Industrial Internship Report on

”Stock Price Prediction”

Prepared by

Diptendu Som

Executive Summary
This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT). This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks’ time. My project was predicting the price of a stock. This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

1 Preface

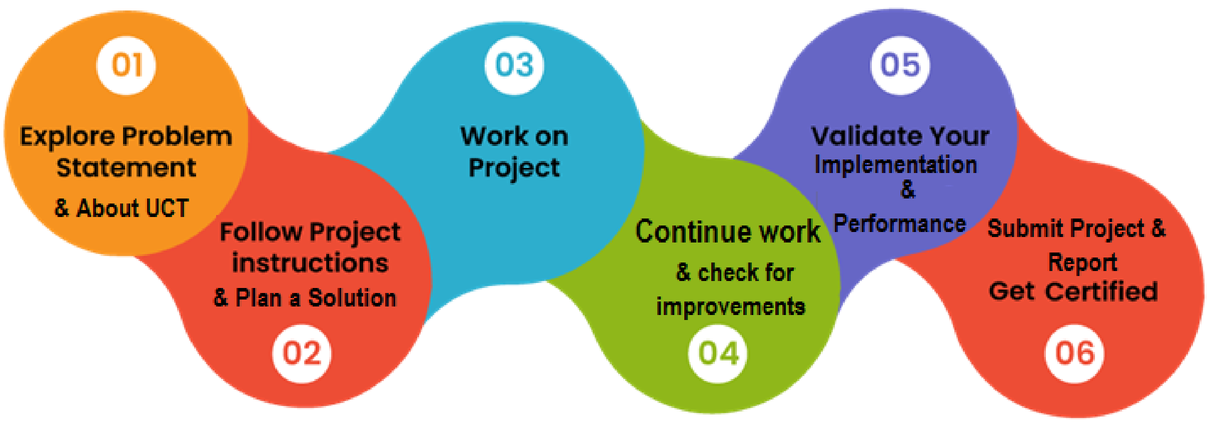
Summary of the whole 6 weeks' work.

About need of relevant Internship in career development.

Brief about Your project/problem statement.

Opportunity given by USC/UCT.

How Program was planned



Your Learnings and overall experience.

Thank to all , who have helped you directly or indirectly.

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.

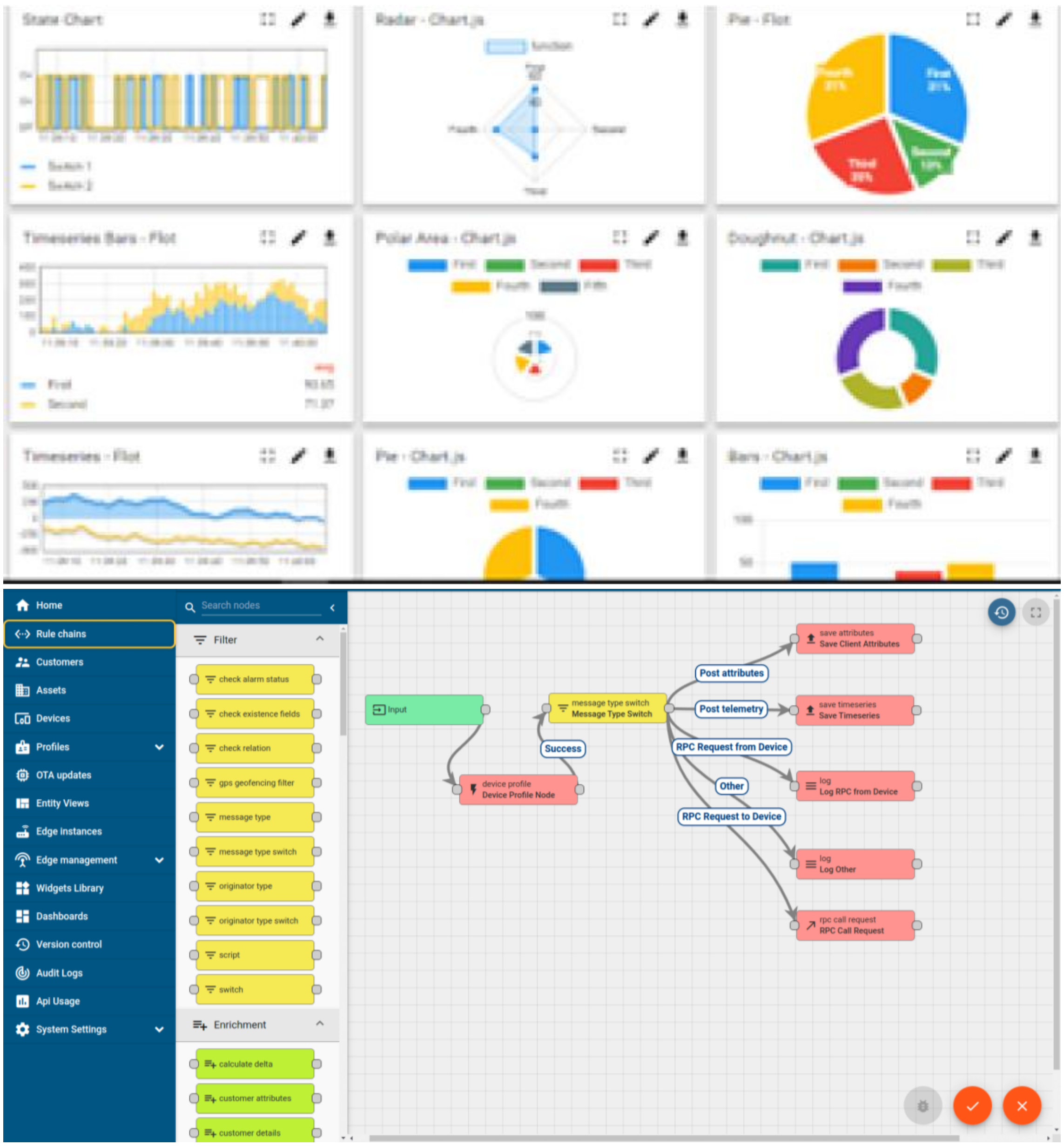
i. UCT IoT Platform (uct Insight)

UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to
• Build Your own dashboard

- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine



ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

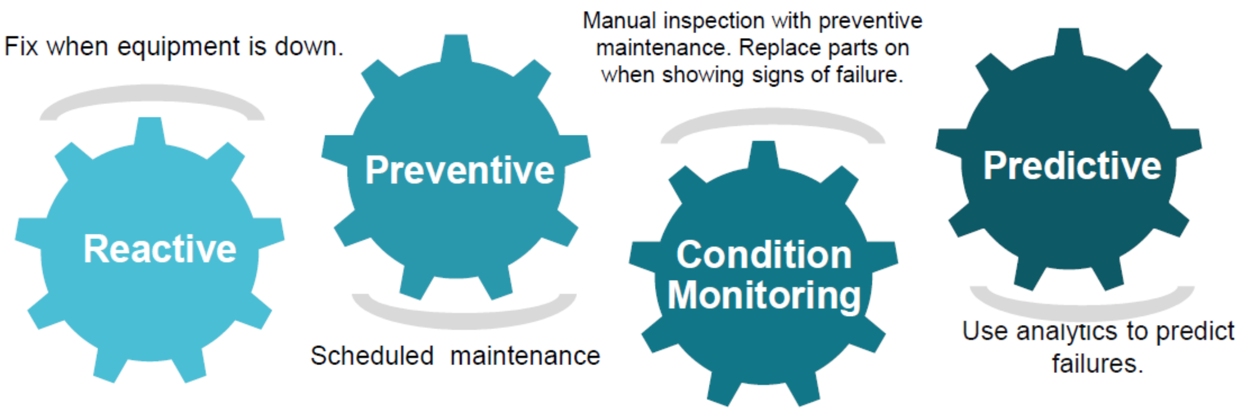
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i

iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

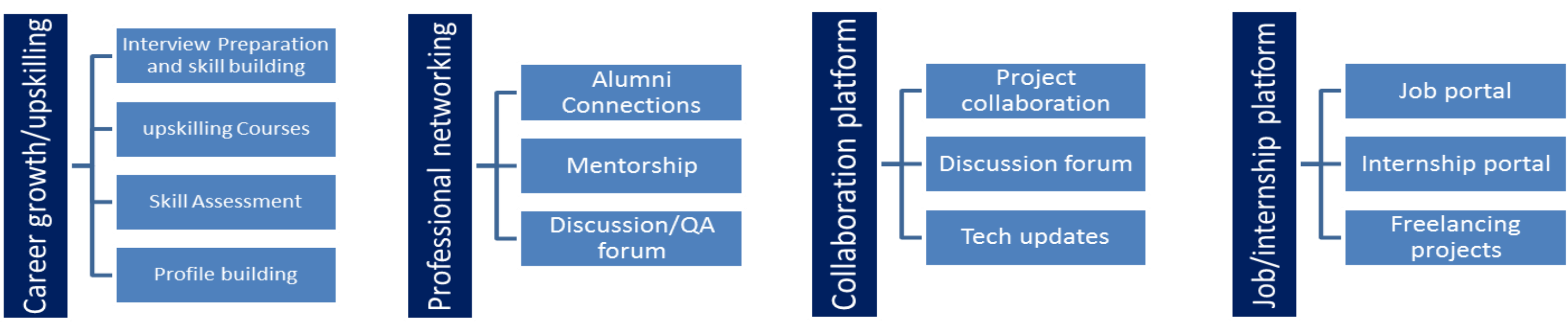
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

☛ get practical experience of working in the industry.

☛ to solve real world problems.

☛ to have improved job prospects.

☛ to have Improved understanding of our field and its applications.

☛ to have Personal growth like better communication and problem solving.

2.5 Reference

[1] Youtube

[2] Yahoo Stocks

[3] Wikipedia

2.6 Glossary

Terms	Acronym

3 Problem Statement

In the assigned problem statement

In the realm of financial markets, accurate stock price prediction remains a critical challenge due to the complex, dynamic, and nonlinear nature of market behavior. Investors, traders, and financial analysts seek robust predictive models to forecast stock prices, aiming to maximize returns and mitigate risks. This project focuses on developing a machine learning model to predict stock prices by leveraging historical market data and advanced predictive techniques. The goal is to build, train, and evaluate a model that can effectively capture underlying patterns and trends in stock prices, providing reliable forecasts that can assist in informed decision-making for stakeholders in the financial sector.

4 Proposed solution

Data Collection:

The first step in developing a stock price prediction model involves collecting historical stock market data. This data includes daily closing prices, trading volumes, opening and closing prices, and other relevant financial metrics. Sources for this data typically include financial market databases, stock exchanges, and financial APIs like Alpha Vantage, Yahoo Finance, or Quandl.

Data Processing:

1.Data Cleaning: Handling missing values, removing outliers, and correcting any inconsistencies in the data. Feature Engineering: Creating additional features that may help the model, such as moving averages, trading volume averages, and other technical indicators. 2. Normalization: Scaling the features to a standard range, usually between 0 and 1, to ensure that the model treats all features equally and improves the convergence speed of the algorithm. 3. Splitting the Dataset: Dividing the data into training and testing sets, typically in an 80:20 ratio, to evaluate the model's performance on unseen data.

Machine Learning Algorithm:

Model Architecture: Designing the LSTM network architecture, which involves defining the number of LSTM layers, the number of neurons in each layer, and other hyperparameters. 1. Training the Model: Using the processed training dataset to train the LSTM model. This involves forward propagation, backpropagation, and optimization techniques like Adam to minimize the loss function. 2. 3.Validation: Monitoring the model's performance on a validation set to fine-tune hyperparameters and prevent overfitting.

Deployment: Model Serialization:

Saving the trained model in a suitable format, such as Keras's HDF5 format, for future use. API Development: Creating an API using frameworks like Flask or FastAPI to expose the model's prediction functionality to external applications. Hosting: Deploying the API on a cloud platform like AWS, Azure, or Google Cloud to ensure scalability and accessibility. User Interface: Developing a user-friendly interface, such as a web application or a mobile app, where users can input stock symbols and receive predicted prices.

Evaluation:

Metrics: Using evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) to quantify the model's prediction accuracy. Backtesting: Comparing the model's predictions with actual historical stock prices to assess its performance over different time periods and market conditions. Continuous Monitoring: Implementing a monitoring system to track the model's performance in real-time and retrain it periodically with new data to maintain its accuracy and relevance.

4.1 Code submission (Github link)

Stock Price Prediction

4.2 Report submission (Github link) : first make placeholder, copy the link.

Stock Price Prediction.pdf

5 Proposed Design/ Model

5.1 **Algorithm Selection** The chosen algorithm for stock price prediction is a Long Short-Term Memory (LSTM) neural network. LSTM is a type of recurrent neural network (RNN) well-suited for time series forecasting due to its ability to capture long-term dependencies in sequential data.

Data Input Data Loading: The historical stock price data is loaded from a CSV file using the Pandas library. This data includes information such as date, open price, high price, low price, close price, and volume. Data Preprocessing: The data is preprocessed by selecting relevant features and scaling them using the MinMaxScaler from Scikit-learn to ensure that all values are within the range [0, 1].

Training Process The data is split into training and test sets. The majority of the data is used for training, and a smaller portion is reserved for testing the model' s performance. Sequences of a specified length (e.g., 60 days) are generated from the training data to create input-output pairs for the LSTM model. An LSTM model is constructed using Keras with an input layer, LSTM layers, and dense layers. The model is compiled using the mean squared error loss function and the Adam optimizer. It is then trained on the training data.

Prediction Process Preparing Test Data: Similar to the training data, sequences are generated from the test data for prediction. Making Predictions: The trained LSTM model is used to make predictions on the test data. Evaluating the Model: The performance of the model is evaluated using metrics such as the root mean squared error (RMSE).

5.2 Price Levels

Figure 1: Current price of the stock for the last 100 days

5.3 Stock Price Level

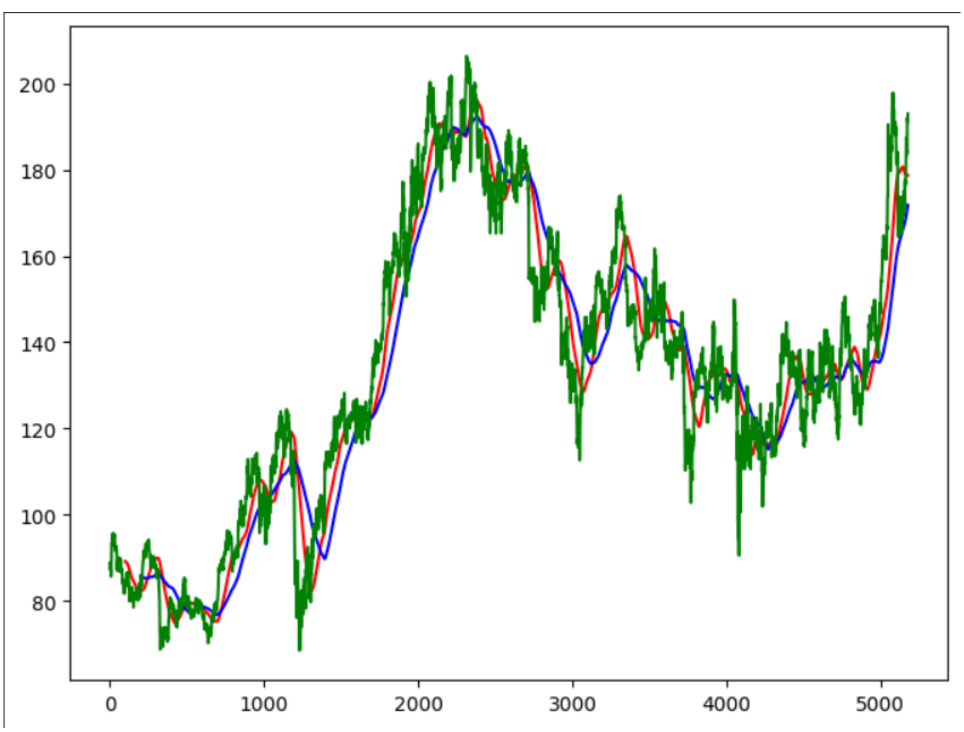


Figure 2: Current price of the stock for the last 200 days

5.4 Interfaces

```
from keras.layers import Dense, Dropout, LSTM
from keras.models import Sequential
```

```
model = Sequential()
model.add(LSTM(units = 50, activation = 'relu', return_sequences = True,
               input_shape = ((x.shape[1],1))))
model.add(Dropout(0.2))

model.add(LSTM(units = 60, activation='relu', return_sequences = True))
model.add(Dropout(0.3))

model.add(LSTM(units = 80, activation = 'relu', return_sequences = True))
model.add(Dropout(0.4))

model.add(LSTM(units = 120, activation = 'relu'))
model.add(Dropout(0.5))

model.add(Dense(units =1))
```

6 Performance Test

This is very important part and defines why this work is meant of Real industries, instead of being just academic project.

Testing was conducted in the machine learning model itself and also the fact that It was working smoothly under most of the conditions.

Constraints : Memory : 8 GB, Storage : 256 GB, Python 3.10, V Memory : 128 MB

6.1 Test Plan/ Test Cases

```
model.fit(x,y, epochs = 50, batch_size =32, verbose =1)
```

```
Epoch 1/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 44s 302ms/step - loss: 0.0785
Epoch 2/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 38s 301ms/step - loss: 0.0092
Epoch 3/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 41s 305ms/step - loss: 0.0088
Epoch 4/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 41s 303ms/step - loss: 0.0083
Epoch 5/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 37s 289ms/step - loss: 0.0070
Epoch 6/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 38s 300ms/step - loss: 0.0064
Epoch 7/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 41s 301ms/step - loss: 0.0056
Epoch 8/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 41s 301ms/step - loss: 0.0051
Epoch 9/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 41s 304ms/step - loss: 0.0047
Epoch 10/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 38s 301ms/step - loss: 0.0046
Epoch 11/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 41s 299ms/step - loss: 0.0047
Epoch 12/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 41s 302ms/step - loss: 0.0046
Epoch 13/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 41s 304ms/step - loss: 0.0042
Epoch 14/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 41s 302ms/step - loss: 0.0036
Epoch 15/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 41s 301ms/step - loss: 0.0036
Epoch 16/50
```

6.2 Test Procedure

```
from keras.layers import Dense, Dropout, LSTM
from keras.models import Sequential
```

```
model = Sequential()
model.add(LSTM(units = 50, activation = 'relu', return_sequences = True,
               input_shape = ((x.shape[1],1))))
model.add(Dropout(0.2))

model.add(LSTM(units = 60, activation='relu', return_sequences = True))
model.add(Dropout(0.3))

model.add(LSTM(units = 80, activation = 'relu', return_sequences = True))
model.add(Dropout(0.4))

model.add(LSTM(units = 120, activation = 'relu'))
model.add(Dropout(0.5))

model.add(Dense(units =1))
```

6.3 Performance Outcome

7 My learnings

The stock price prediction model developed in this project demonstrates the potential of machine learning techniques in forecasting financial market trends. By utilizing historical stock data and advanced predictive algorithms, the model achieved a satisfactory level of accuracy, highlighting the efficacy of machine learning in capturing complex market patterns. While the model provides valuable insights and can assist investors in making informed decisions, it is important to note that stock markets are influenced by a myriad of unpredictable factors. Therefore, continuous model refinement and incorporation of additional data sources, such as real-time news and macroeconomic indicators, are recommended for enhancing prediction accuracy. This project underscores the transformative impact of machine learning in financial analysis and sets the stage for further exploration and innovation in predictive analytics for the stock market.

8 Future work scope

The stock price prediction ideation includes integrating diverse data sources like social media sentiment and news, and exploring advanced models such as LSTM networks and ensemble methods to boost accuracy. Emphasizing model explainability, incorporating algorithmic trading, and enhancing risk management and portfolio optimization features will improve usability. Customizing for different market conditions, ensuring scalability for real-time predictions, and developing user-friendly interfaces are crucial. Collaborating with financial institutions for validation and commercialization, while maintaining ethical standards and regulatory compliance, will enhance the model's reliability and practicality for financial applications.