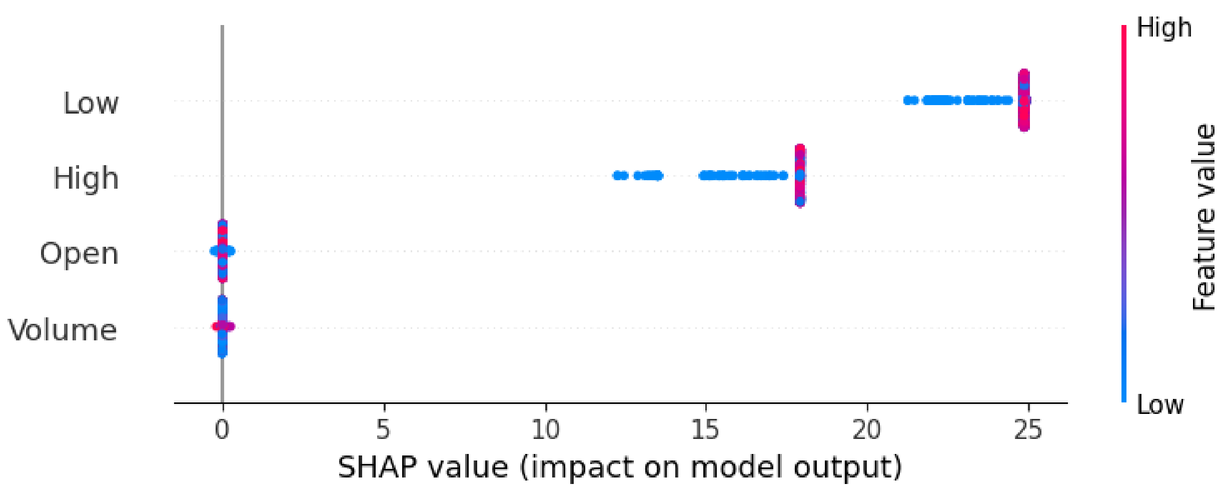
Apple - 5 -year Range - Decision Trees Evaluation:



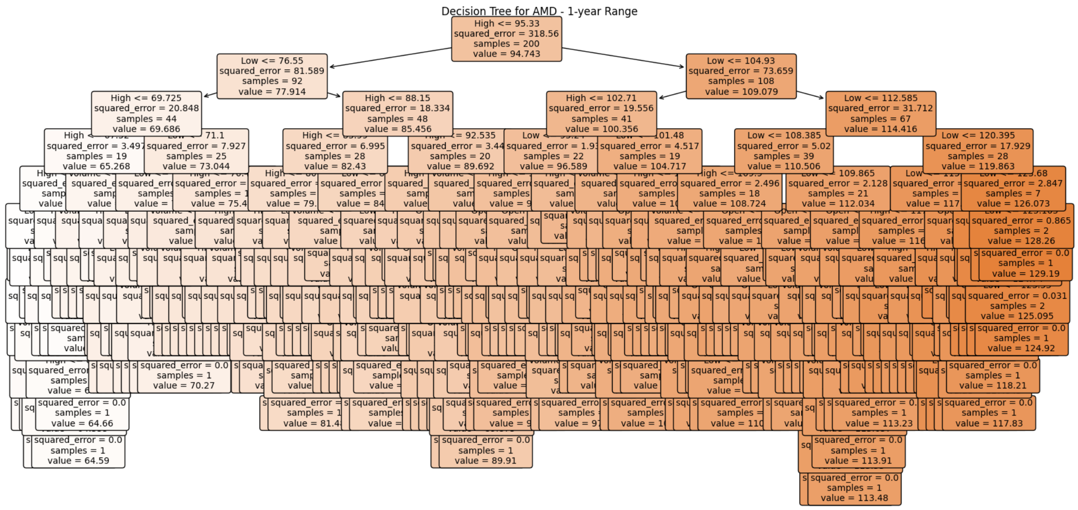
Apple - 10 -year Range - Decision Trees Evaluation:



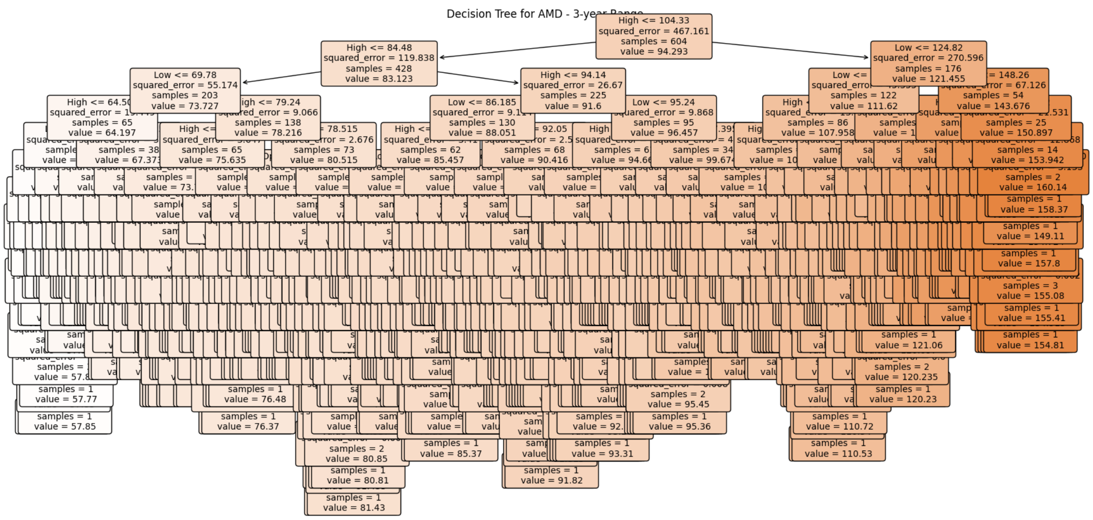
Apple - 22 -year Range - Decision Trees Evaluation:



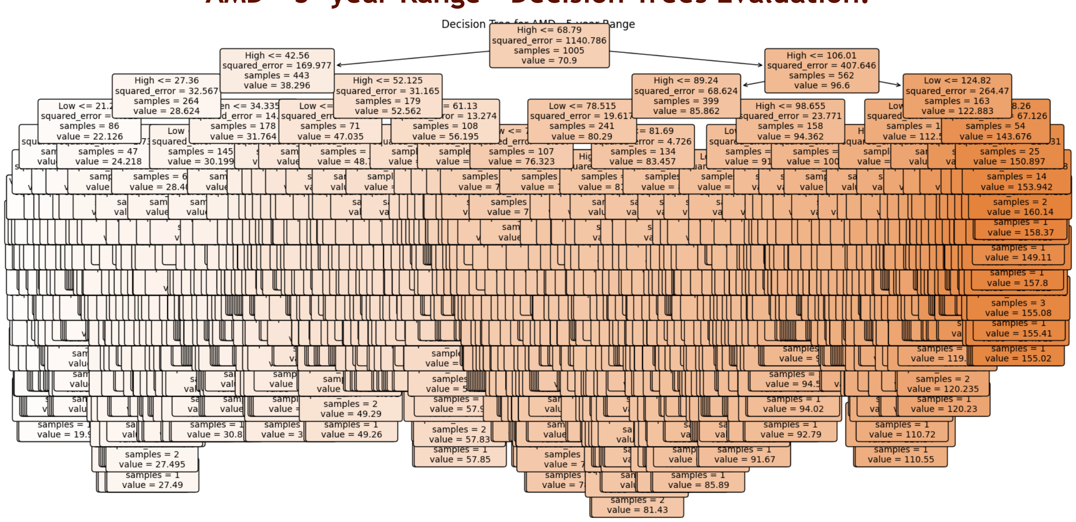
AMD - 1 -year Range - Decision Trees Evaluation:



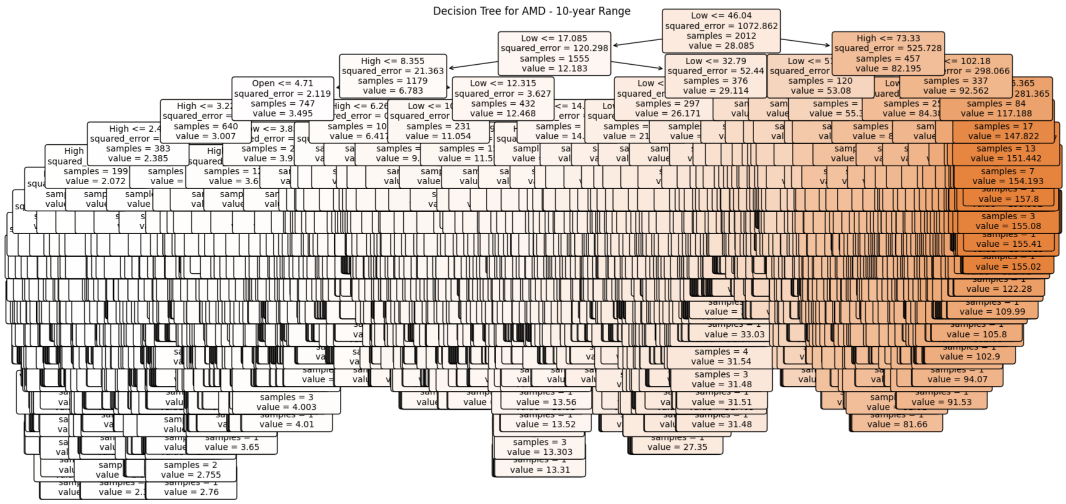
AMD - 3 -year Range - Decision Trees Evaluation:



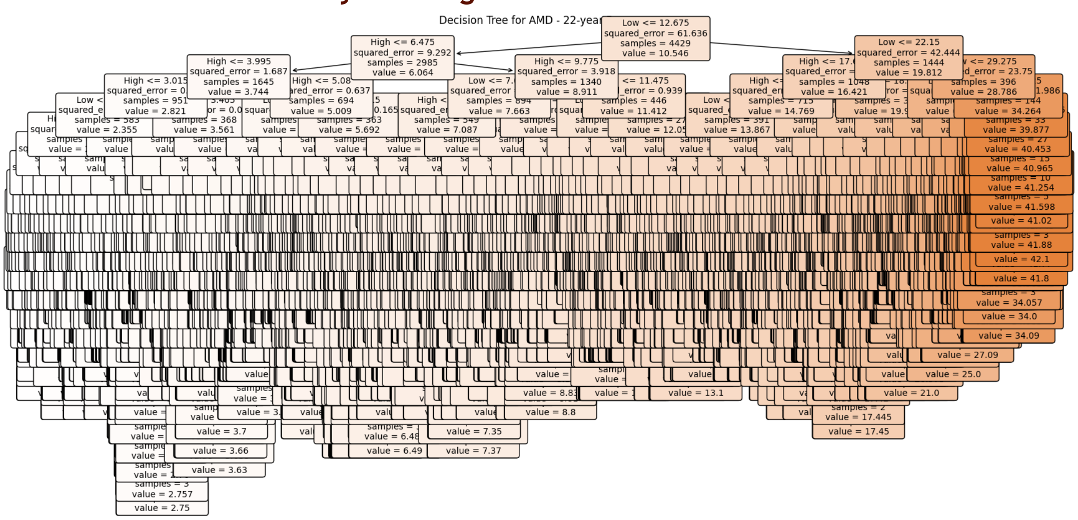
AMD - 5 -year Range - Decision Trees Evaluation:



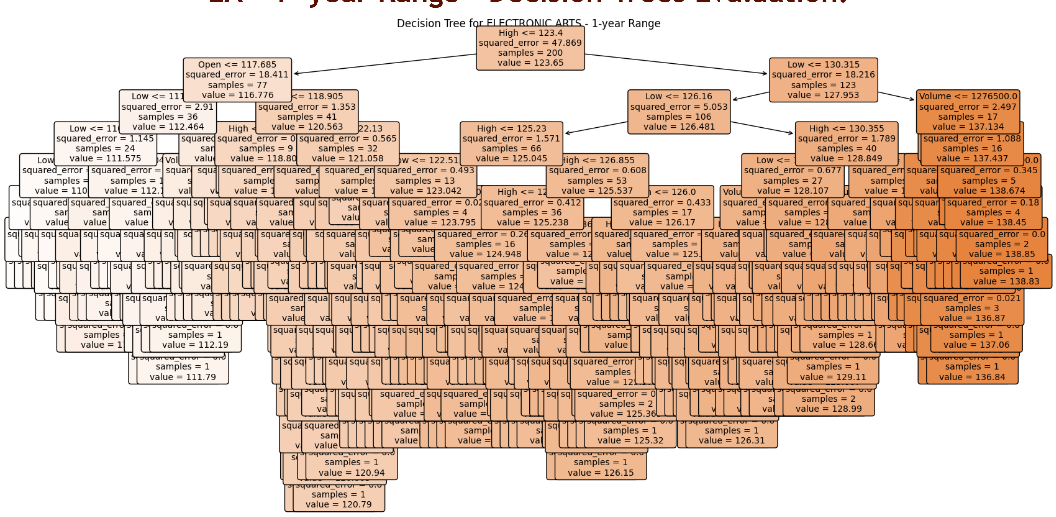
AMD - 10 -year Range - Decision Trees Evaluation:



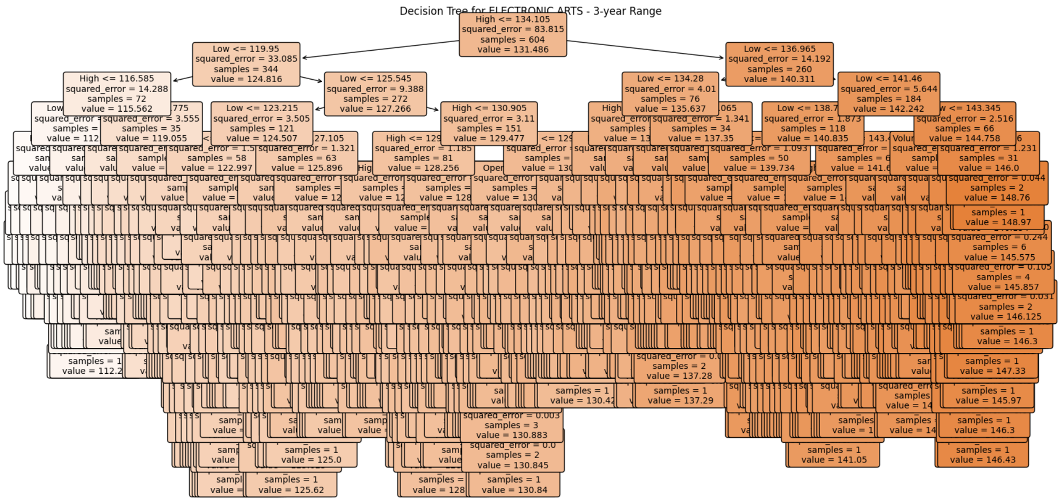
AMD - 22 -year Range - Decision Trees Evaluation:



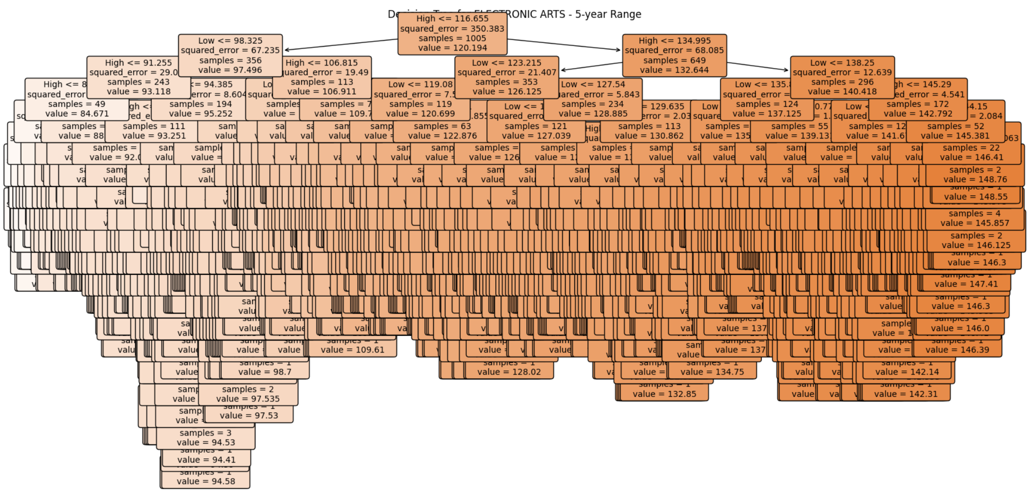
EA - 1 -year Range - Decision Trees Evaluation:



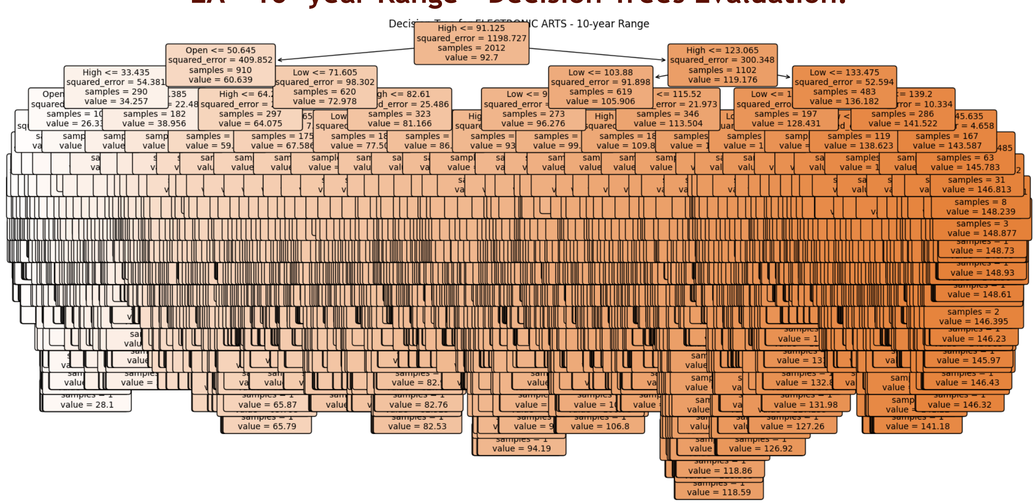
EA - 3 -year Range - Decision Trees Evaluation:



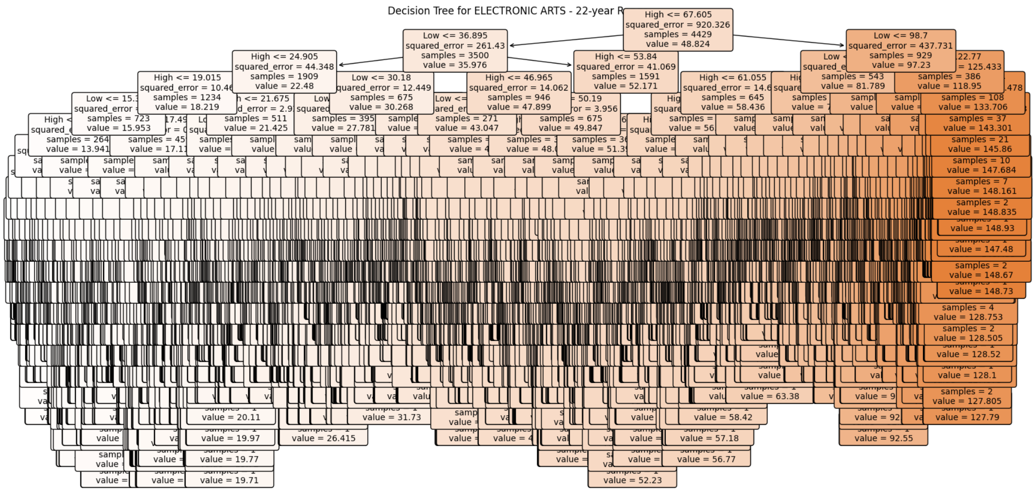
EA - 5 -year Range - Decision Trees Evaluation:



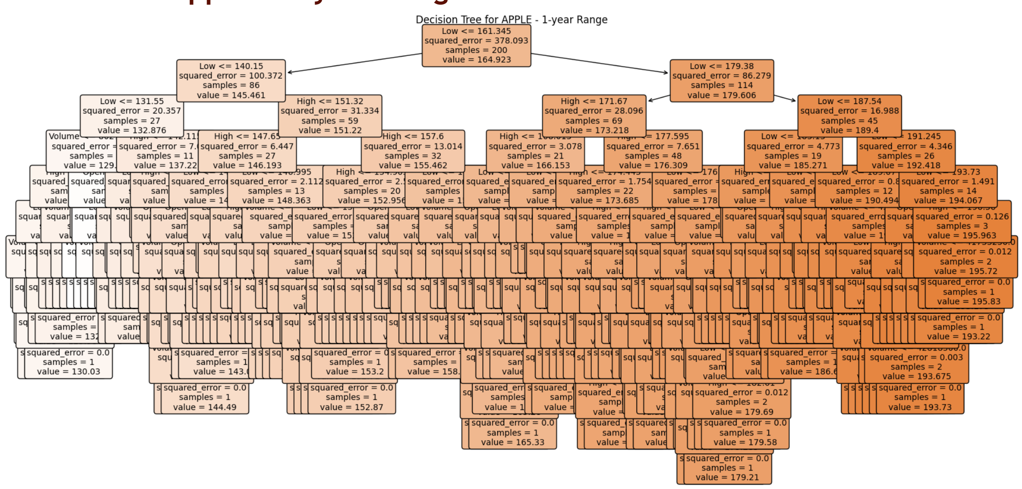
EA - 10 -year Range - Decision Trees Evaluation:



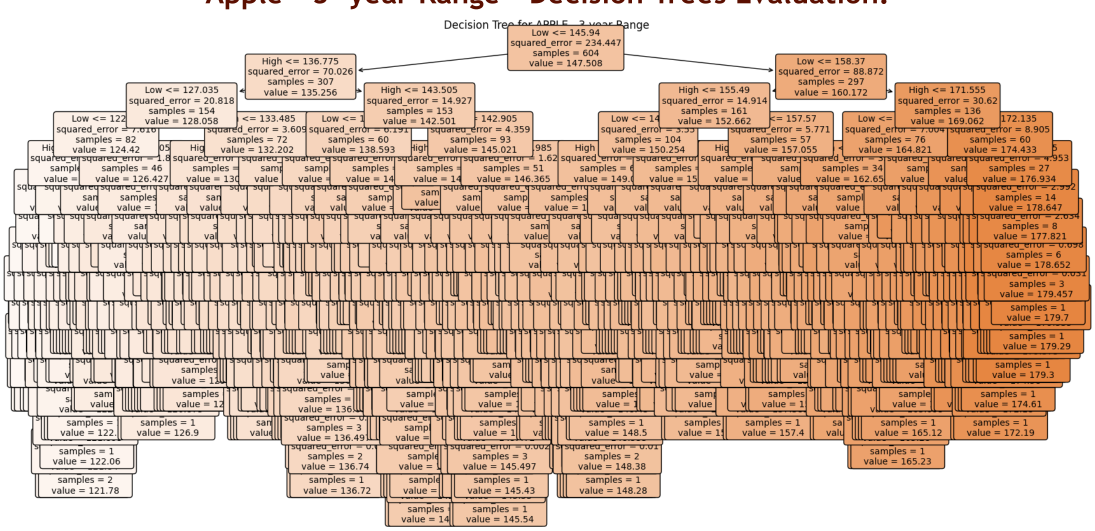
EA - 22 -year Range - Decision Trees Evaluation:



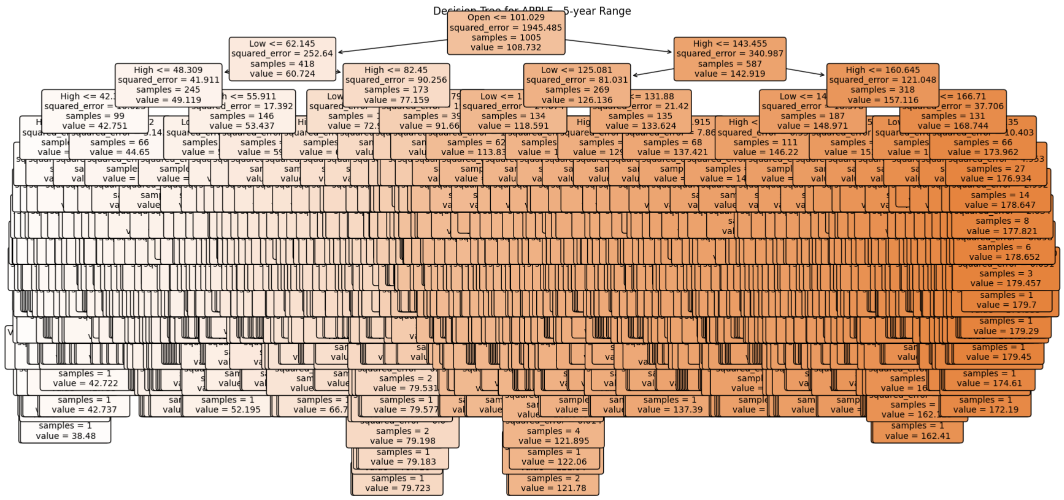
Apple - 1 -year Range - Decision Trees Evaluation:



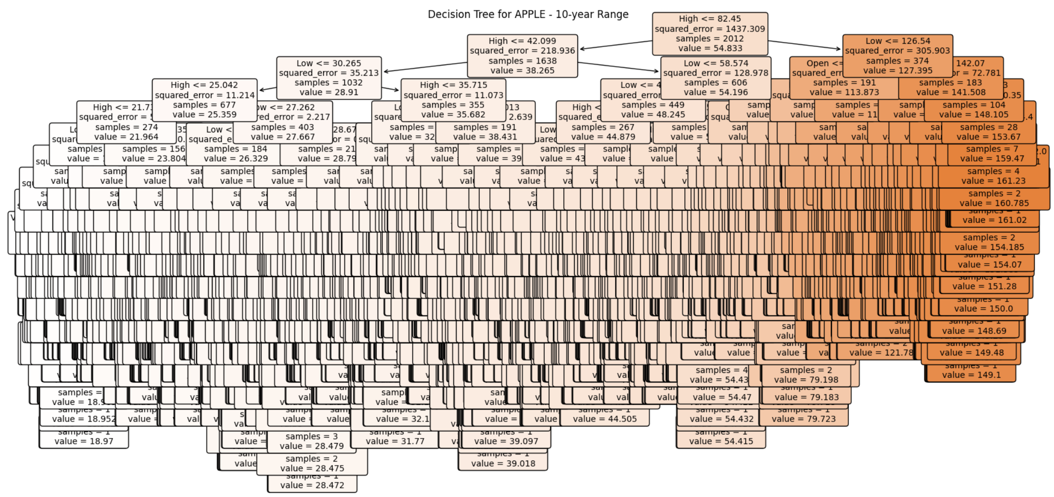
Apple - 3 -year Range - Decision Trees Evaluation:



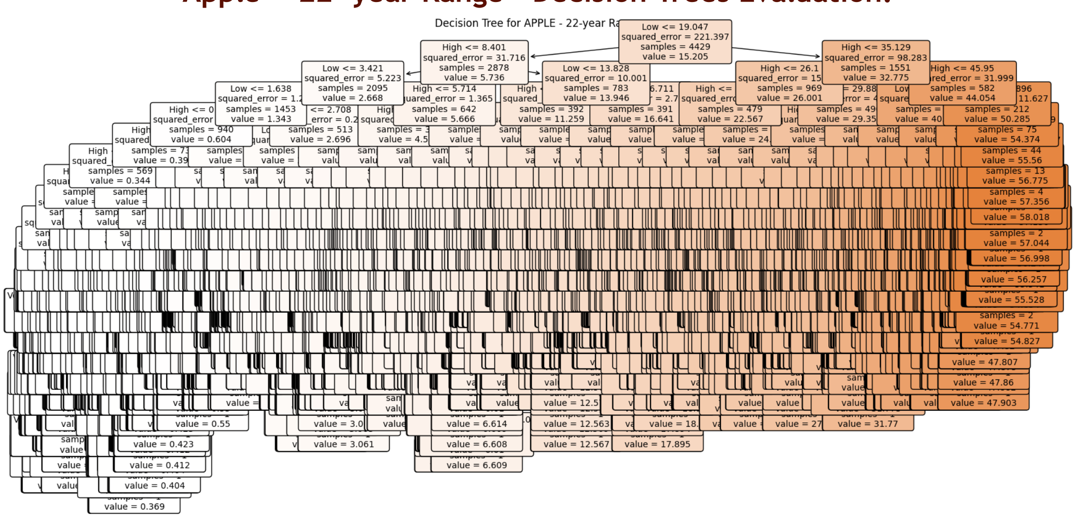
Apple - 5 -year Range - Decision Trees Evaluation:

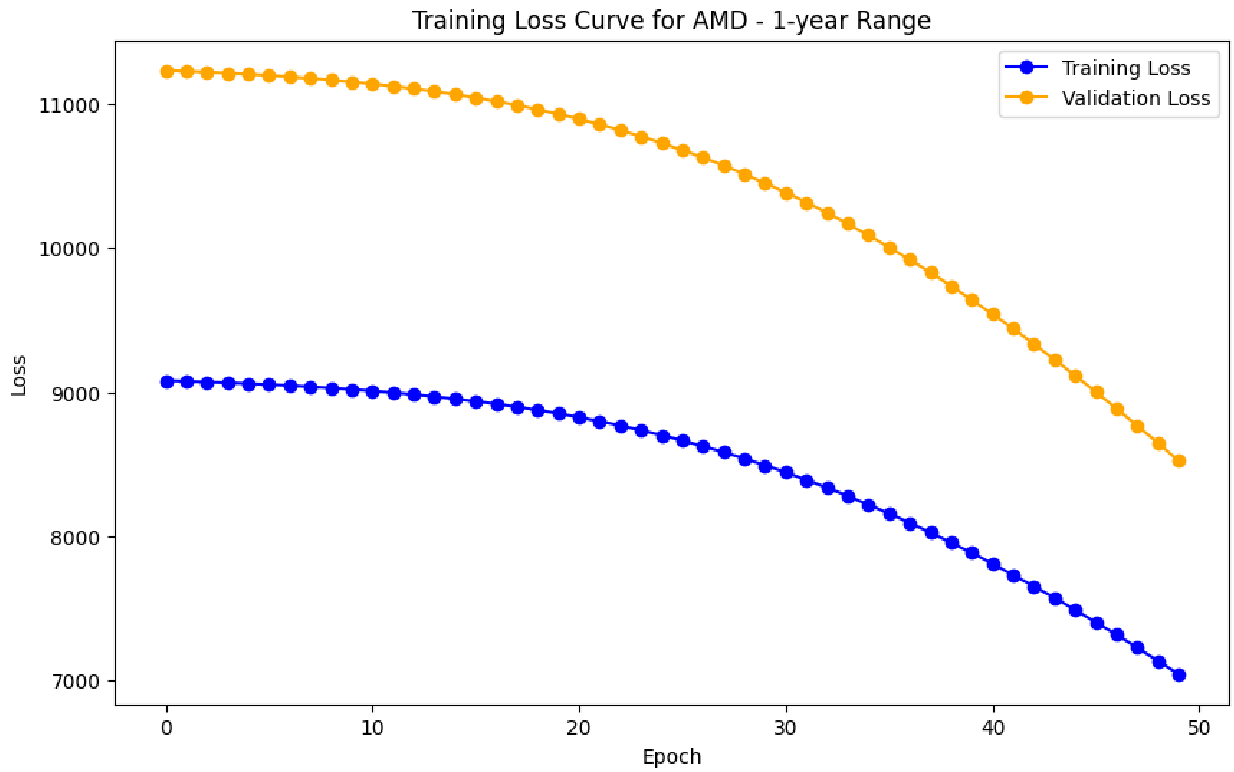


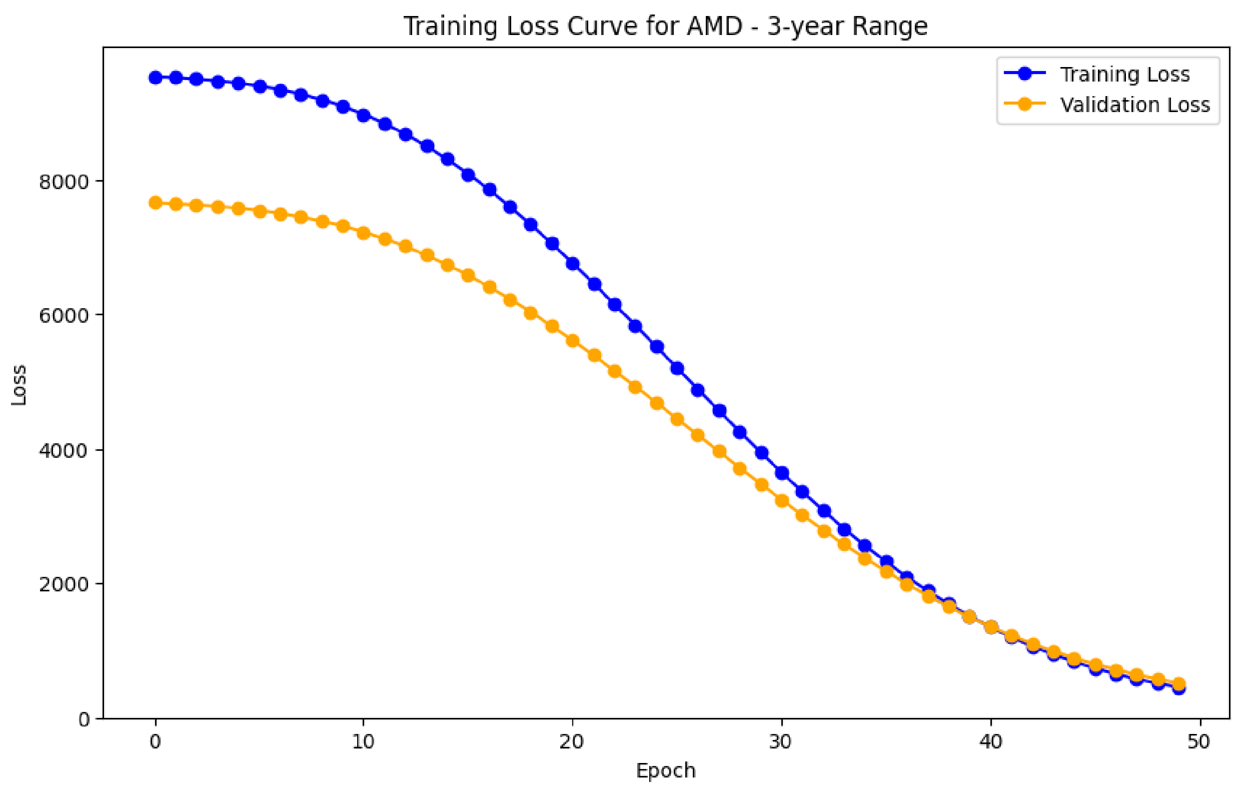
Apple - 10 -year Range - Decision Trees Evaluation:

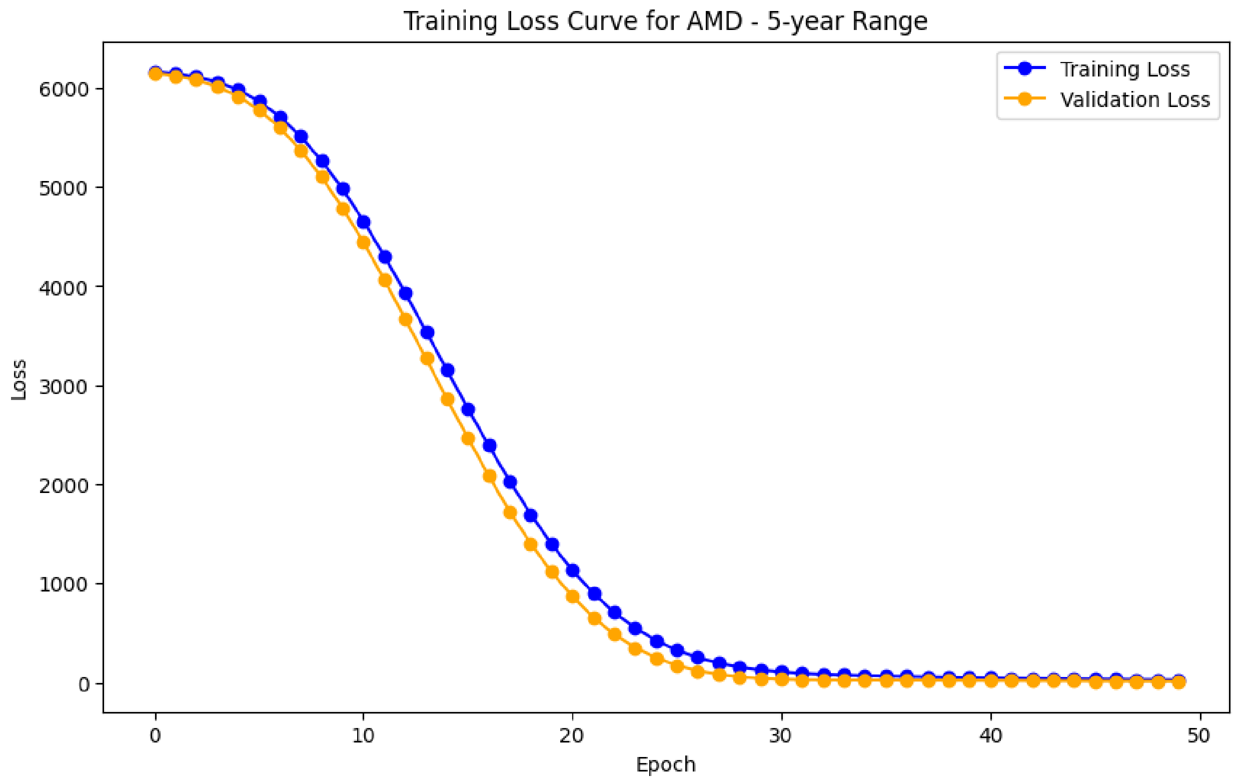


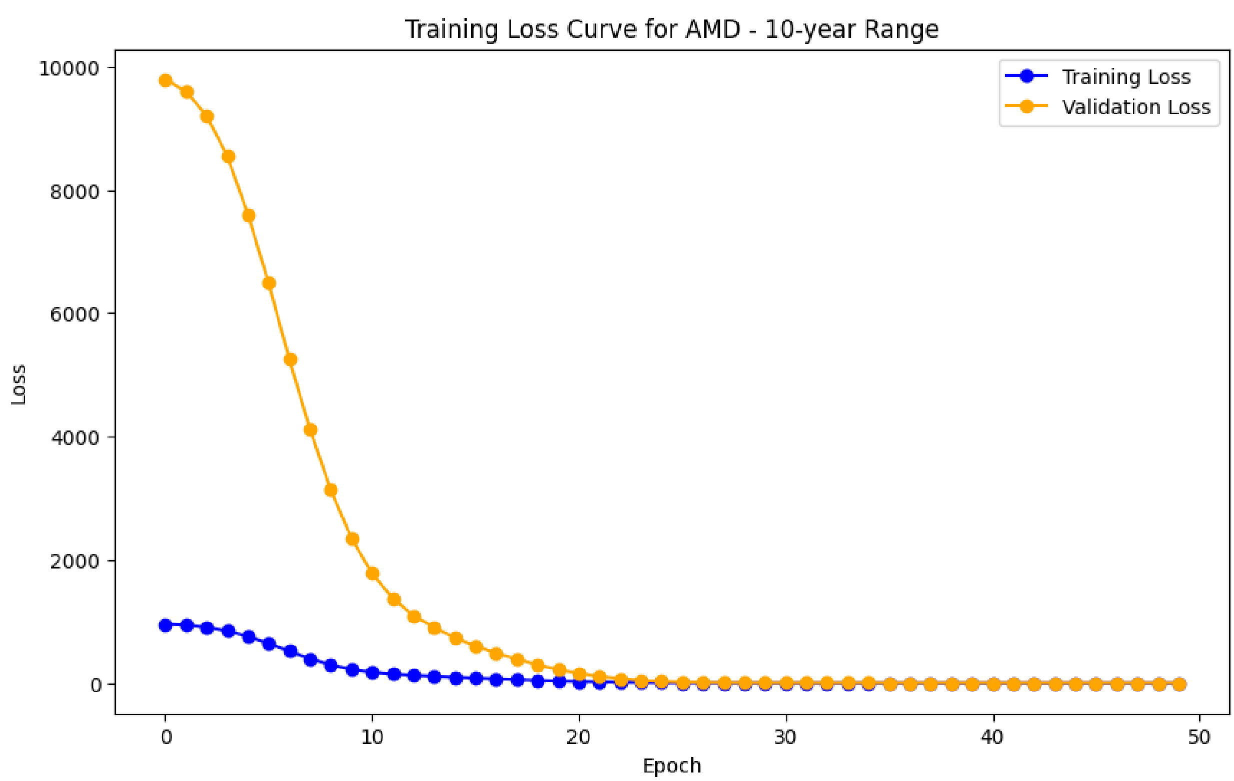
Apple - 22 -year Range - Decision Trees Evaluation:

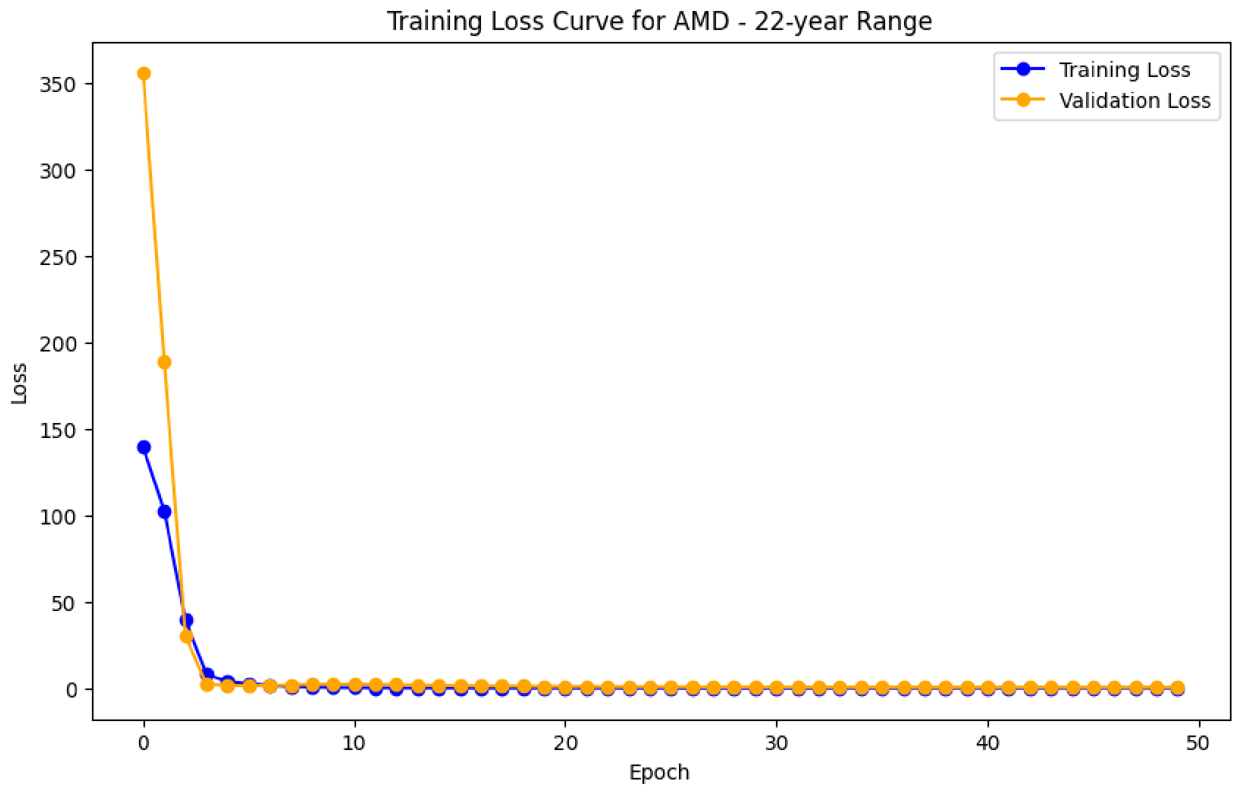


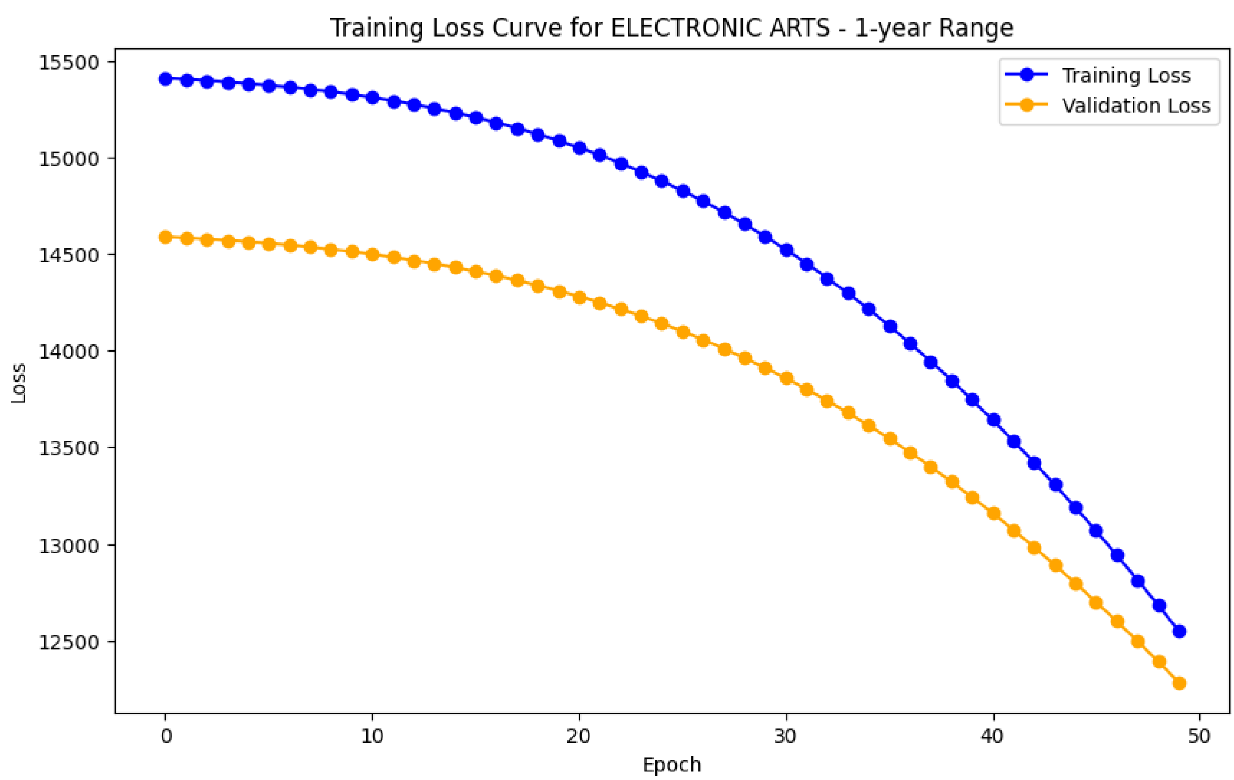

Training Loss Curve for AMD - 1-year Range
Training Loss
Validation Loss
Loss
Epoch
11000
10000
9000
8000
7000
0
10
20
30
40
50

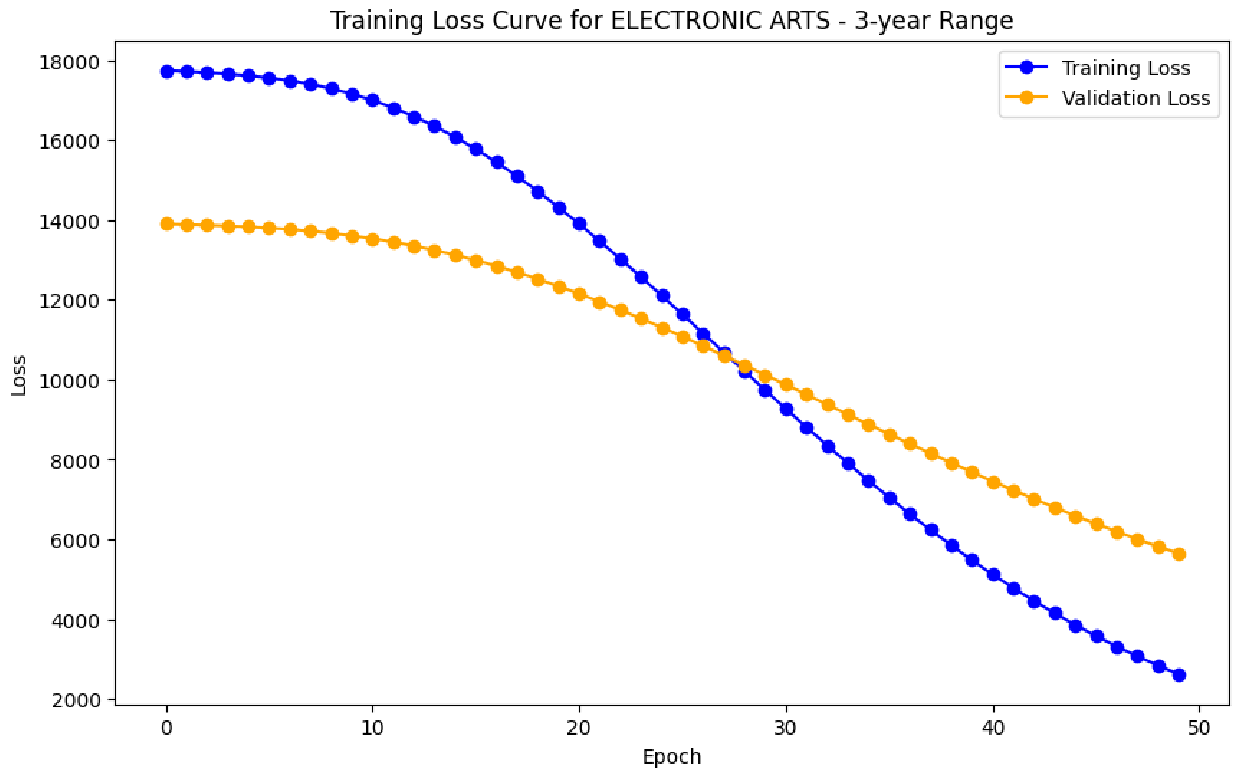

Training Loss Curve for AMD - 3-year Range
Training Loss
Validation Loss
Loss
Epoch
8000
6000
4000
2000
0
0
10
20
30
40
50

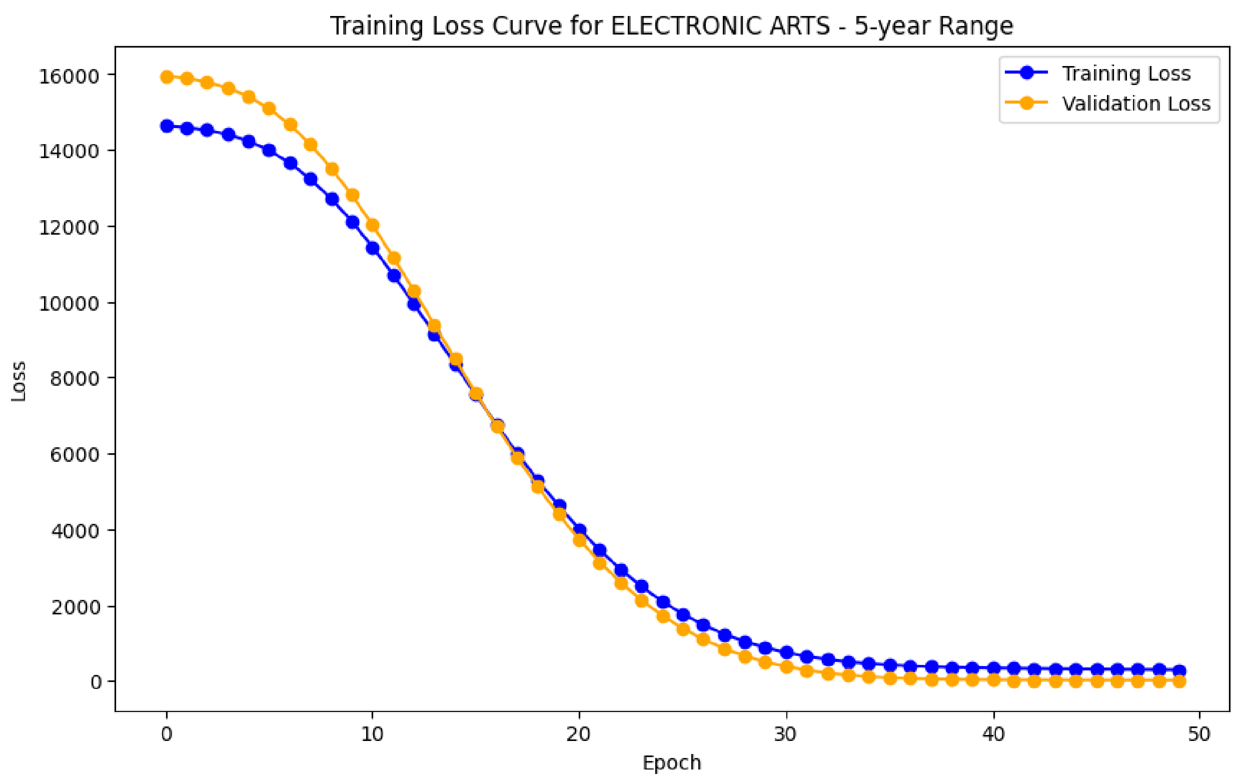

Training Loss Curve for AMD - 5-year Range
Training Loss
Validation Loss
Loss
Epoch
0
1000
2000
3000
4000
5000
6000
0
10
20
30
40
50

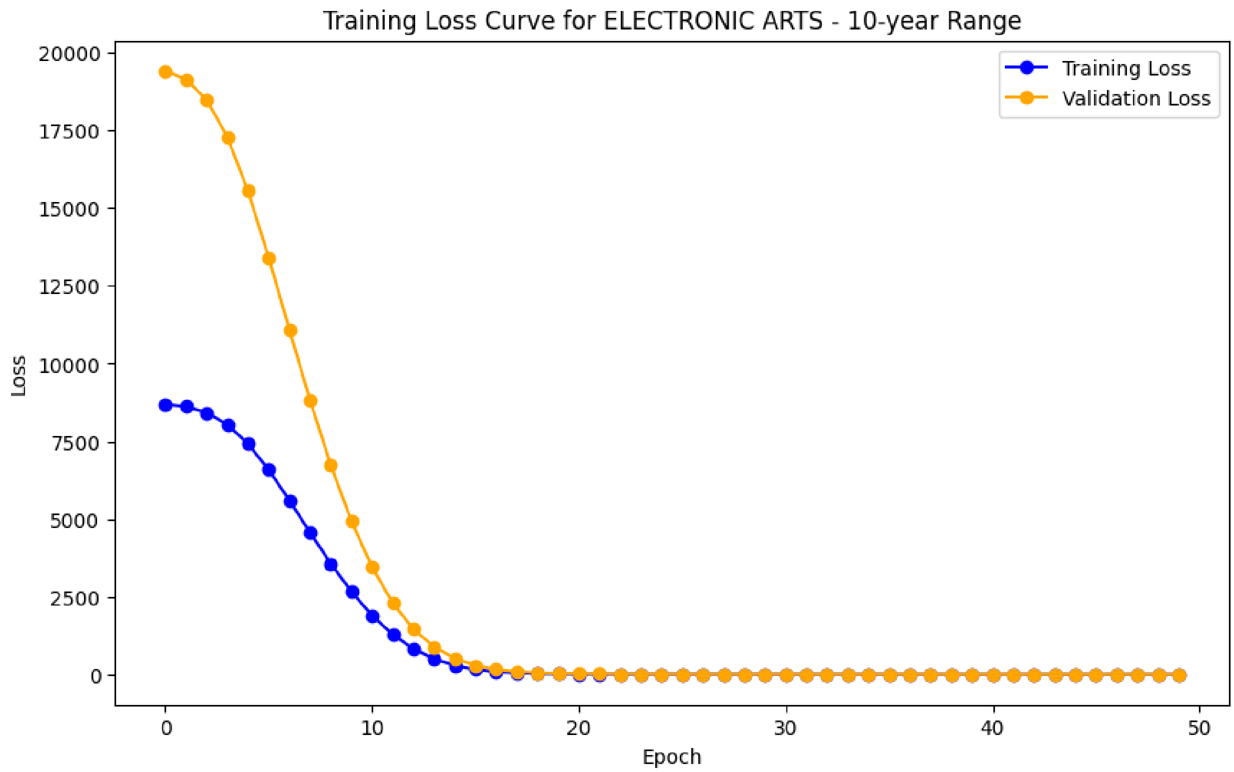

Training Loss Curve for AMD - 10-year Range
Training Loss
Validation Loss
Loss
Epoch
0
2000
4000
6000
8000
10000
0
10
20
30
40
50

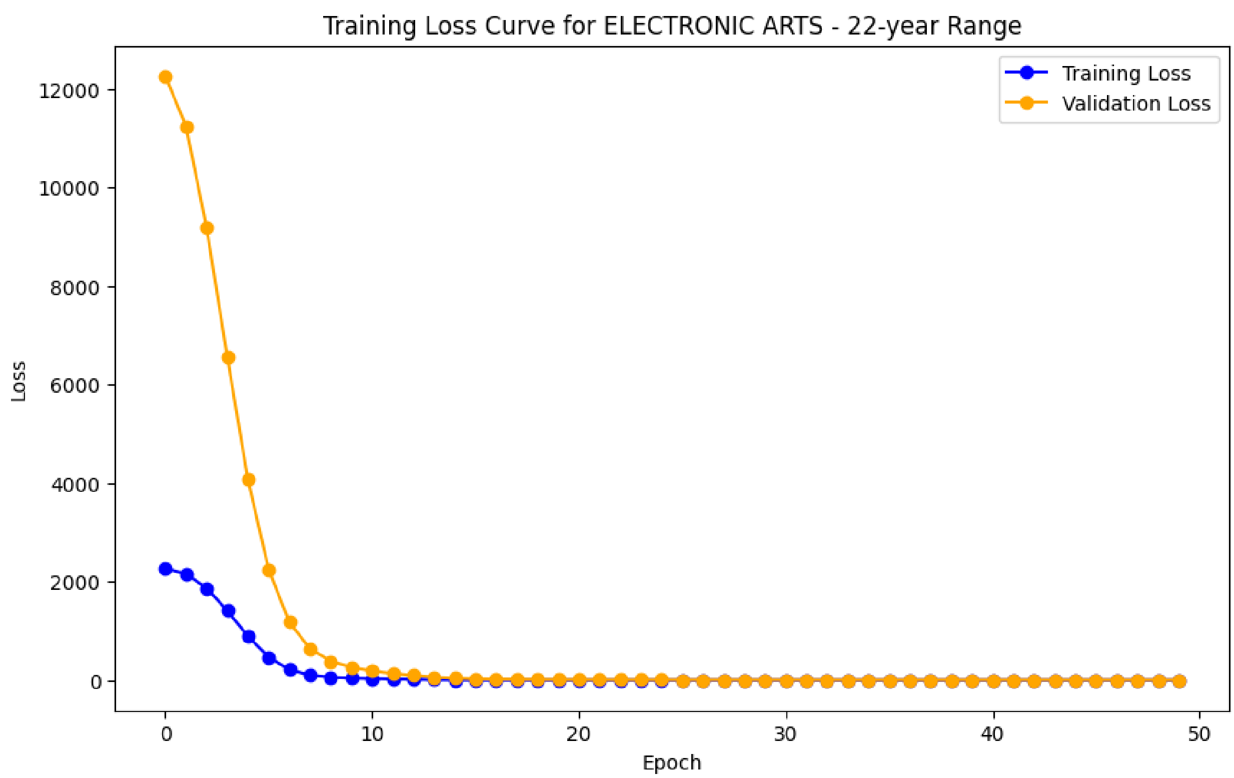

Training Loss Curve for AMD - 22-year Range
Training Loss
Validation Loss
Loss
Epoch
0
50
100
150
200
250
300
350
0
10
20
30
40
50

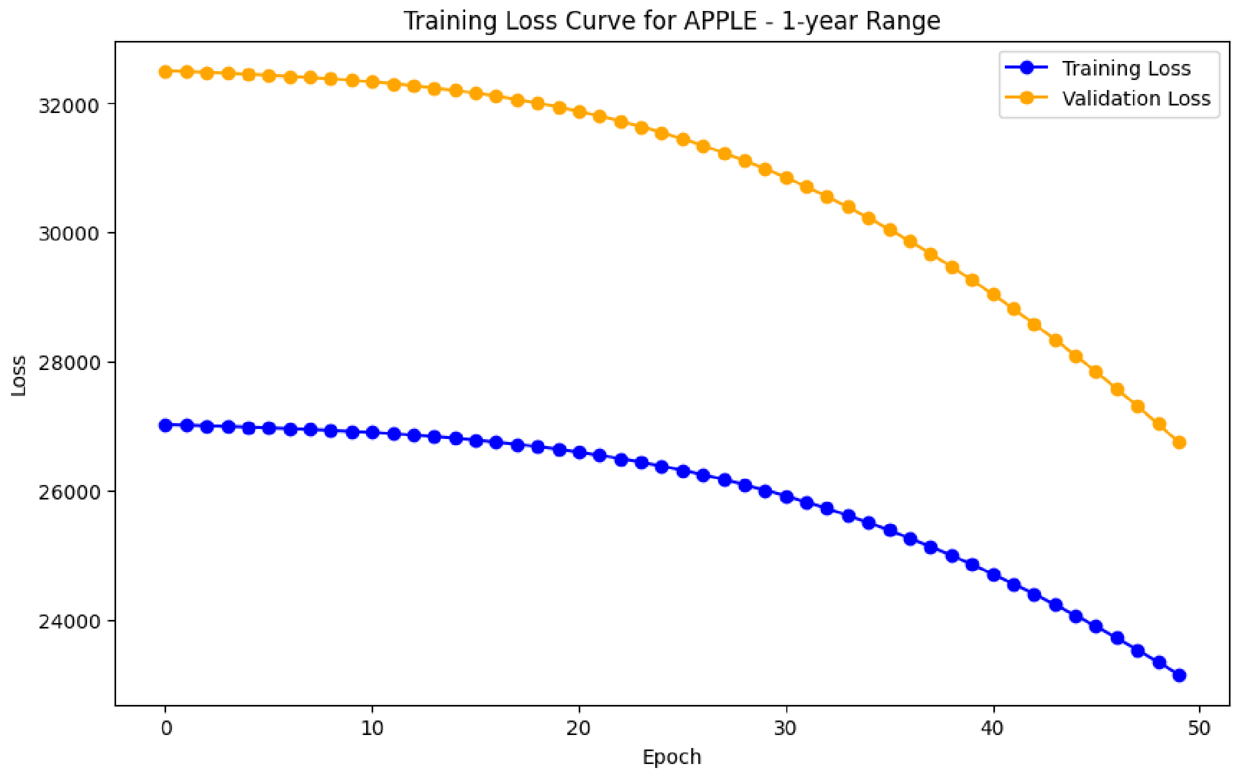

Training Loss Curve for ELECTRONIC ARTS - 1-year Range
Training Loss
Validation Loss
Loss
Epoch
12500
13000
13500
14000
14500
15000
15500
0
10
20
30
40
50

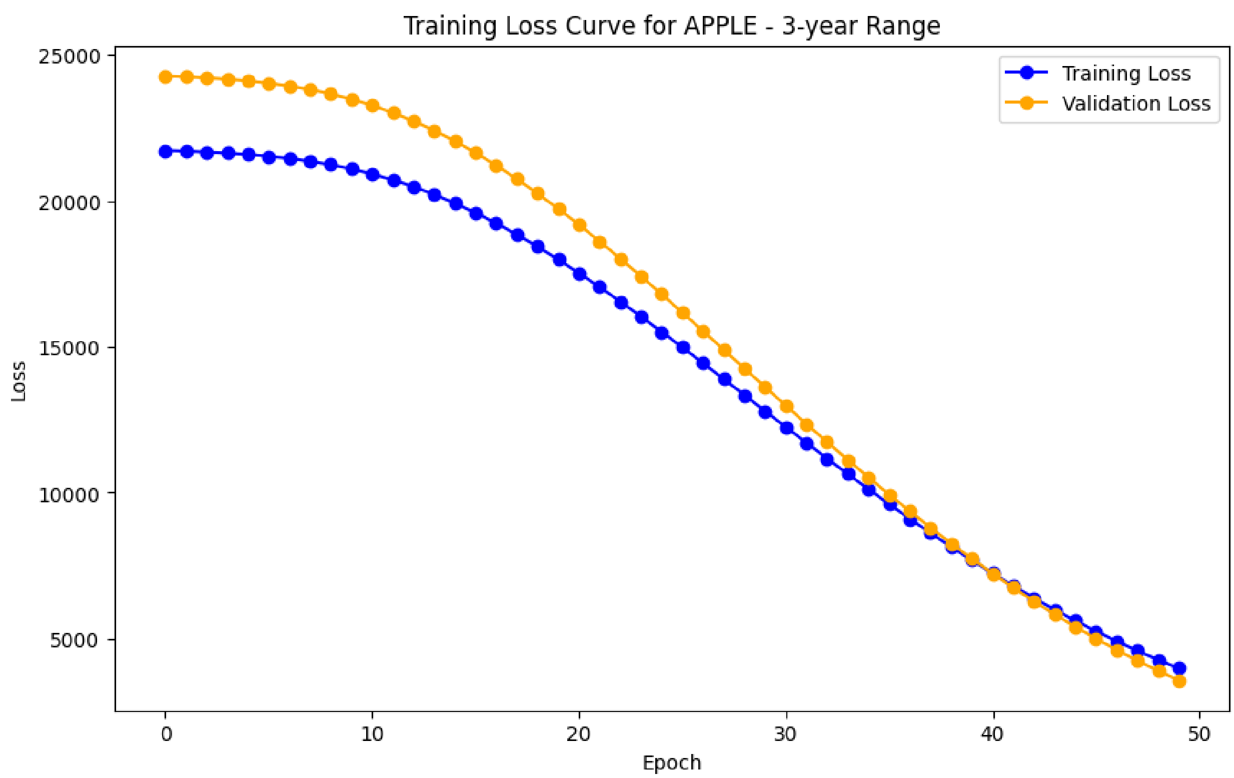

Training Loss Curve for ELECTRONIC ARTS - 3-year Range
Training Loss
Validation Loss
Loss
Epoch
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
10
20
30
40
50

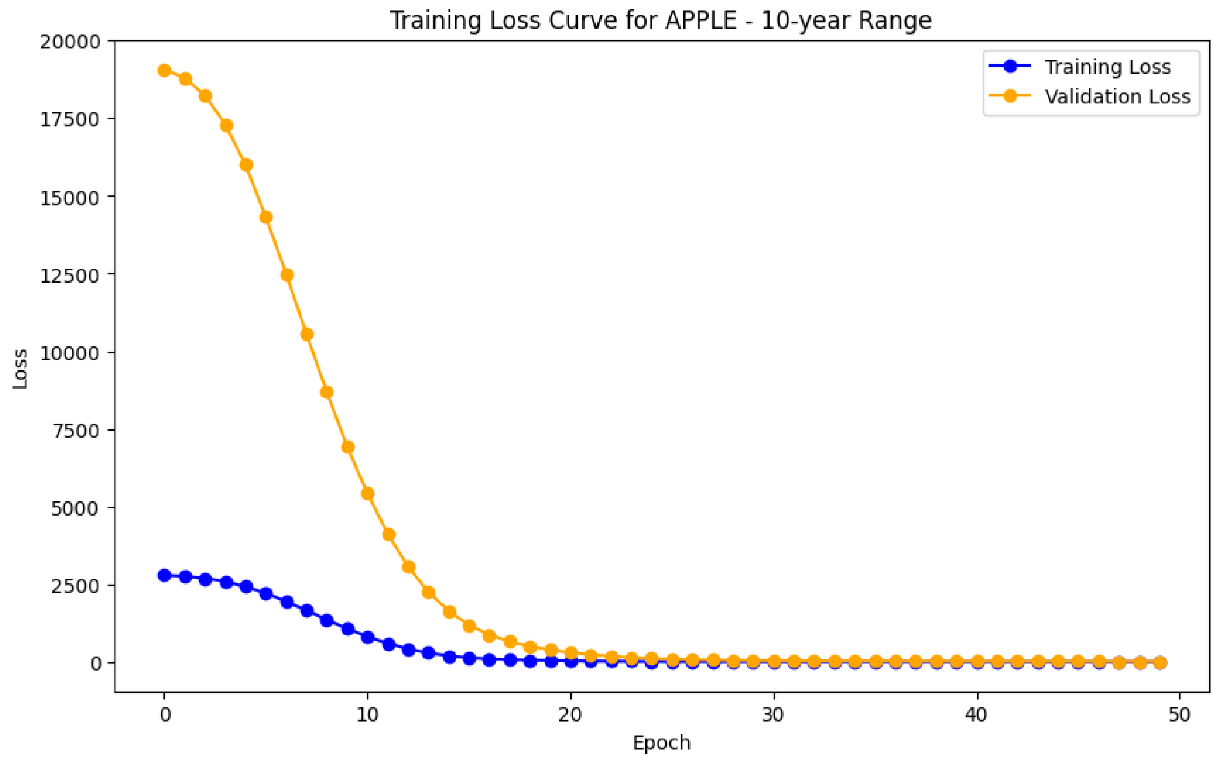

Training Loss Curve for ELECTRONIC ARTS - 5-year Range
Training Loss
Validation Loss
Loss
Epoch
16000
14000
12000
10000
8000
6000
4000
2000
0
10
20
30
40
50

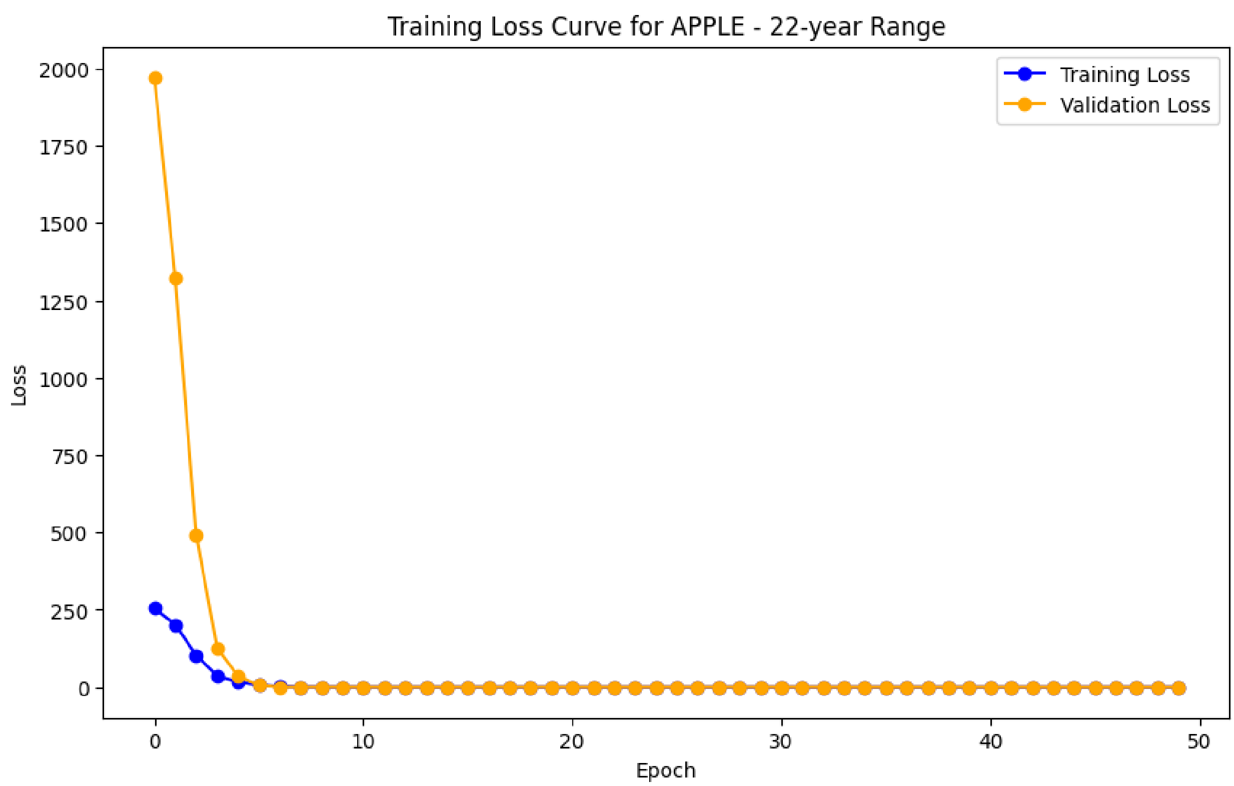

Training Loss Curve for ELECTRONIC ARTS - 10-year Range
Training Loss
Validation Loss
Loss
Epoch
20000
17500
15000
12500
10000
7500
5000
2500
0
0
10
20
30
40
50


Training Loss Curve for ELECTRONIC ARTS - 22-year Range
Training Loss
Validation Loss
Loss
Epoch
12000
10000
8000
6000
4000
2000
0
0
10
20
30
40
50


Training Loss Curve for APPLE - 1-year Range
Training Loss
Validation Loss
Loss
Epoch
32000
30000
28000
26000
24000
0
10
20
30
40
50

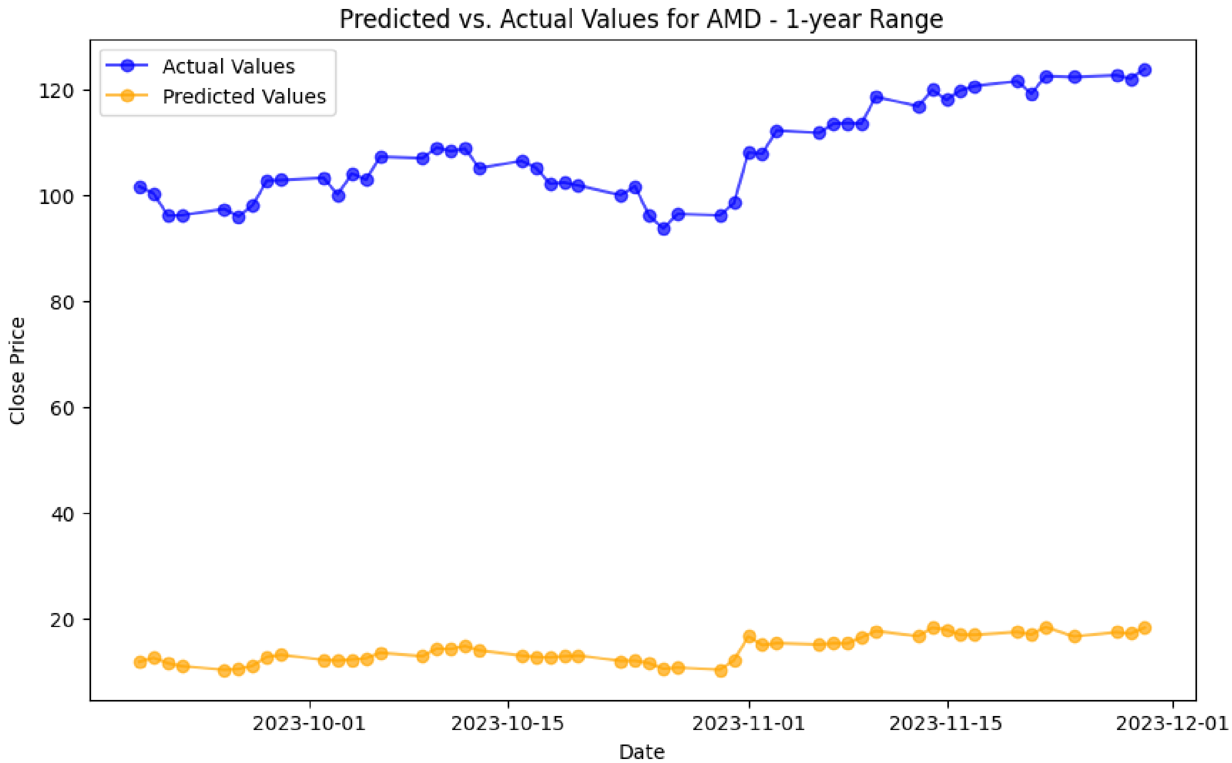

Training Loss Curve for APPLE - 3-year Range
Training Loss
Validation Loss
Loss
Epoch
25000
20000
15000
10000
5000
0
10
20
30
40
50


Training Loss Curve for APPLE - 10-year Range
Training Loss
Validation Loss
20000
17500
15000
12500
10000
7500
5000
2500
0
Loss
0
10
20
30
40
50
Epoch

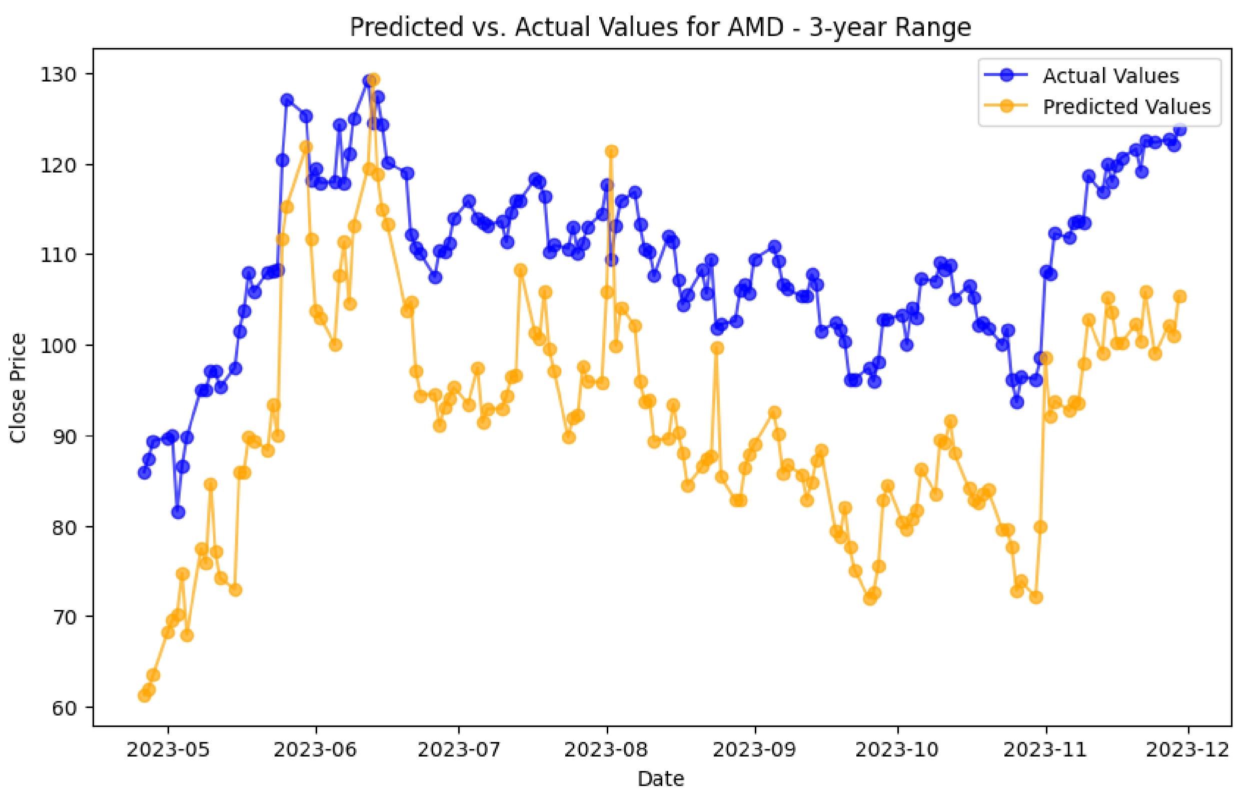

Training Loss Curve for APPLE - 22-year Range
Training Loss
Validation Loss
2000
1750
1500
1250
1000
750
500
250
0
Loss
0
10
20
30
40
50
Epoch

LSTM Evaluation

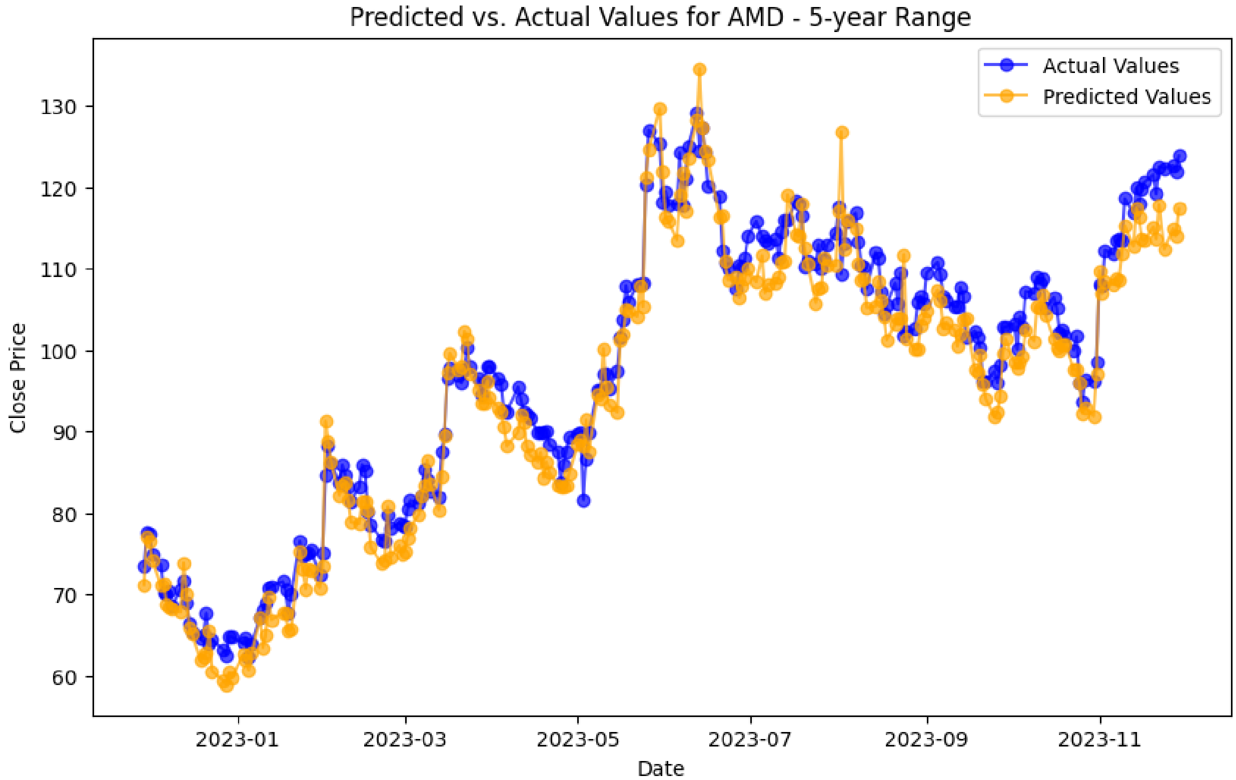
AMD - 1 -year Range - LSTM Evaluation:



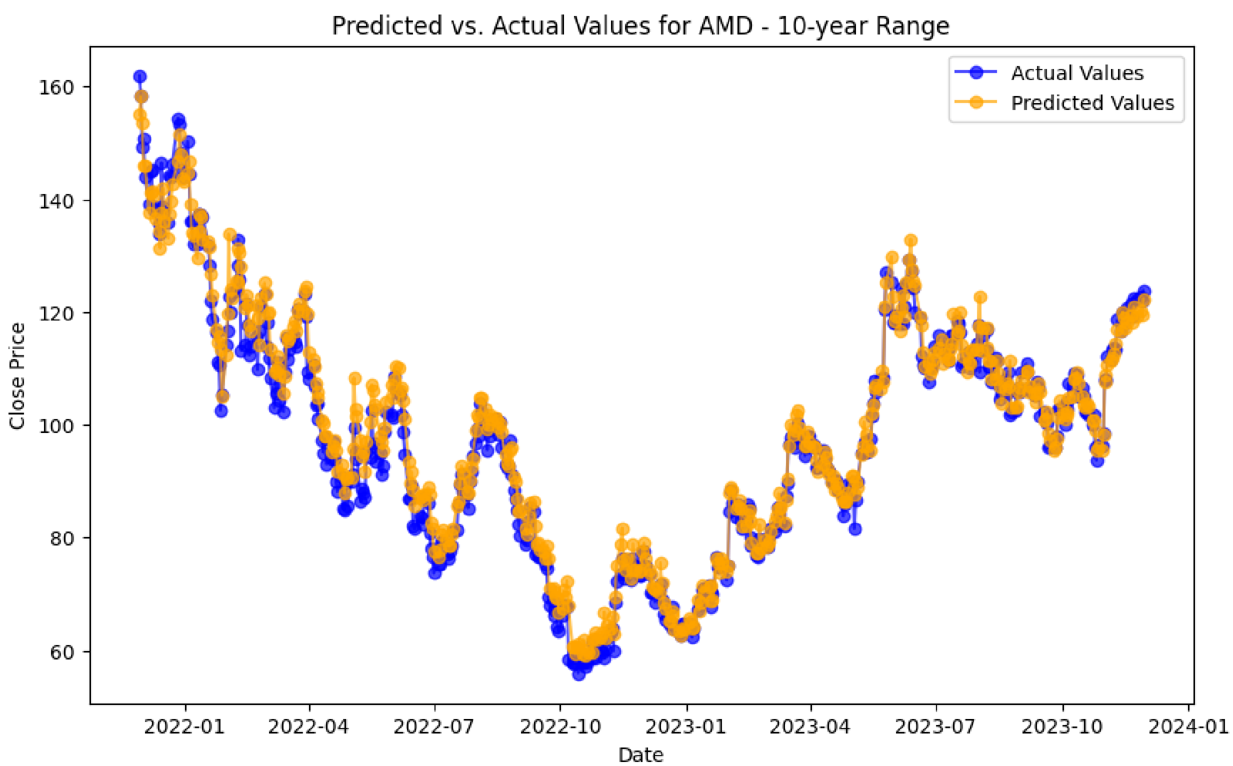
AMD - 3 -year Range - LSTM Evaluation:



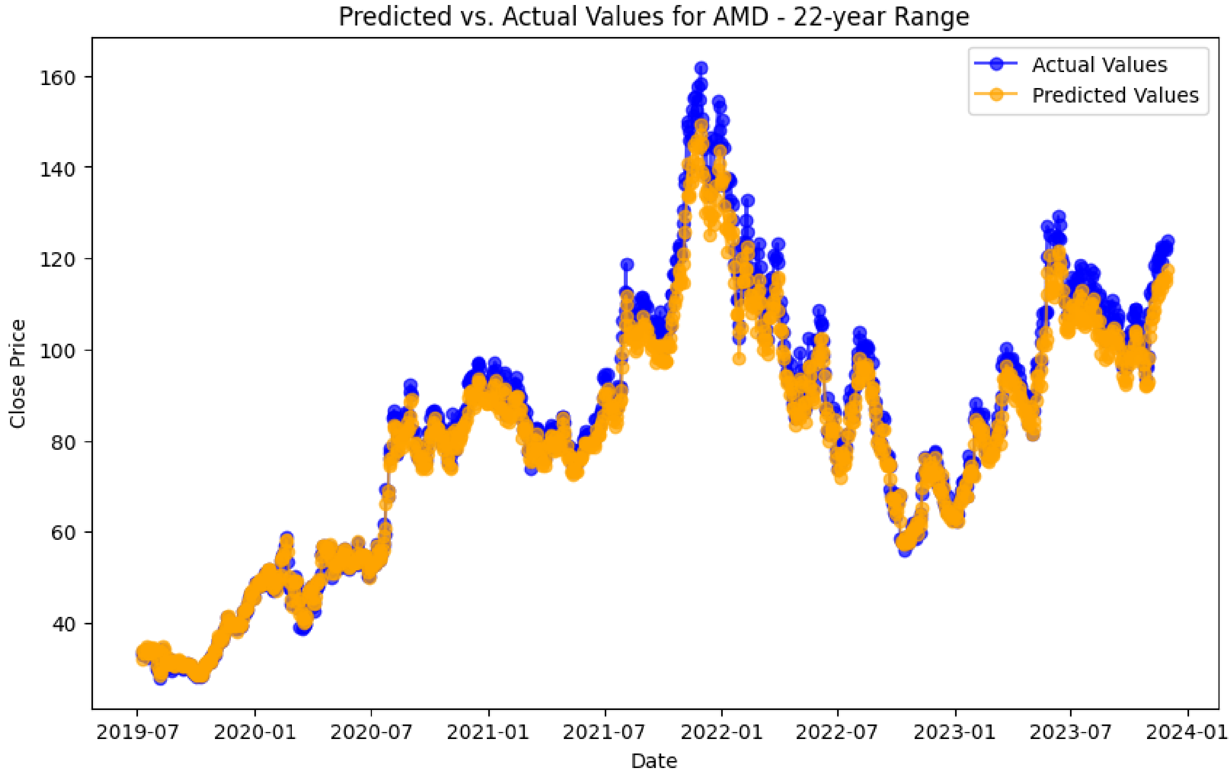
AMD - 5 -year Range - LSTM Evaluation:



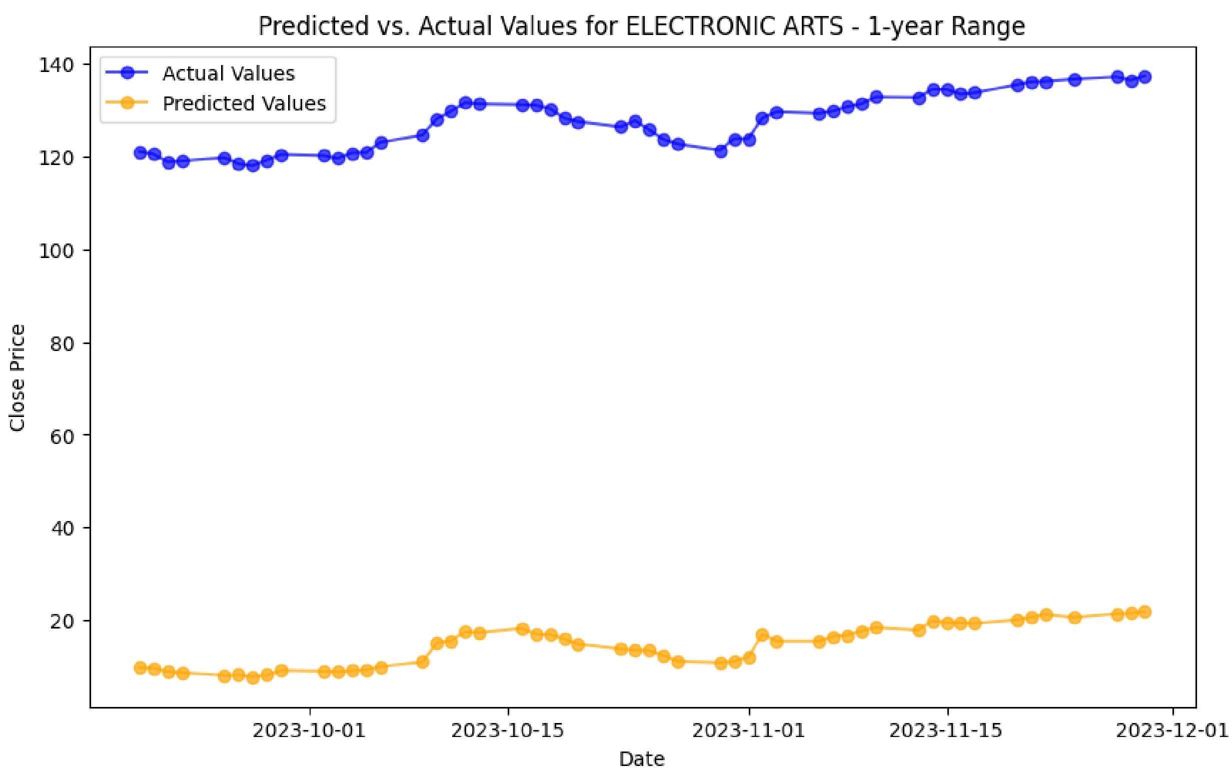
AMD - 10 -year Range - LSTM Evaluation:



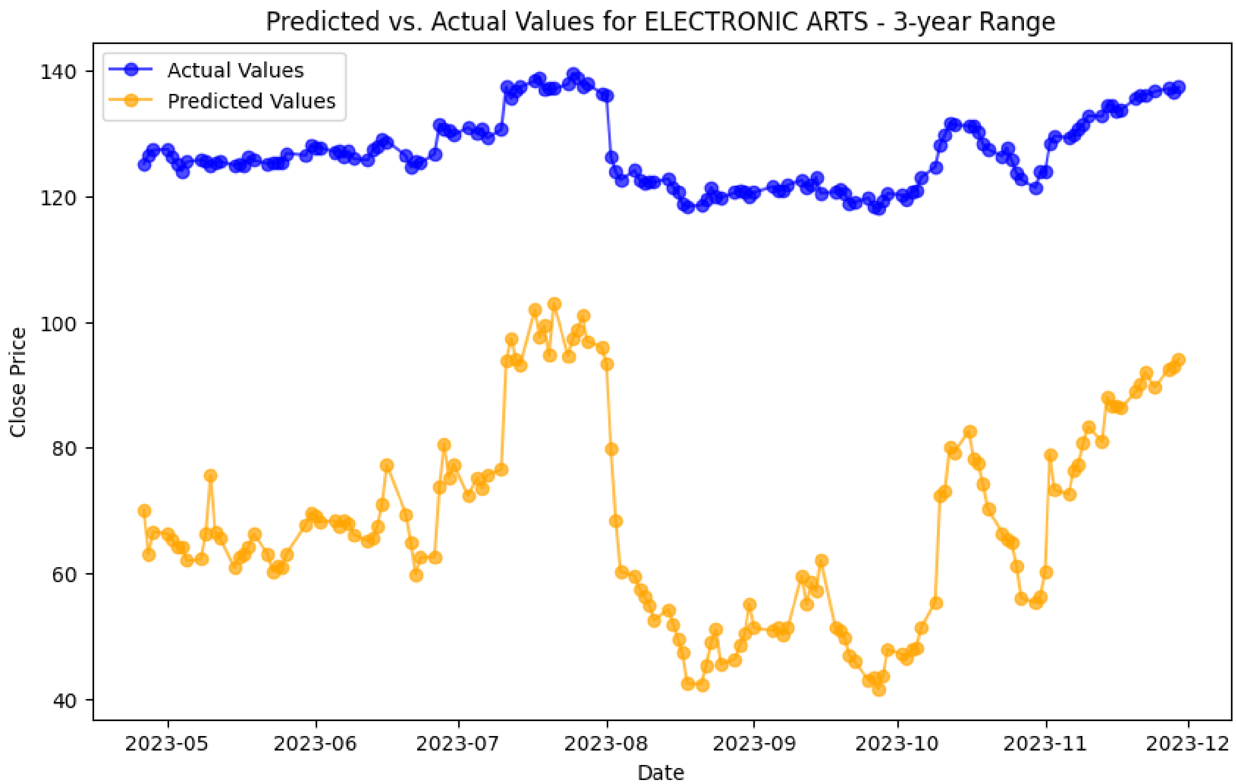
AMD - 22 -year Range - LSTM Evaluation:



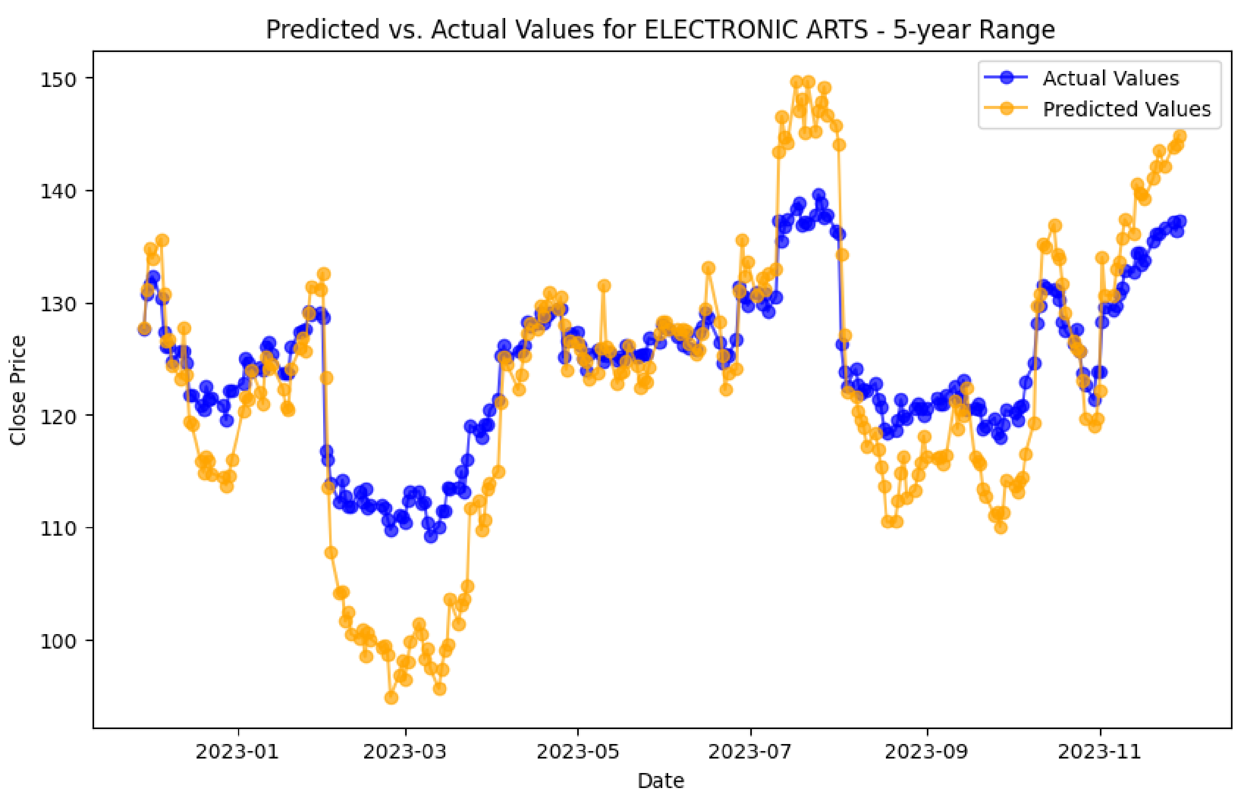
EA - 1 -year Range - LSTM Evaluation:



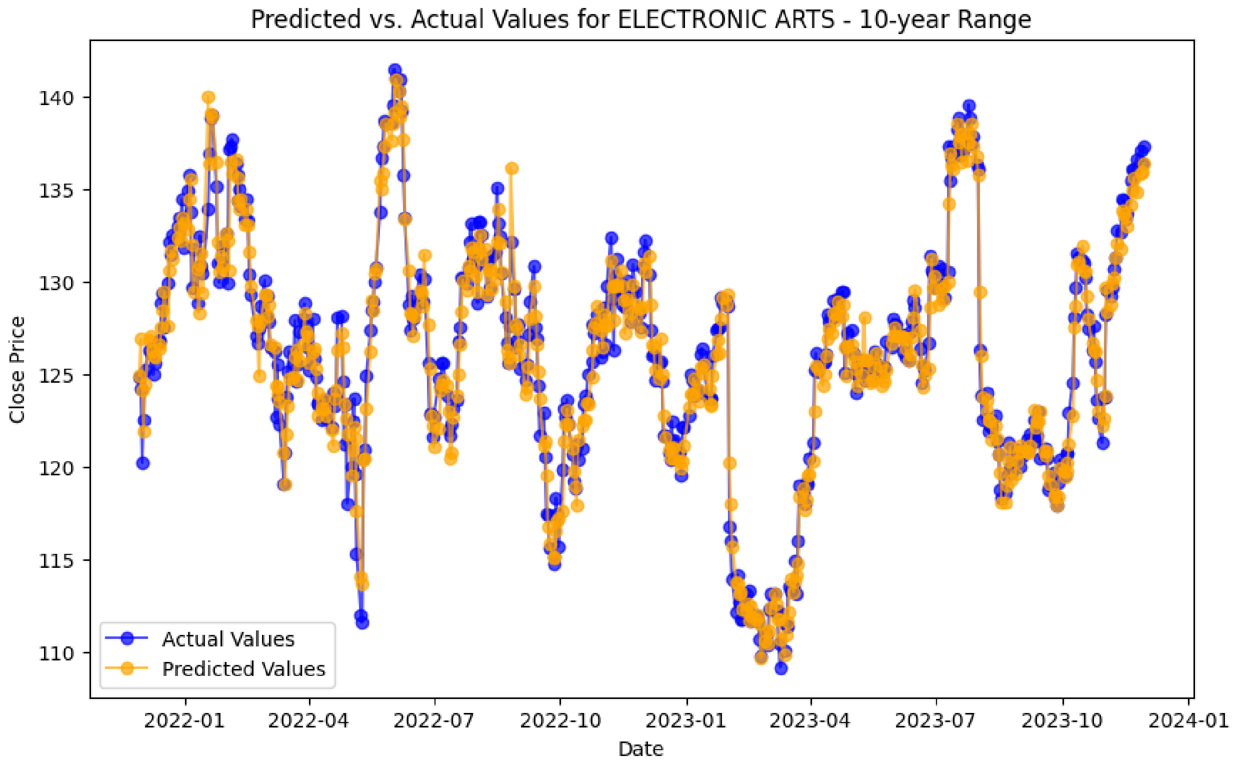
EA - 3 -year Range - LSTM Evaluation:



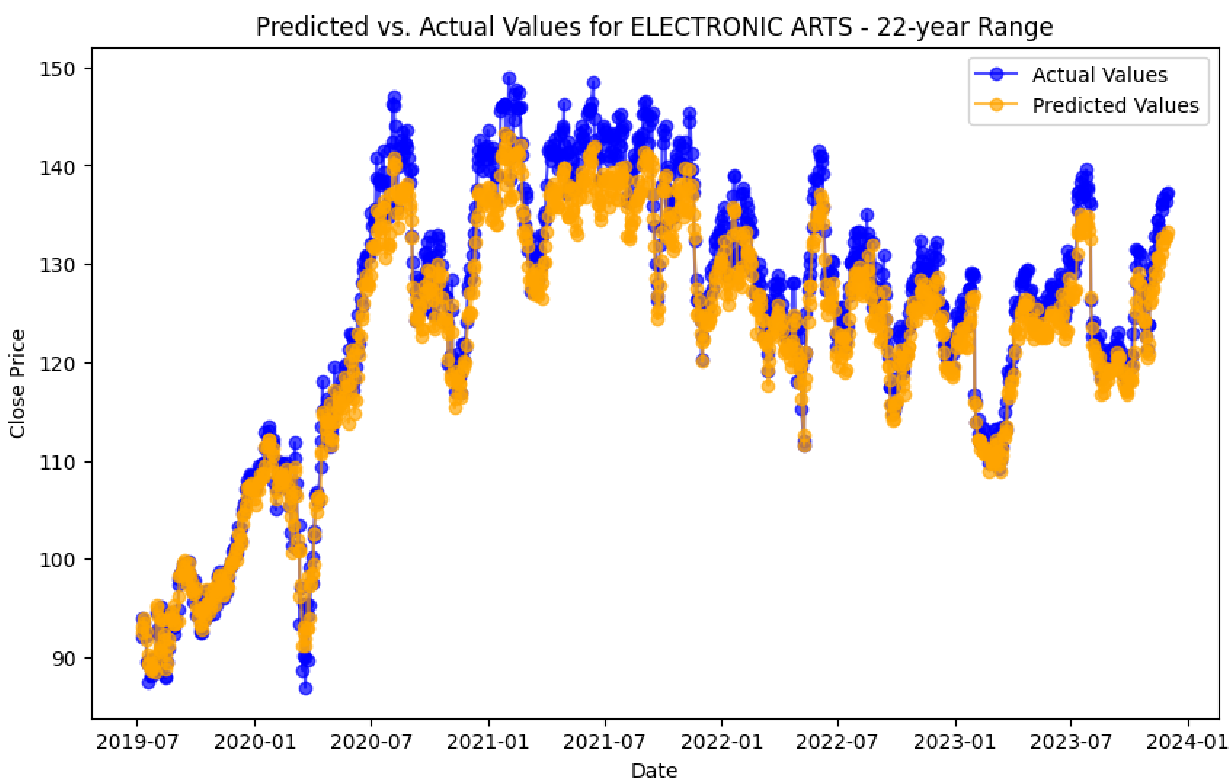
EA - 5 -year Range - LSTM Evaluation:



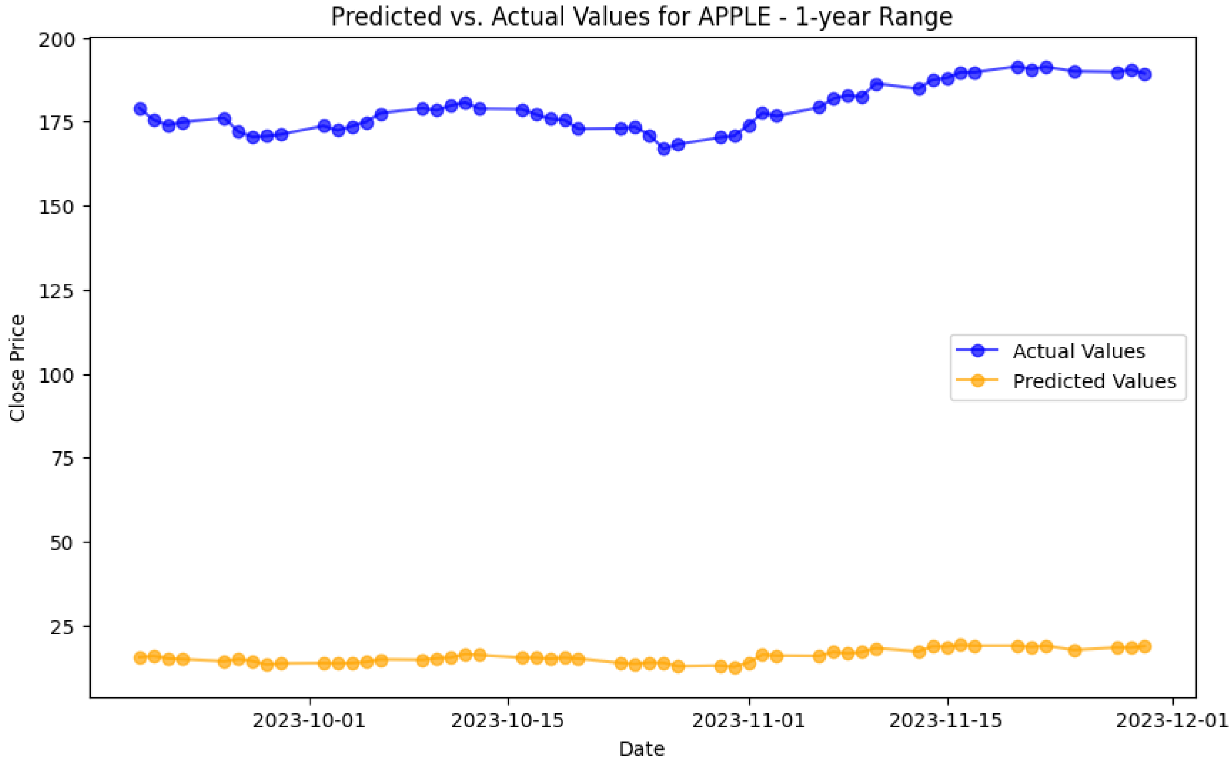
EA - 10 -year Range - LSTM Evaluation:



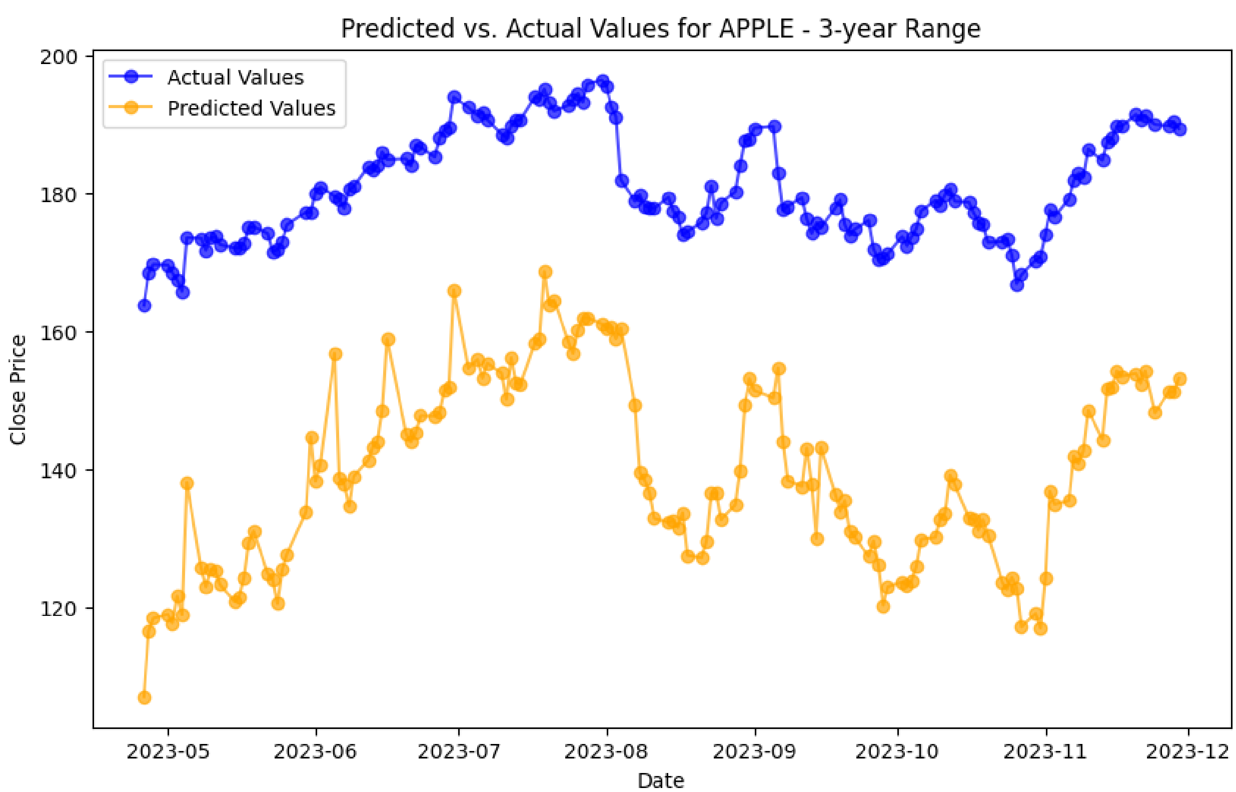
EA - 22 -year Range - LSTM Evaluation:



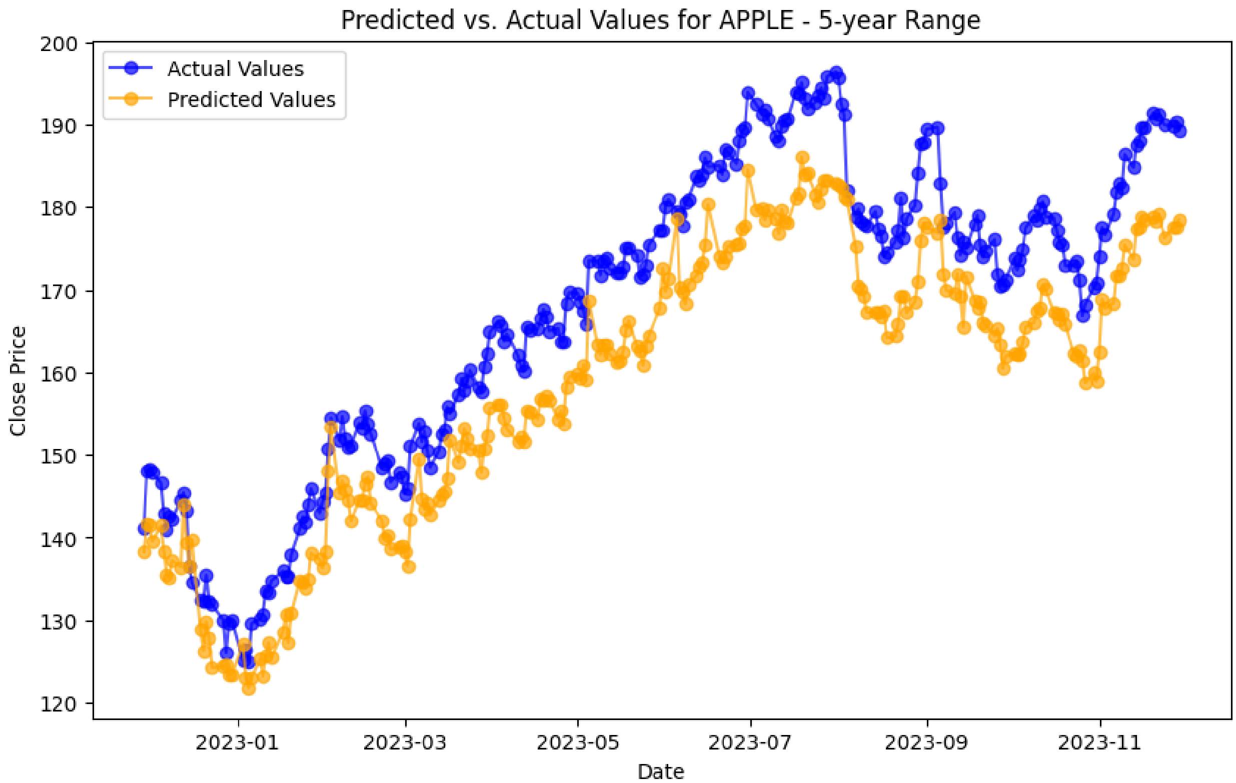
Apple - 1 -year Range - LSTM Evaluation:



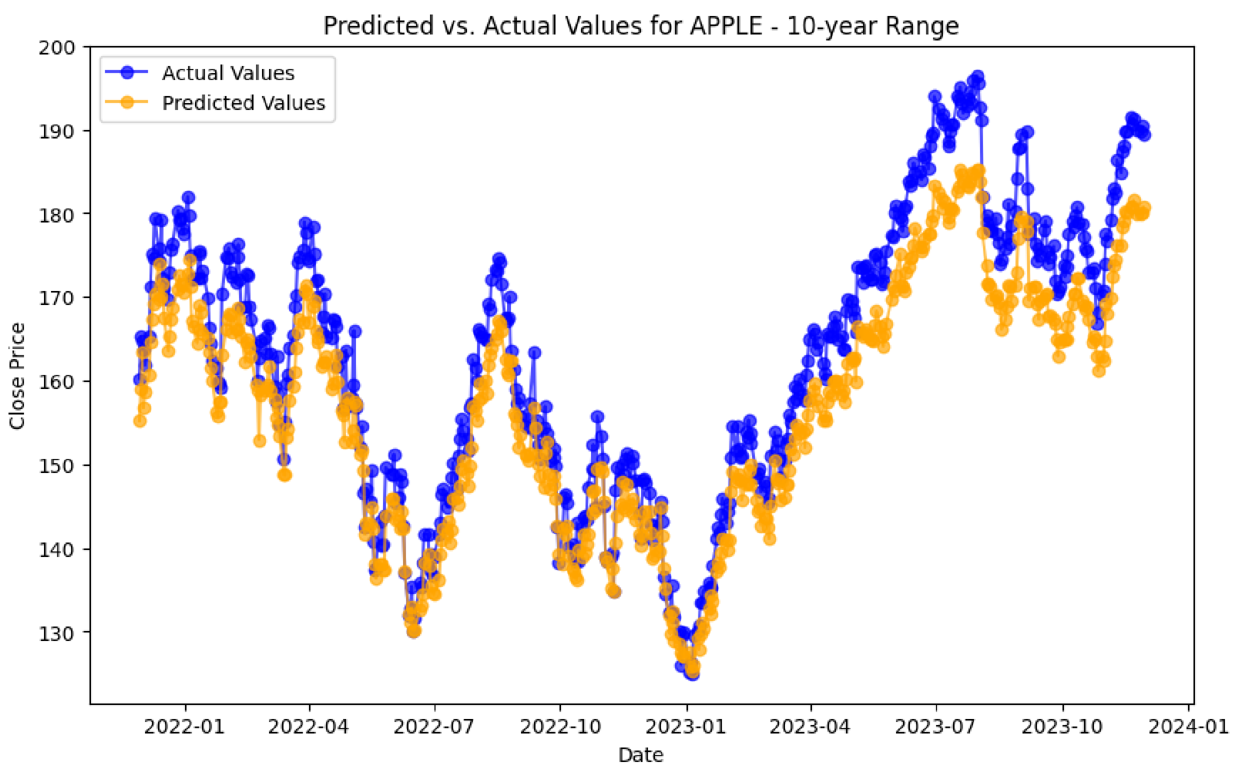
Apple - 3 -year Range - LSTM Evaluation:



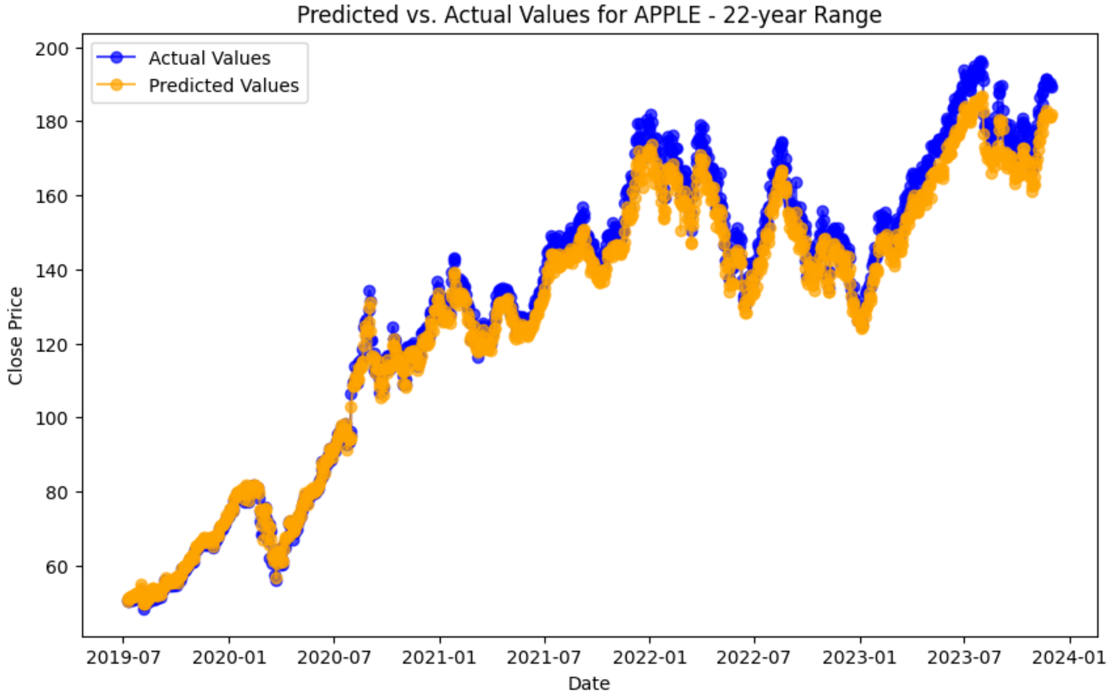
Apple - 5 -year Range - LSTM Evaluation:



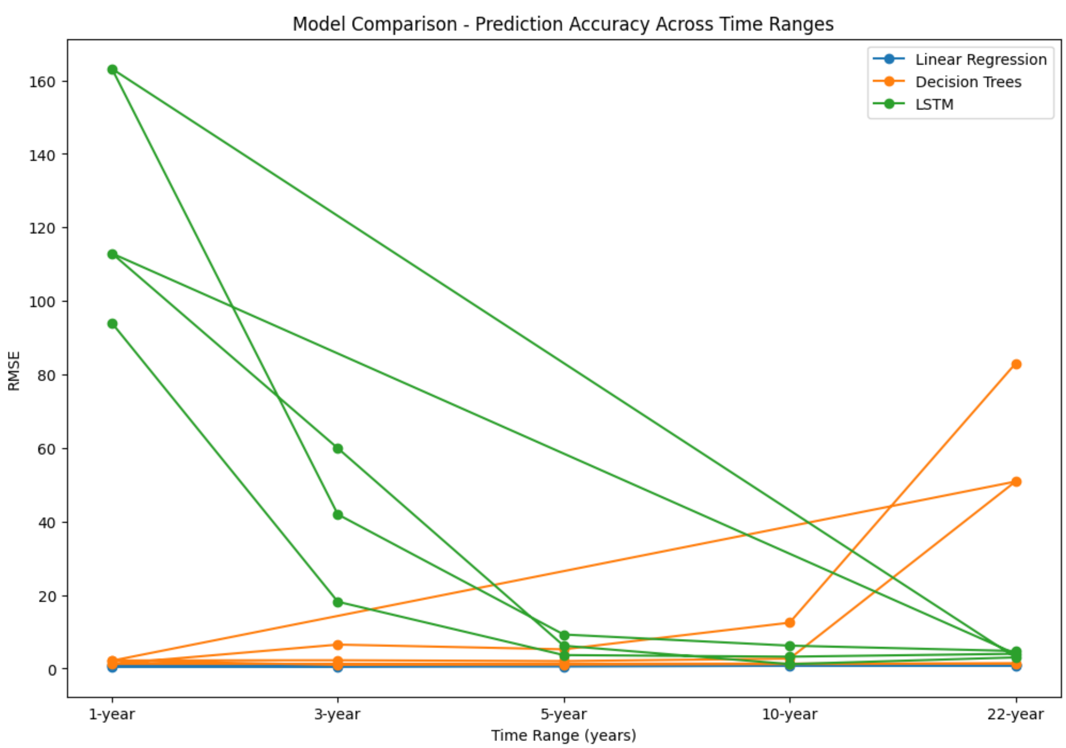
Apple - 10 -year Range - LSTM Evaluation:



Apple -22 -year Range - LSTM Evaluation:



Evaluation

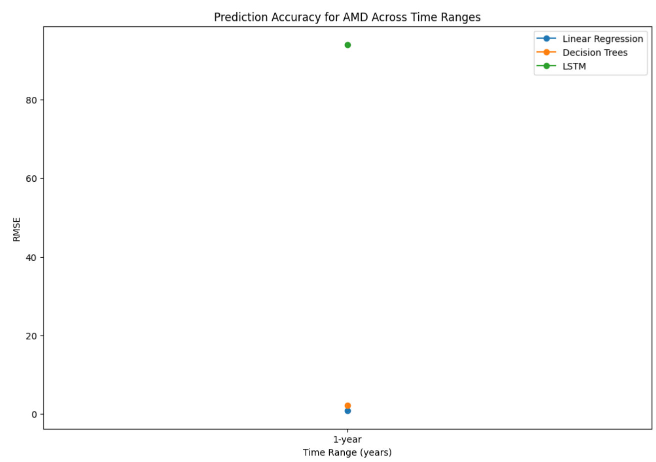


1. Accuracy Analysis for Each Company - Time Frame Basis:

AMD:

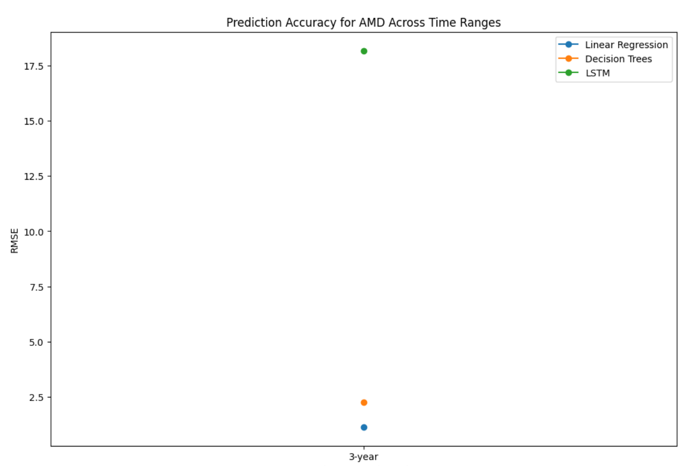
1-year Range:

Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	0.84937	0.768062	0.921613	0.989419	0.726038
Decision Trees	4.89371	1.70137	2.21217	0.939037	1.56892
LSTM	8833.46	93.7538	93.9865	-109.041	86.9162



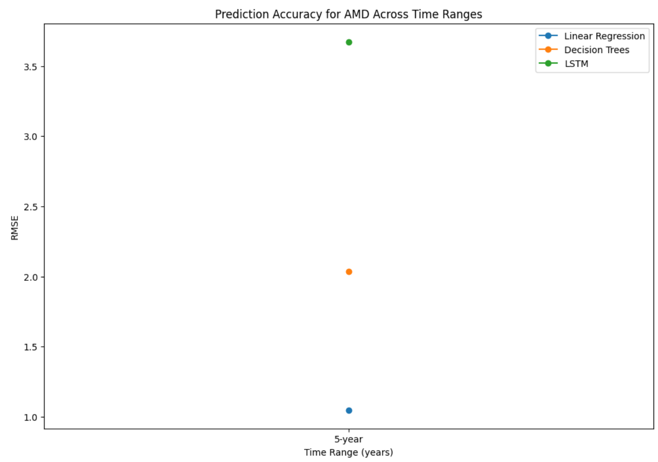
3-year Range:

Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	1.28049	0.892871	1.13159	0.985599	0.826677
Decision Trees	5.11875	1.64907	2.26246	0.942431	1.52051
LSTM	330.534	17.585	18.1806	-2.71739	17.7504



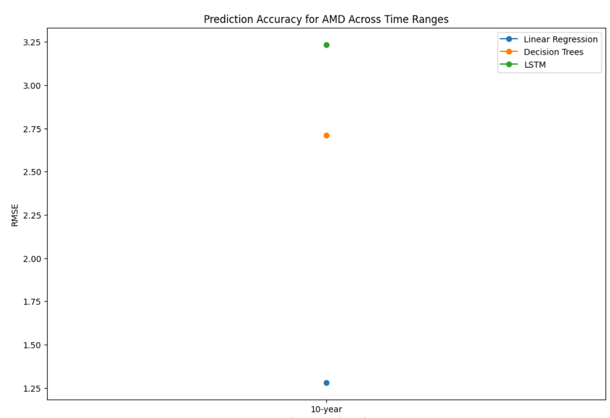
5-year Range:

Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	1.09902	0.817415	1.04834	0.99632	0.849674
Decision Trees	4.14491	1.5156	2.03591	0.986122	1.56735
LSTM	13.488	3.01513	3.67259	0.954839	21.5313



10-year Range:

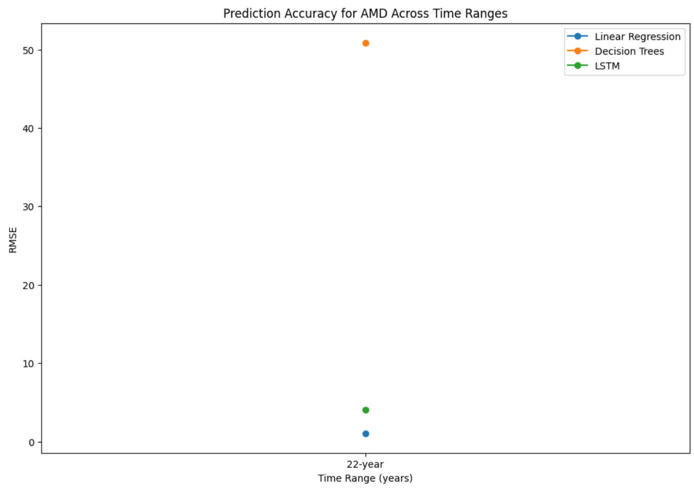
Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	1.64253	0.9848	1.28161	0.996473	1.01362
Decision Trees	7.34742	1.93901	2.71061	0.984222	2.0392
LSTM	10.4594	2.35517	3.2341	0.97754	26.5383



22-year Range:

22-year Range:

Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	1.11568	0.754288	1.05626	0.998652	0.909298
Decision Trees	2588.35	42.9146	50.8758	-2.12819	44.7544
LSTM	16.1933	2.94816	4.02409	0.980429	46.6224

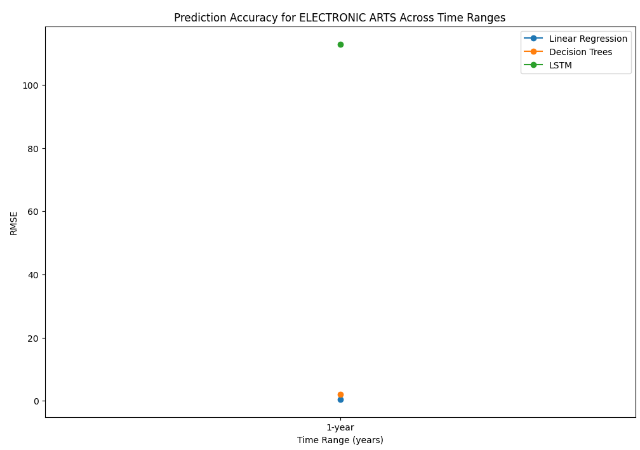


ELECTRONIC ARTS:

ELECTRONIC ARTS:

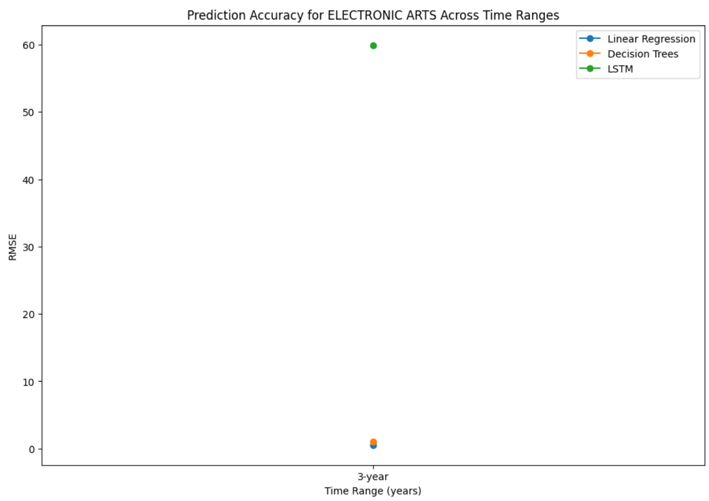
1-year Range:

Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	0.256254	0.418106	0.506215	0.992875	0.327605
Decision Trees	4.6113	1.51059	2.14739	0.871787	1.15934
LSTM	12754.3	112.922	112.935	-353.621	88.5394



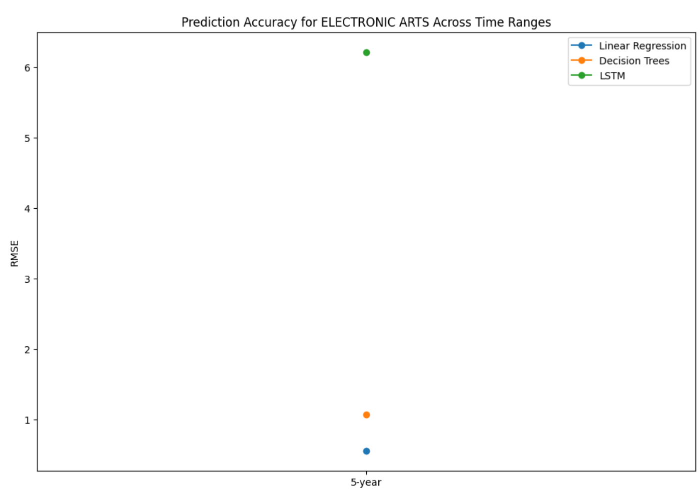
3-year Range:

Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	0.28393	0.437327	0.532851	0.991328	0.343445
Decision Trees	1.0299	0.79543	1.01484	0.968544	0.626351
LSTM	3591.99	59.0437	59.9332	-108.708	46.3823



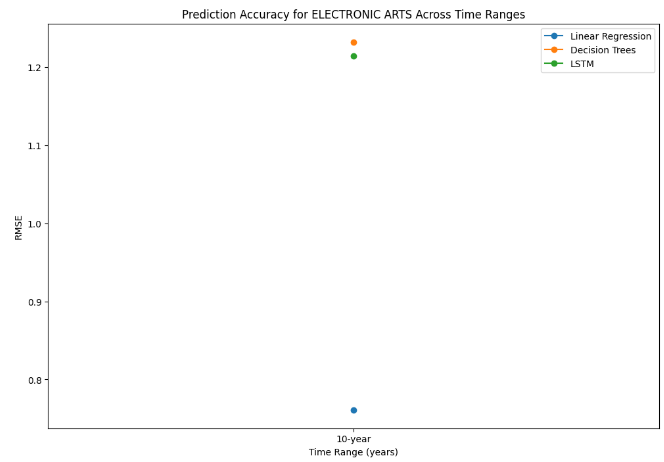
5-year Range:

Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	0.310719	0.434033	0.557422	0.993486	0.348292
Decision Trees	1.14783	0.839563	1.07137	0.975938	0.683194
LSTM	38.6504	4.8192	6.21694	0.189762	9.21826



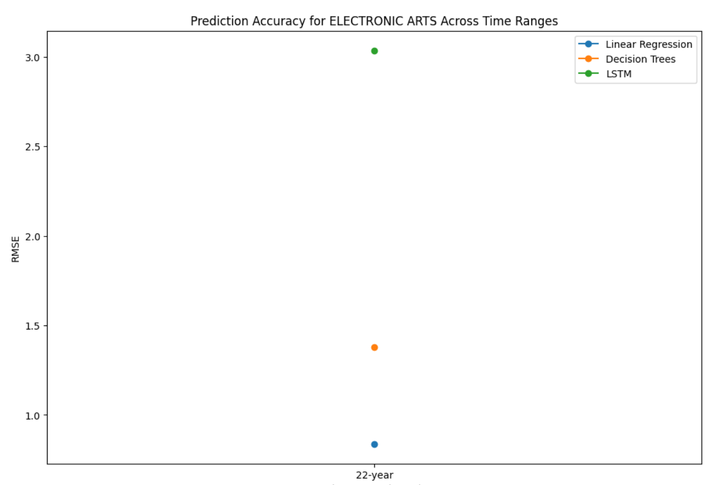
10-year Range:

Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	0.579225	0.579173	0.761068	0.986033	0.457318
Decision Trees	1.51879	0.944603	1.23239	0.963376	0.747632
LSTM	1.47569	0.933944	1.21478	0.964416	5.67585



22-year Range:

Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	0.702021	0.631086	0.837867	0.996609	0.513418
Decision Trees	1.90086	1.03046	1.37872	0.990817	0.83789
LSTM	9.22058	2.58156	3.03654	0.955455	12.7682

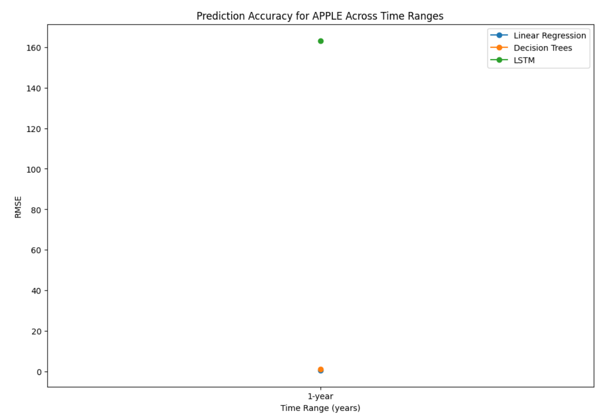


APPLE:

APPLE:

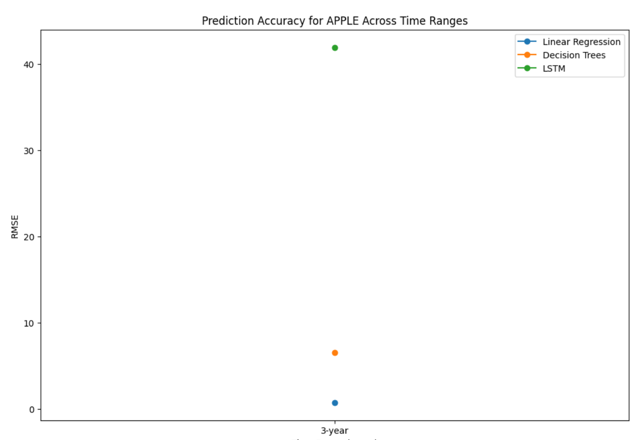
1-year Range:

Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	0.523755	0.60397	0.723709	0.989277	0.340022
Decision Trees	1.62618	0.993724	1.27522	0.966706	0.563948
LSTM	26622.2	163.08	163.163	-544.061	91.1842



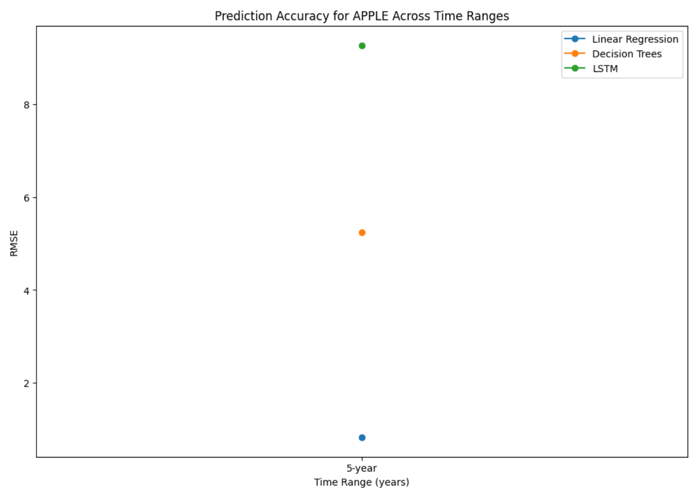
3-year Range:

Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	0.540234	0.598289	0.735006	0.991287	0.332517
Decision Trees	42.4986	4.45305	6.5191	0.31461	2.36747
LSTM	1758.53	41.4066	41.9348	-27.3604	22.7756



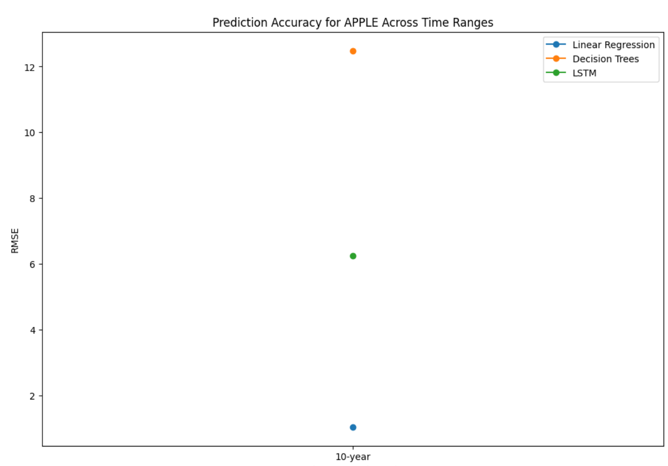
5-year Range:

Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	0.695427	0.670298	0.833923	0.997978	0.413099
Decision Trees	27.4654	3.10558	5.24075	0.920135	1.70877
LSTM	85.8328	8.84173	9.2646	0.750411	12.878



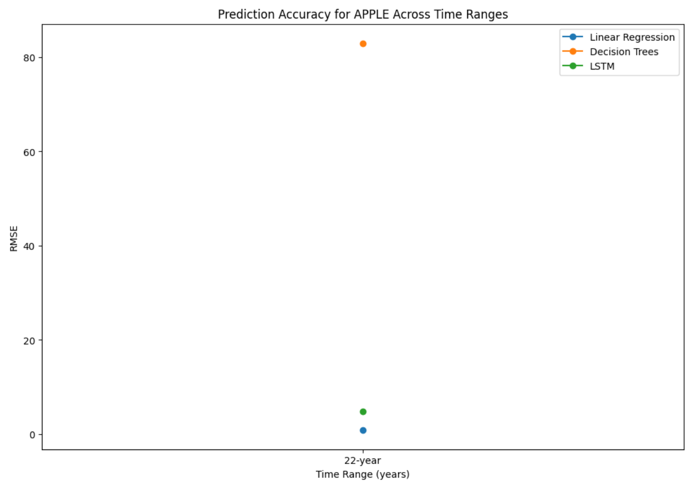
10-year Range:

Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	1.07986	0.827446	1.03916	0.996102	0.519974
Decision Trees	155.884	8.53936	12.4854	0.437236	4.81221
LSTM	38.9302	5.60743	6.23941	0.859457	11.304



22-year Range:

Model	MSE	MAE	RMSE	R2 Score	MAPE
Linear Regression	0.840753	0.677252	0.916926	0.999476	0.53742
Decision Trees	6882.47	72.9967	82.9607	-3.28717	49.7372
LSTM	22.8428	3.8483	4.77941	0.985771	41.8432



2. Overall Accuracy Analysis - Time Frame Basis:

2. Overall Accuracy Analysis - Time Frame Basis:

Time Range	RMSE	MAE	MSE	MAPE	R2 Score
1-year	41.9856	41.7502	5358.08	30.1473	-111.219
10-year	3.35539	2.52344	24.3242	5.90091	0.907206
22-year	16.6518	14.2647	1058.18	22.0582	0.165761
3-year	14.6938	14.0957	636.867	10.325	-14.8435
5-year	3.32687	2.67317	19.2038	5.46644	0.862777

3. Overall Accuracy Analysis - Model Basis:

3. Overall Accuracy Analysis - Model Basis:

Company Name	Time Range	Model	RMSE
APPLE	1-year	Decision Trees	1.27522
ELECTRONIC ARTS	10-year	Decision Trees	1.23239
ELECTRONIC ARTS	22-year	Decision Trees	1.37872
ELECTRONIC ARTS	3-year	Decision Trees	1.01484
ELECTRONIC ARTS	5-year	Decision Trees	1.07137
AMD	1-year	LSTM	93.9865
ELECTRONIC ARTS	10-year	LSTM	1.21478
ELECTRONIC ARTS	22-year	LSTM	3.03654
AMD	3-year	LSTM	18.1806
AMD	5-year	LSTM	3.67259
ELECTRONIC ARTS	1-year	Linear Regression	0.506215
ELECTRONIC ARTS	10-year	Linear Regression	0.761068
ELECTRONIC ARTS	22-year	Linear Regression	0.837867
ELECTRONIC ARTS	3-year	Linear Regression	0.532851
ELECTRONIC ARTS	5-year	Linear Regression	0.557422

4. Best Model and Time Frame for Each Company:

Company Name	Time Range	Model	RMSE
AMD	1-year	Linear Regression	0.921613
APPLE	1-year	Linear Regression	0.723709
ELECTRONIC ARTS	1-year	Linear Regression	0.506215

5. Overall Best Model and Time Frame:

15
ELECTRONIC ARTS
1-year
Linear Regression
0.5062152105293628

6. Time Frame with Maximum Variability (High Standard Deviation):

For Linear Regression:

Time Range	RMSE
10-year	1.28161

For Decision Trees:

Time Range	RMSE
22-year	82.9607

For LSTM:

Time Range	RMSE
1-year	163.163

3. Conclusion

In analyzing the performance of various machine learning models on predicting stock prices for AMD, Electronic Arts, and Apple across different time ranges (1-year, 3-year, 5-year, 10-year, and 22-year), several important observations were made.

1. AMD Stock Analysis:

1-year Range:
Linear Regression and Decision Trees models exhibit high accuracy, with low Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).LSTM model performs poorly, suggesting that its architecture might not be suitable for short-term predictions.

3-year Range:
Similar to the 1-year range, Linear Regression and Decision Trees models show high accuracy, while the LSTM model's performance is significantly worse.

5-year Range:
The Linear Regression model excels in accuracy, indicating that it performs well in predicting AMD stock prices over this period.Decision Trees also perform well, and the LSTM model shows a decline in accuracy.

10-year Range:
Linear Regression continues to perform exceptionally well, maintaining high accuracy. Decision Trees remain competitive, while the LSTM model shows a decrease in performance.

22-year Range:
Linear Regression demonstrates outstanding accuracy, reaffirming its suitability for long-term predictions. Decision Trees perform reasonably well, whereas the LSTM model struggles, suggesting challenges in capturing long-term trends.

2. Electronic Arts (EA) Stock Analysis:

1-year Range:
Linear Regression and Decision Trees models exhibit high accuracy, while the LSTM model performs poorly.

3-year Range:
Linear Regression and Decision Trees models continue to demonstrate high accuracy. The LSTM model's performance is notably worse, suggesting challenges in capturing EA's stock behavior over three years.

5-year Range:
Linear Regression and Decision Trees maintain high accuracy. The LSTM model's performance is significantly worse than the other models, indicating challenges in predicting EA's stock prices over a more extended period.

10-year Range:
Linear Regression excels in accuracy, while Decision Trees also perform well. The LSTM model shows a decrease in accuracy over this extended time range.

22-year Range:
Linear Regression remains highly accurate, proving its effectiveness for long-term predictions.Decision Trees perform reasonably well, while the LSTM model faces challenges in capturing EA's long-term stock trends.

3. Apple Stock Analysis:

1-year Range:
Linear Regression and Decision Trees models exhibit high accuracy, while the LSTM model performs poorly.

3-year Range:
Linear Regression and Decision Trees models continue to demonstrate high accuracy. The LSTM model's performance is notably worse, indicating challenges in capturing Apple's stock behavior over three years.

5-year Range:
Linear Regression and Decision Trees maintain high accuracy.The LSTM model's performance is significantly worse than the other models, indicating challenges in predicting Apple's stock prices over a more extended period.
10-year Range:
Linear Regression excels in accuracy, while Decision Trees also perform well.The LSTM model shows a decrease in accuracy over this extended time range.
22-year Range:
Linear Regression remains highly accurate, proving its effectiveness for long-term predictions.Decision Trees perform reasonably well, while the LSTM model faces challenges in capturing Apple's long-term stock trends.

Overall Observations:

- Linear Regression consistently proves to be a robust model across all time ranges for all three companies, demonstrating high accuracy.
- Decision Trees also perform well, especially in the 1-year and 3-year ranges.
- LSTM models struggle to capture the complexities of stock price movements, especially over longer time frames.

Recommendations:

- For short to medium-term predictions (1 to 5 years), Linear Regression or Decision Trees may be more suitable.
- For longer-term predictions (10 to 22 years), Linear Regression remains a reliable choice, while Decision Trees might also provide acceptable results.
- Further fine-tuning of the LSTM model's architecture and hyperparameters may be necessary to improve its performance across different time ranges.