Mutual Fund Style Classification from Prospectus

MF815 Final Project
Xinyu Deng, email:xinyu618@bu.edu

1 Executive Summary

1.1 Project Objective

The primary goal of this project was to research several methods to automatically classify three main mutual fund types based on their investment strategies by using Machine Learning to analyze their text summaries.

1.2 Main Findings

My findings indicate that leveraging natural language processing (NLP) techniques, specifically custom-trained word embeddings using skip-gram models effectively captures the contextual nuances of language used in mutual fund summaries. These embeddings were instrumental in classifying funds into three main categories, Balanced Funds (Low Risk), Fixed Income Long Only (Low Risk), and Equity Long Only (Low Risk).

By comparing these custom embeddings with traditional methods such as TF-IDF, I observed that word embeddings provided a richer representation of textual data, leading to improved classification accuracies. This enhancement was primarily due to the ability of word embeddings to capture semantic relationships between words, which are often pivotal in understanding financial texts.

After using two deep-learning classifier methods, CNN(Convolutional Neural Network) and RNN (Recurrent Neural Network), to analyze three main fund types, it is concluded that both models achieve high classification accuracy. However, the RNN has a slight advantage in test accuracy, which indicates that it may handle the sequential nature of text data more efficiently than CNN.

2 Methodology

2.1 Data Preparation and Split

2.1.1 Data collection

For the project, I sourced the data from two distinct categories of documents labeled as "healthy" and "unhealthy." These labels represent the financial health status of the funds described in the articles. To make the data processing manageable and enhance the training speed, randomly sampled 1000 articles from each category.

2.1.2 Data Processing

The articles were read into pandas dataframes from their respective CSV files. Each dataframe was processed to extract the text content of the articles. For corpus creation, combined the text from both healthy and unhealthy articles to form our corpus, with a corresponding target array indicating the health of the fund (1 for healthy, 0 for unhealthy).

2.1.3 Data Splitting

The corpus was divided into training and testing datasets, with 30% of the data reserved for testing. This split ensures that have sufficient data to train our models while also retaining an independent set to evaluate their performance. The use of a fixed random state ensures that the split is reproducible, allowing for consistent results across multiple runs. I use cross-validation with grid search to optimize hyperparameters, which implies further splitting the training dataset into multiple subsets;

2.2 Build word embedding dictionary

2.2.1 Objective

The primary goal of implementing a Word2Vec skip-gram model in our project is to create a contextual vectorizer. This model allows us to translate natural language into mathematical representations that encapsulate the contextual meanings of words based on their usage in mutual fund summaries.

2.2.2 Contextual Vectorization with Skip-Gram

The skip-gram model functions by predicting surrounding words given a target word, effectively learning word associations from the local context within the text data. For our application, we define the context as a set number of words before and after the target word in each occurrence in the data.

2.2.3 Process of building a dictionary

First of all, the process begins by reading and formatting the data from mutual fund summaries. This involves using a tokenizer to clean and tokenize the text. This tokenizer removes stopwords and non-alphabetic characters, retaining meaningful words that will each receive their own vector representation.

The tokenizer is a critical component as it preprocesses the text for the skip-gram model. It ensures that only relevant words are included in the model training process, filtering out common English stopwords and any tokens without alphabetic characters. After loading, the skip-gram model is trained by setting important parameters. Then, the encoder is used to vectorize words. Therefore, after completing the above process, the word embedding dictionary has been built. Finally, the word embedding is visualized to validate the quality of embedding.

2.3 Strategy to build knowledge bases

2.3.1 Objective

The primary goal of this stage is to extract relevant sentences from mutual fund summaries that specifically address the fund's investment strategy. The styles being considered are Balanced, Fixed Income, and Equity. This extraction is based on a tailored knowledge base for each fund type, facilitating a model that predicts the fund's investment strategy from its summary text.

2.3.2 Process of building knowledge bases

To construct each knowledge base, I start with a predefined set of keywords associated with each mutual fund type:

Balanced Funds: Keywords include 'balanced', 'diversified', 'stable', 'stability', 'low', 'asset', 'conservative', 'allocation', 'mixed', 'mix', 'balance'.

Fixed Income Funds: Keywords includes 'bonds', 'fixed', 'income', 'debt', 'secure', 'securities', 'corporate', 'low', 'government', 'yield', 'interest', 'treasury’.

Equity Funds: Keywords includes 'equity', 'equities', 'share', 'shares', 'stocks', 'stock', 'markets', 'market', 'capital', 'growth'.

Using the previously trained Word2Vec model, I extend these lists by adding words that are closest in the vector space to each of the initial keywords. This process ensures that the knowledge bases are comprehensive and cover various ways in which the fund types might be described in the text.

2.4 Classification algorithm

2.4.1 Objective

The objective is to design deep learning models that can classify mutual fund summaries based on their investment styles using extracted sentences. Two models were selected for this task: a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN), both leveraging the power of pre-trained word embeddings.

2.4.2 Model Design

In this project, I utilized two main deep learning architectures, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) with LSTM units, to classify mutual fund summaries by investment style. The choice of these models was informed by their distinct capabilities and suitability for processing text data, which is central to our objective of accurately identifying investment strategies from textual summaries.

I found that CNNs, with their ability to detect patterns and features efficiently, were particularly adept at handling the spatial hierarchies within sentences. This capability made them excellent for recognizing local context and specific arrangements of words, which are crucial for identifying terms and phrases indicative of different investment styles. The CNN model's layers of convolution and pooling efficiently extracted relevant features from the text without being overly sensitive to the sentence length variations, a typical challenge in text data.

Turning to RNNs, especially those equipped with LSTM units, I noticed their superior capability in managing sequences and retaining information across lengthy texts. This was particularly beneficial for our dataset, where the nuances of investment strategies in mutual fund summaries often unfold over several sentences. The LSTM's ability to avoid the long-term dependency problem allowed it to learn from dependencies between words in long sentences, making it highly effective for understanding financial texts' intricate details and contextual nuances.

In our specific context of classifying mutual fund summaries, I leaned towards RNNs for their ability to comprehend and remember the sequence of financial terms across the entire text. This depth of understanding was crucial for accurately predicting investment styles, which often rely on a nuanced interpretation of the text rather than just keyword recognition.

3 Result

3.1 Objective

This section of the report presents the comparative analysis of two deep learning models, Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), utilized to classify textual summaries into different types of mutual funds. The evaluation focuses on each model's accuracy, loss, and Receiver Operating Characteristic (ROC) curves to determine which method more effectively classifies the specified mutual fund type.

3.2 Balanced funds

3.2.1 Model Performance Evaluation

3.2.1.1 CNN

For Accuracy and Loss, the CNN model consistently improved in accuracy over training epochs, starting from 77.21% and reaching up to 99.20%. Validation accuracy followed closely, beginning at 86.17% and peaking at 94.68%. This positive trend was mirrored in the loss metrics, where the training loss reduced significantly from 0.5319 to 0.0219, indicating the model's capability to generalize effectively from the training data. The validation loss displayed some fluctuations, highlighting areas of potential model improvement under different data scenarios. For ROC Curve Analysis, the ROC analysis yielded an AUC of 0.96, underscoring the model’s proficient classification capabilities. This metric indicates a high true positive rate relative to the false positive rate, which is crucial for accurate fund classification.

See Appendix A1, A2, and A3 for detailed figures.

3.2.1.2 RNN

For Accuracy and Loss, utilizing LSTM layers, the RNN model started with an initial accuracy of 80.97% on the training set and improved to 96.25%. The validation accuracy was impressive, achieving a high of 97.87%, which suggests the model's effectiveness in capturing and learning from the sequential nature of text. The loss metrics for both training and validation showed a downward trend, indicative of the model's efficient learning and generalization. For ROC Curve Analysis, with an AUC of 0.99, the RNN model demonstrated excellent discriminative power. This high score highlights its capability to distinguish between classes with high reliability, making it particularly suited for tasks requiring detailed contextual understanding.

See Appendix A4, A5, and A6 for detailed figures.

3.3 Fixed income funds

3.3.1 Model Performance Evaluation

3.3.1.1 CNN

In the approach with CNN, I observed that the model quickly adapted to the training data, as reflected in the training accuracy and loss. The model's training accuracy swiftly ascended to approximately 99.46%, while the corresponding training loss plummeted, signifying an efficient learning curve. However, the validation accuracy and loss fluctuated, indicating some overfitting issues where the model performed exceedingly well on the training data but less on unseen data. The CNN model's Receiver Operating Characteristic (ROC) curve achieved a remarkable Area Under the Curve (AUC) of 0.96, demonstrating the model's strong capability to distinguish between classes.

See Appendix A7, A8, and A9 for detailed figures.

3.3.1.2 RNN

On the other hand, the RNNs approach showed a more robust handling of temporal dependencies in the data. The RNN model mirrored the learning efficiency of the CNN with a training accuracy peaking at around 96.25%. However, unlike the CNN, the validation accuracy showed more stability, maintaining a tighter range of around 94.62%, indicating better generalization to new data. The model's validation loss confirmed this with a more consistent decline and less fluctuation compared to its CNN counterpart. The ROC curve for the RNN model also exhibited a compelling performance with an AUC of 0.99, affirming its strong discriminative power.

See Appendix A10, A11, and A12 for detailed figures.

3.4 Equity funds

3.4.1 Model Performance Evaluation

3.4.1.1 CNN

The CNN model consistently showed high accuracy and lower loss over ten epochs, attesting to its efficiency in capturing spatial hierarchies in data. The model's receiver operating characteristic (ROC) curve and area under the curve (AUC) score of 0.89 confirmed its robustness in distinguishing between classes.

See Appendix A13, A14, and A15 for detailed figures.

3.4.1.2 RNN

The RNN model effectively captured temporal dependencies, which is crucial given the sequential nature of text data. The accuracy improved steadily across epochs, reaching near-perfect scores on the training set. The AUC of 0.89 was comparable to the CNN, indicating that both models were equally adept at classifying Equity funds.

See Appendix A16, A17, and A18 for detailed figures.

3.5 Comparison of the results of three types of mutual funds

Across the three fund categories, both models showcased strengths that could be leveraged depending on the specific requirements of the classification task. The CNN proved to be extremely effective for quick spatial pattern recognition, which is advantageous for datasets with strong spatial correlations. On the other hand, the RNN excelled in capturing temporal dynamics, making it suitable for data where sequence plays a crucial role. The ROC curves and AUC scores for all models across different fund types further substantiated their efficacy, with scores consistently above 0.84, showcasing the models' excellent capability to discriminate between classes under various scenarios.

4 Conclusion

4.1 Implication

The analysis and application of CNN and RNN for classifying mutual fund investment strategies have yielded promising results, demonstrating the efficacy of advanced machine learning techniques in financial data interpretation and decision support. The following are some implications of this project:

- Enhanced Decision-Making: The ability of the models to classify mutual fund strategies accurately can aid investment managers and individual investors in making more informed decisions, contributing to optimized portfolio management based on the model's predictions.
- Automation and Efficiency: Automating the analysis of mutual fund characteristics and investment strategies through machine learning can significantly enhance efficiency, reducing the need for manual review and enabling quicker responses to market changes.
- Scalability: Machine learning models, such as CNNs and RNNs, can be scaled to handle vast amounts of data and various types of investment vehicles, making them invaluable tools as the volume of financial data continues to grow.
- Risk Assessment and Management: By predicting the investment strategy, stakeholders can better assess the risk profiles of different funds, leading to improved risk management and asset allocation in investment portfolios.

4.2 Limitation

Despite the advantages, there are several limitations to consider when interpreting the results and considering the application of these models in real-world scenarios:

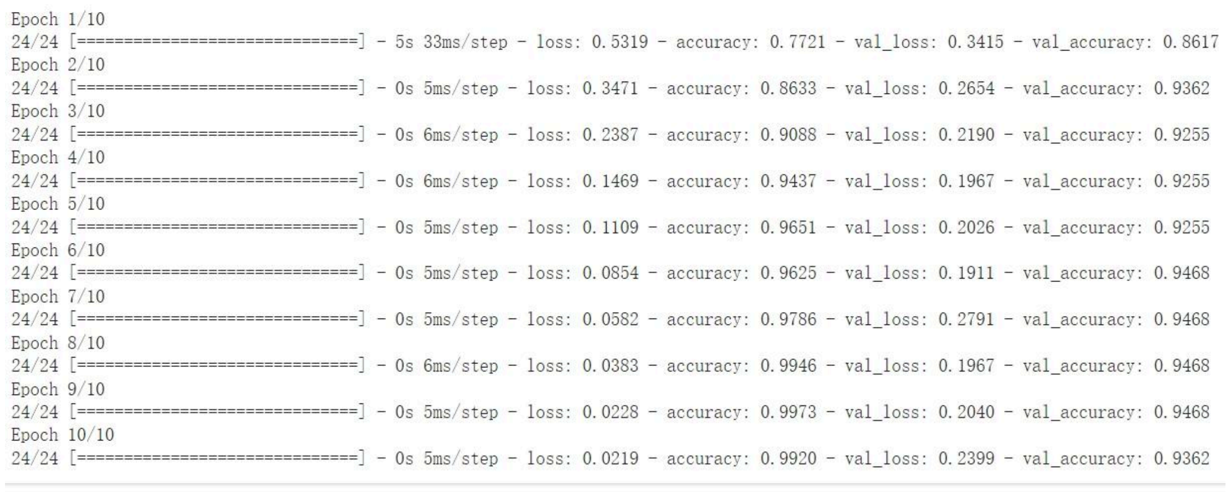
- Model Overfitting: The high accuracy of training data compared to test data suggests potential overfitting. Although steps can be taken to mitigate this (e.g., dropout layers, regularization techniques), it is a fundamental challenge in machine learning that requires careful model tuning and validation.
- Dynamic Market Conditions: Financial markets are highly dynamic, with frequent shifts influenced by a myriad of factors, from global economic changes to market sentiment. The static nature of trained models might not adapt quickly to such changes unless continually retrained with new data.

4.3 Future Work

- Expanding Fund Types: The model has been trained and tested on three primary types of mutual funds. However, the mutual fund summary offers two more of fund types, each with distinct strategies and risk profiles. Including more fund types in future models could provide a more comprehensive tool that caters to a wider range of investor needs and improves the model's utility in real-world scenarios.
- Integration of Advanced Pre-trained Models: While traditional embeddings like Word2Vec and GloVe have been used, there is potential in exploring more advanced pre-trained models such as BERT (Bidirectional Encoder Representations from Transformers). These models, which are pre-trained on vast amounts of text and fine-tuned for specific tasks, could capture deeper semantic meanings and contextual relationships within the financial data, potentially leading to higher accuracy and better generalization across different datasets.

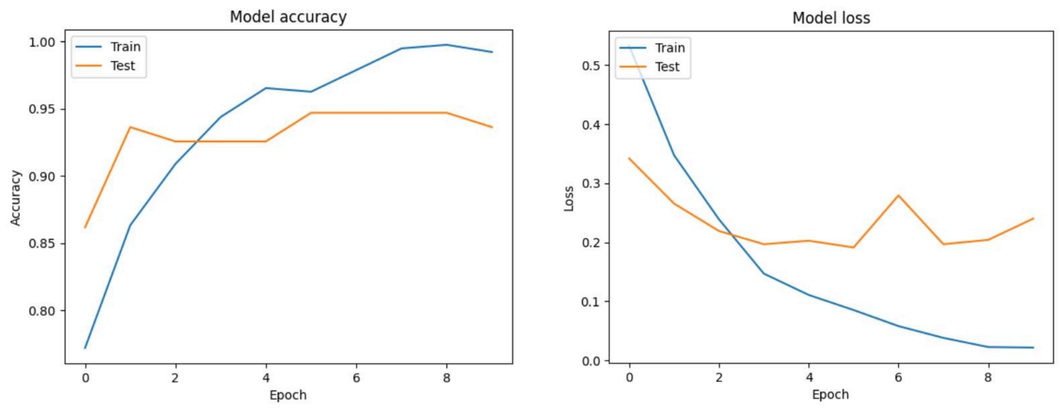
5 Appendix

Appendix A1: CNN Metrics for Balanced Fund

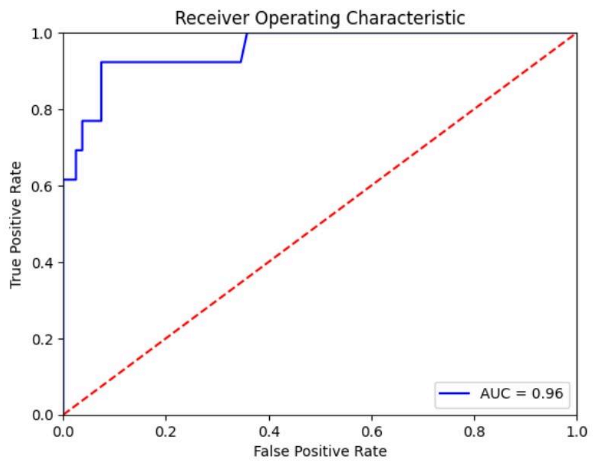


```
Epoch 1/10
24/24 [==============================] - 5s 33ms/step - loss: 0.5319 - accuracy: 0.7721 - val_loss: 0.3415 - val_accuracy: 0.8617
Epoch 2/10
24/24 [==============================] - 0s 5ms/step - loss: 0.3471 - accuracy: 0.8633 - val_loss: 0.2654 - val_accuracy: 0.9362
Epoch 3/10
24/24 [==============================] - 0s 6ms/step - loss: 0.2387 - accuracy: 0.9088 - val_loss: 0.2190 - val_accuracy: 0.9255
Epoch 4/10
24/24 [==============================] - 0s 6ms/step - loss: 0.1469 - accuracy: 0.9437 - val_loss: 0.1967 - val_accuracy: 0.9255
Epoch 5/10
24/24 [==============================] - 0s 5ms/step - loss: 0.1109 - accuracy: 0.9651 - val_loss: 0.2026 - val_accuracy: 0.9255
Epoch 6/10
24/24 [==============================] - 0s 5ms/step - loss: 0.0854 - accuracy: 0.9625 - val_loss: 0.1911 - val_accuracy: 0.9468
Epoch 7/10
24/24 [==============================] - 0s 5ms/step - loss: 0.0582 - accuracy: 0.9786 - val_loss: 0.2791 - val_accuracy: 0.9468
Epoch 8/10
24/24 [==============================] - 0s 6ms/step - loss: 0.0383 - accuracy: 0.9946 - val_loss: 0.1967 - val_accuracy: 0.9468
Epoch 9/10
24/24 [==============================] - 0s 5ms/step - loss: 0.0228 - accuracy: 0.9973 - val_loss: 0.2040 - val_accuracy: 0.9468
Epoch 10/10
24/24 [==============================] - 0s 5ms/step - loss: 0.0219 - accuracy: 0.9920 - val_loss: 0.2399 - val_accuracy: 0.9362
```

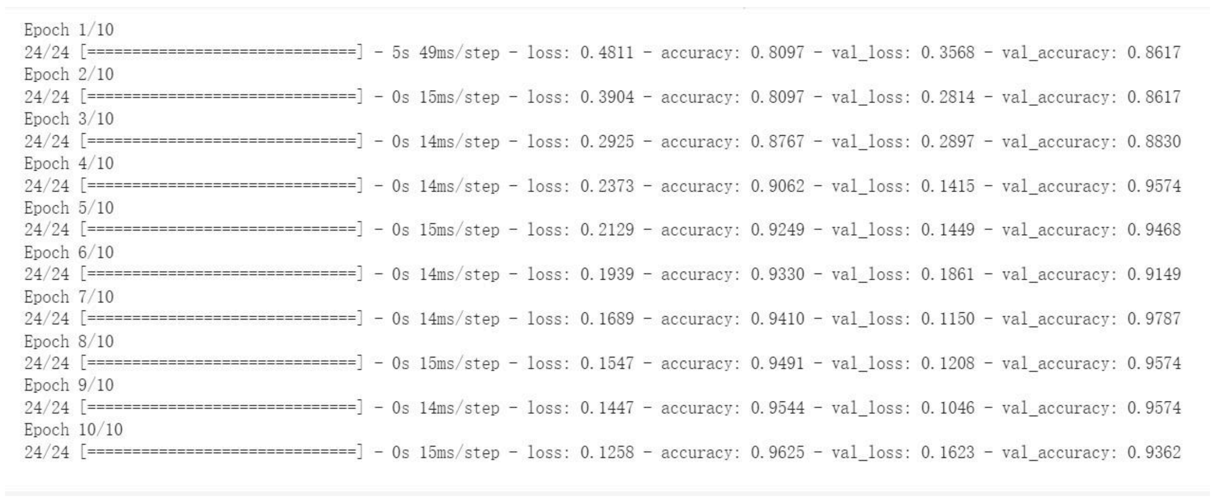
Appendix A2: CNN Model Accuracy and Loss for Balanced Fund



Appendix A3: CNN ROC Curve for Balanced Fund

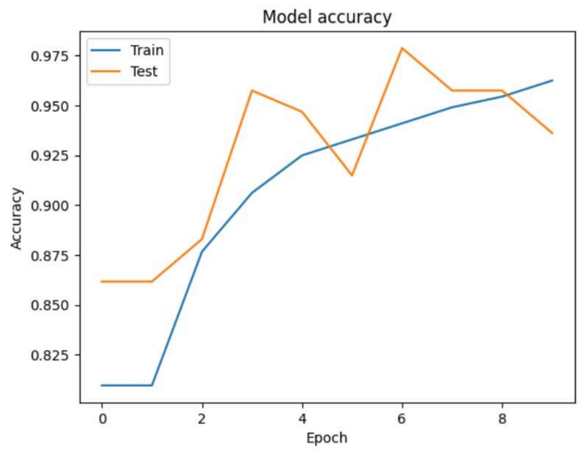


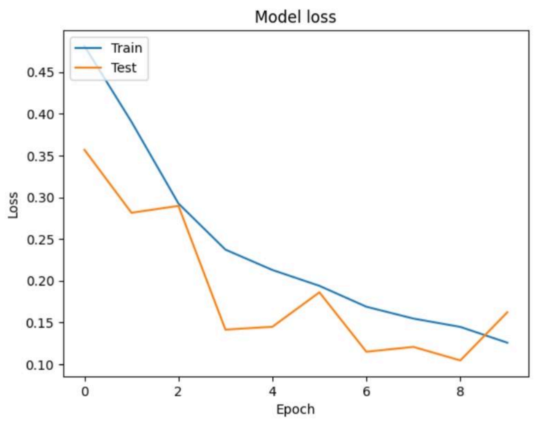
Appendix A4: RNN Metrics for Balanced Fund



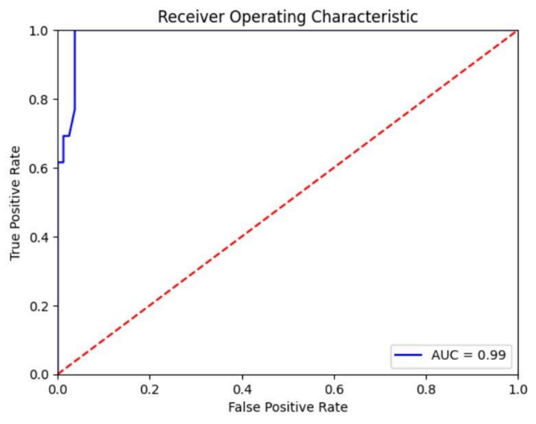
```
Epoch 1/10
24/24 [==============================] - 5s 49ms/step - loss: 0.4811 - accuracy: 0.8097 - val_loss: 0.3568 - val_accuracy: 0.8617
Epoch 2/10
24/24 [==============================] - 0s 15ms/step - loss: 0.3904 - accuracy: 0.8097 - val_loss: 0.2814 - val_accuracy: 0.8617
Epoch 3/10
24/24 [==============================] - 0s 14ms/step - loss: 0.2925 - accuracy: 0.8767 - val_loss: 0.2897 - val_accuracy: 0.8830
Epoch 4/10
24/24 [==============================] - 0s 14ms/step - loss: 0.2373 - accuracy: 0.9062 - val_loss: 0.1415 - val_accuracy: 0.9574
Epoch 5/10
24/24 [==============================] - 0s 15ms/step - loss: 0.2129 - accuracy: 0.9249 - val_loss: 0.1449 - val_accuracy: 0.9468
Epoch 6/10
24/24 [==============================] - 0s 14ms/step - loss: 0.1939 - accuracy: 0.9330 - val_loss: 0.1861 - val_accuracy: 0.9149
Epoch 7/10
24/24 [==============================] - 0s 14ms/step - loss: 0.1689 - accuracy: 0.9410 - val_loss: 0.1150 - val_accuracy: 0.9787
Epoch 8/10
24/24 [==============================] - 0s 15ms/step - loss: 0.1547 - accuracy: 0.9491 - val_loss: 0.1208 - val_accuracy: 0.9574
Epoch 9/10
24/24 [==============================] - 0s 14ms/step - loss: 0.1447 - accuracy: 0.9544 - val_loss: 0.1046 - val_accuracy: 0.9574
Epoch 10/10
24/24 [==============================] - 0s 15ms/step - loss: 0.1258 - accuracy: 0.9625 - val_loss: 0.1623 - val_accuracy: 0.9362
```

Appendix A5: RNN Model Accuracy and Loss Graphs for Balanced Fund

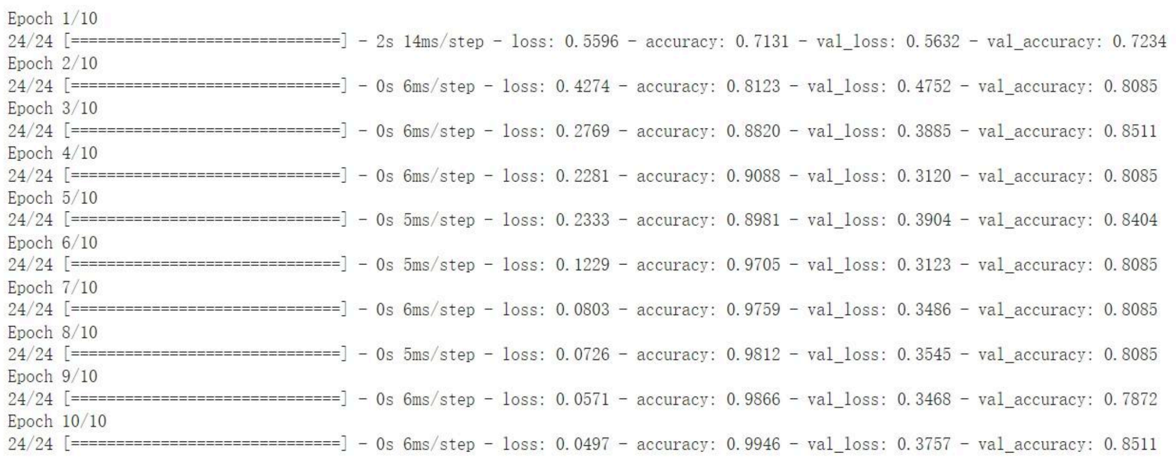




Appendix A6: RNN ROC Curve for Balanced Fund

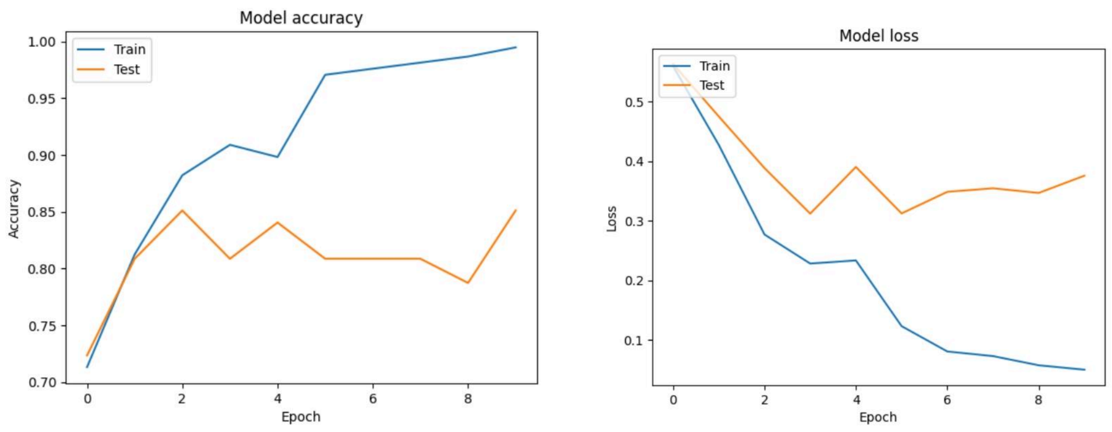


Appendix A7: CNN Model Metrics for Fixed Income Fund

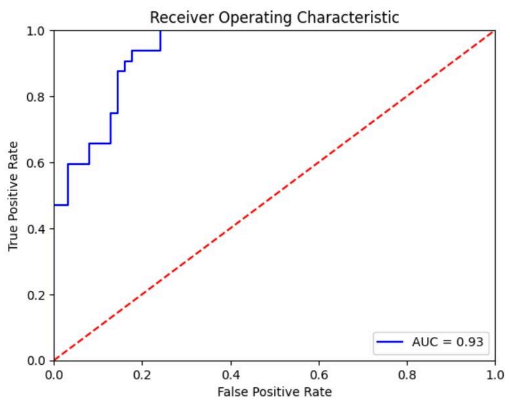


```
Epoch 1/10
24/24 [==============================] - 2s 14ms/step - loss: 0.5596 - accuracy: 0.7131 - val_loss: 0.5632 - val_accuracy: 0.7234
Epoch 2/10
24/24 [==============================] - 0s 6ms/step - loss: 0.4274 - accuracy: 0.8123 - val_loss: 0.4752 - val_accuracy: 0.8085
Epoch 3/10
24/24 [==============================] - 0s 6ms/step - loss: 0.2769 - accuracy: 0.8820 - val_loss: 0.3885 - val_accuracy: 0.8511
Epoch 4/10
24/24 [==============================] - 0s 6ms/step - loss: 0.2281 - accuracy: 0.9088 - val_loss: 0.3120 - val_accuracy: 0.8085
Epoch 5/10
24/24 [==============================] - 0s 5ms/step - loss: 0.2333 - accuracy: 0.8981 - val_loss: 0.3904 - val_accuracy: 0.8404
Epoch 6/10
24/24 [==============================] - 0s 5ms/step - loss: 0.1229 - accuracy: 0.9705 - val_loss: 0.3123 - val_accuracy: 0.8085
Epoch 7/10
24/24 [==============================] - 0s 6ms/step - loss: 0.0803 - accuracy: 0.9759 - val_loss: 0.3486 - val_accuracy: 0.8085
Epoch 8/10
24/24 [==============================] - 0s 5ms/step - loss: 0.0726 - accuracy: 0.9812 - val_loss: 0.3545 - val_accuracy: 0.8085
Epoch 9/10
24/24 [==============================] - 0s 6ms/step - loss: 0.0571 - accuracy: 0.9866 - val_loss: 0.3468 - val_accuracy: 0.7872
Epoch 10/10
24/24 [==============================] - 0s 6ms/step - loss: 0.0497 - accuracy: 0.9946 - val_loss: 0.3757 - val_accuracy: 0.8511
```

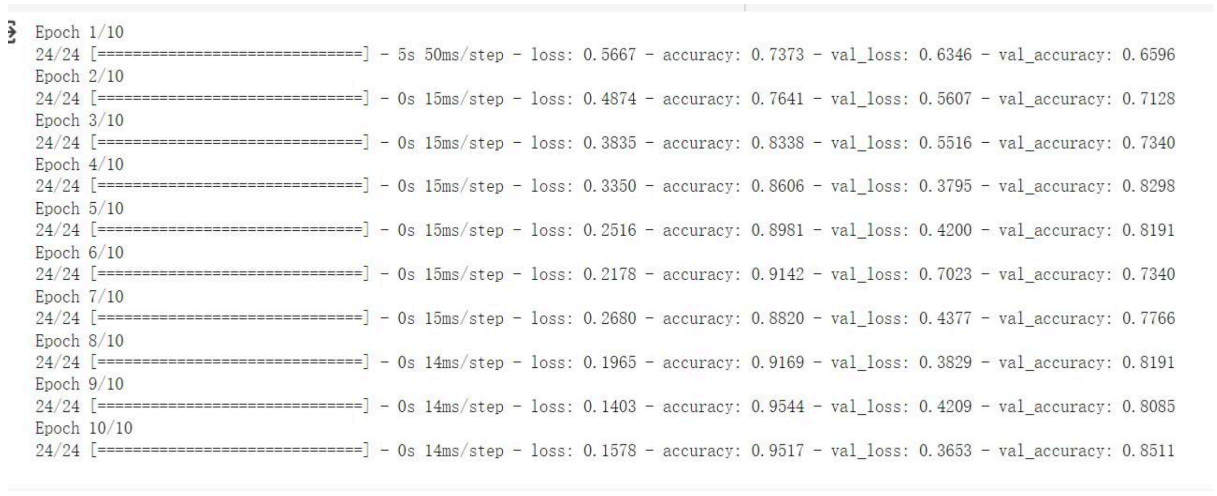
Appendix A8: CNN Model Accuracy and Loss for Fixed Income Fund



Appendix A9: CNN Model ROC Curve for Fixed Income Fund

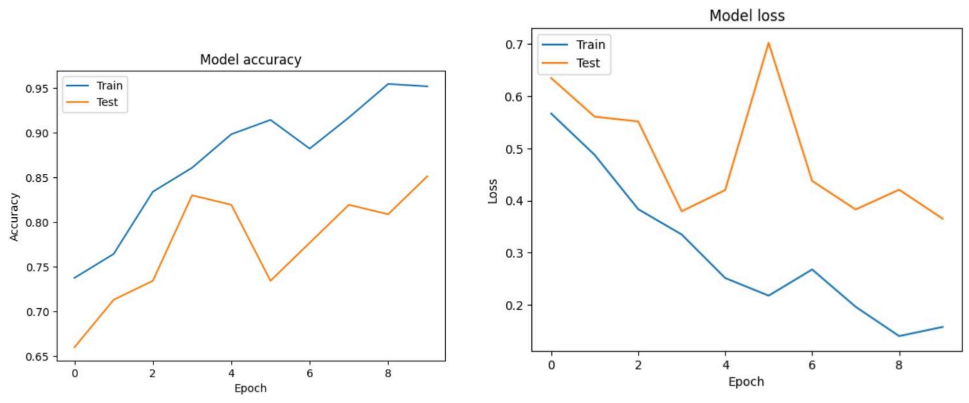


Appendix A10: RNN Model Metrics for Fixed Income Fund

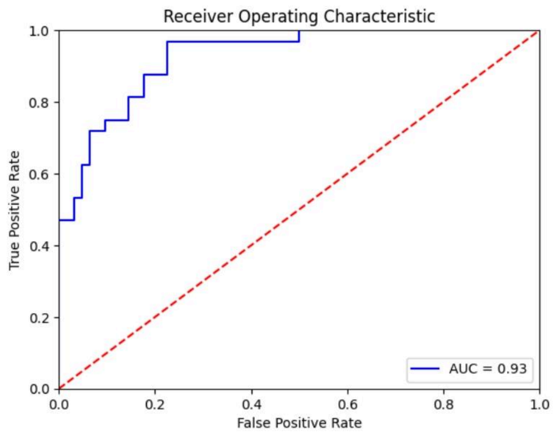


```
Epoch 1/10
24/24 [==============================] - 5s 50ms/step - loss: 0. 5667 - accuracy: 0. 7373 - val_loss: 0. 6346 - val_accuracy: 0. 6596
Epoch 2/10
24/24 [==============================] - 0s 15ms/step - loss: 0. 4874 - accuracy: 0. 7641 - val_loss: 0. 5607 - val_accuracy: 0. 7128
Epoch 3/10
24/24 [==============================] - 0s 15ms/step - loss: 0. 3835 - accuracy: 0. 8338 - val_loss: 0. 5516 - val_accuracy: 0. 7340
Epoch 4/10
24/24 [==============================] - 0s 15ms/step - loss: 0. 3350 - accuracy: 0. 8606 - val_loss: 0. 3795 - val_accuracy: 0. 8298
Epoch 5/10
24/24 [==============================] - 0s 15ms/step - loss: 0. 2516 - accuracy: 0. 8981 - val_loss: 0. 4200 - val_accuracy: 0. 8191
Epoch 6/10
24/24 [==============================] - 0s 15ms/step - loss: 0. 2178 - accuracy: 0. 9142 - val_loss: 0. 7023 - val_accuracy: 0. 7340
Epoch 7/10
24/24 [==============================] - 0s 15ms/step - loss: 0. 2680 - accuracy: 0. 8820 - val_loss: 0. 4377 - val_accuracy: 0. 7766
Epoch 8/10
24/24 [==============================] - 0s 14ms/step - loss: 0. 1965 - accuracy: 0. 9169 - val_loss: 0. 3829 - val_accuracy: 0. 8191
Epoch 9/10
24/24 [==============================] - 0s 14ms/step - loss: 0. 1403 - accuracy: 0. 9544 - val_loss: 0. 4209 - val_accuracy: 0. 8085
Epoch 10/10
24/24 [==============================] - 0s 14ms/step - loss: 0. 1578 - accuracy: 0. 9517 - val_loss: 0. 3653 - val_accuracy: 0. 8511
```

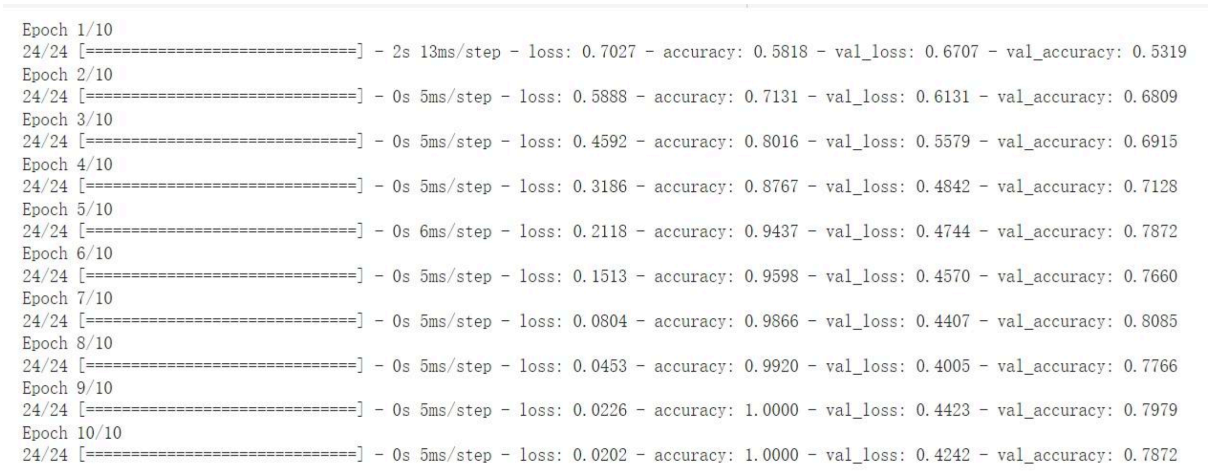
Appendix A11: RNN Model Accuracy and Loss for Fixed Income Fund



Appendix A12: RNN Model ROC Curve for Fixed Income Fund

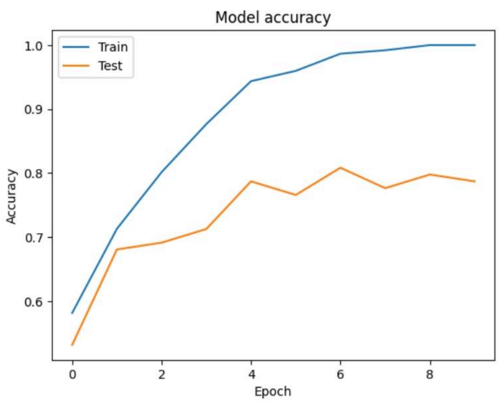


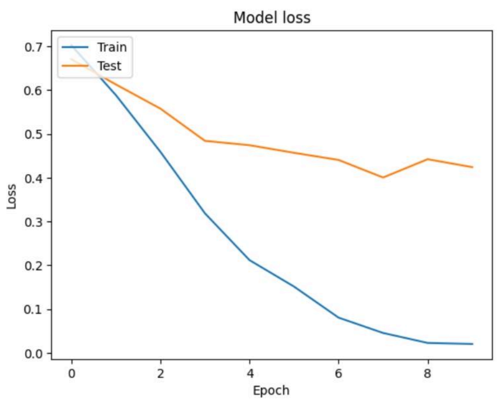
Appendix A13: CNN Model Metrics for Equity Fund



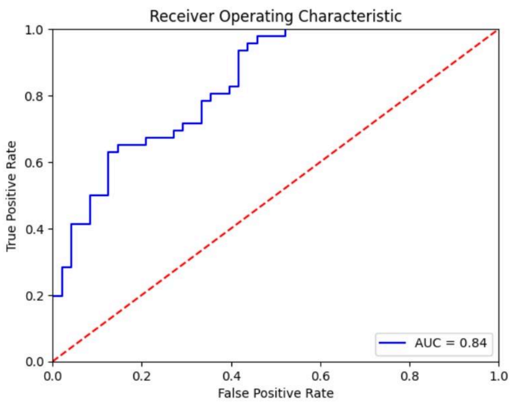
```
Epoch 1/10
24/24 [==============================] - 2s 13ms/step - loss: 0.7027 - accuracy: 0.5818 - val_loss: 0.6707 - val_accuracy: 0.5319
Epoch 2/10
24/24 [==============================] - 0s 5ms/step - loss: 0.5888 - accuracy: 0.7131 - val_loss: 0.6131 - val_accuracy: 0.6809
Epoch 3/10
24/24 [==============================] - 0s 5ms/step - loss: 0.4592 - accuracy: 0.8016 - val_loss: 0.5579 - val_accuracy: 0.6915
Epoch 4/10
24/24 [==============================] - 0s 5ms/step - loss: 0.3186 - accuracy: 0.8767 - val_loss: 0.4842 - val_accuracy: 0.7128
Epoch 5/10
24/24 [==============================] - 0s 6ms/step - loss: 0.2118 - accuracy: 0.9437 - val_loss: 0.4744 - val_accuracy: 0.7872
Epoch 6/10
24/24 [==============================] - 0s 5ms/step - loss: 0.1513 - accuracy: 0.9598 - val_loss: 0.4570 - val_accuracy: 0.7660
Epoch 7/10
24/24 [==============================] - 0s 5ms/step - loss: 0.0804 - accuracy: 0.9866 - val_loss: 0.4407 - val_accuracy: 0.8085
Epoch 8/10
24/24 [==============================] - 0s 5ms/step - loss: 0.0453 - accuracy: 0.9920 - val_loss: 0.4005 - val_accuracy: 0.7766
Epoch 9/10
24/24 [==============================] - 0s 5ms/step - loss: 0.0226 - accuracy: 1.0000 - val_loss: 0.4423 - val_accuracy: 0.7979
Epoch 10/10
24/24 [==============================] - 0s 5ms/step - loss: 0.0202 - accuracy: 1.0000 - val_loss: 0.4242 - val_accuracy: 0.7872
```

Appendix A14: CNN Model Accuracy and Loss for Equity Fund

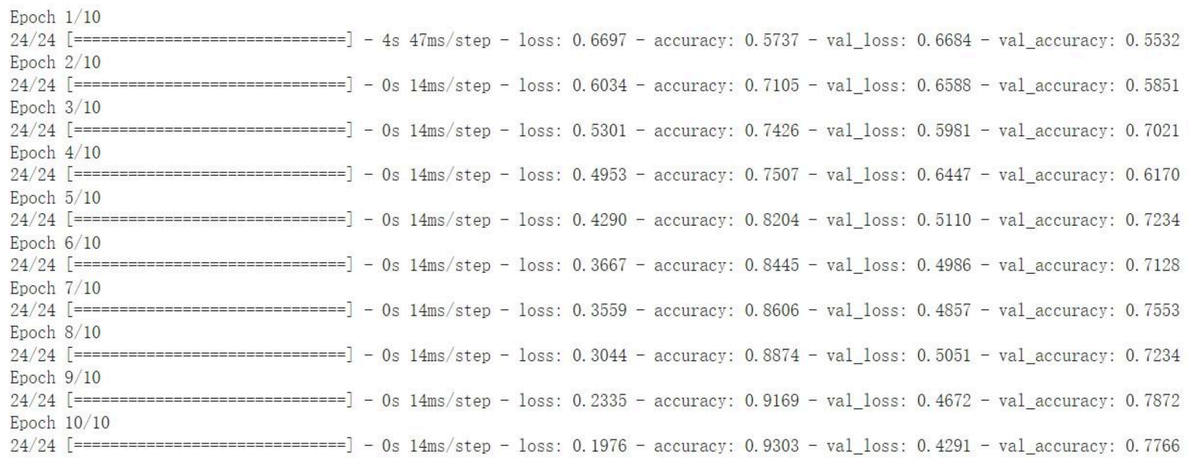




Appendix A15: CNN Model ROC Curve for Equity Fund

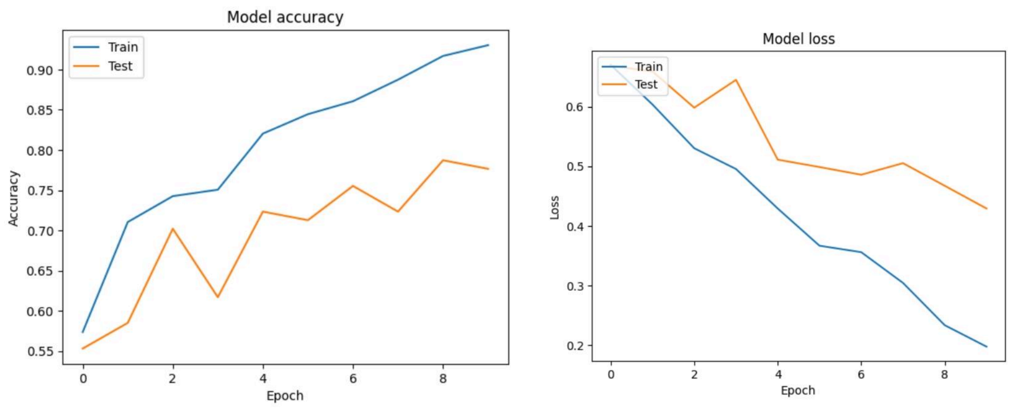


Appendix A16: RNN Model Metrics for Equity Fund

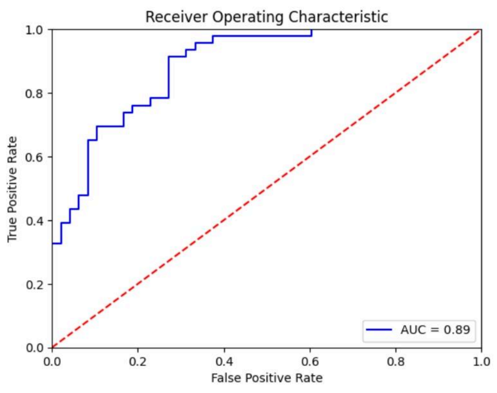


```
Epoch 1/10
24/24 [==============================] - 4s 47ms/step - loss: 0.6697 - accuracy: 0.5737 - val_loss: 0.6684 - val_accuracy: 0.5532
Epoch 2/10
24/24 [==============================] - 0s 14ms/step - loss: 0.6034 - accuracy: 0.7105 - val_loss: 0.6588 - val_accuracy: 0.5851
Epoch 3/10
24/24 [==============================] - 0s 14ms/step - loss: 0.5301 - accuracy: 0.7426 - val_loss: 0.5981 - val_accuracy: 0.7021
Epoch 4/10
24/24 [==============================] - 0s 14ms/step - loss: 0.4953 - accuracy: 0.7507 - val_loss: 0.6447 - val_accuracy: 0.6170
Epoch 5/10
24/24 [==============================] - 0s 14ms/step - loss: 0.4290 - accuracy: 0.8204 - val_loss: 0.5110 - val_accuracy: 0.7234
Epoch 6/10
24/24 [==============================] - 0s 14ms/step - loss: 0.3667 - accuracy: 0.8445 - val_loss: 0.4986 - val_accuracy: 0.7128
Epoch 7/10
24/24 [==============================] - 0s 14ms/step - loss: 0.3559 - accuracy: 0.8606 - val_loss: 0.4857 - val_accuracy: 0.7553
Epoch 8/10
24/24 [==============================] - 0s 14ms/step - loss: 0.3044 - accuracy: 0.8874 - val_loss: 0.5051 - val_accuracy: 0.7234
Epoch 9/10
24/24 [==============================] - 0s 14ms/step - loss: 0.2335 - accuracy: 0.9169 - val_loss: 0.4672 - val_accuracy: 0.7872
Epoch 10/10
24/24 [==============================] - 0s 14ms/step - loss: 0.1976 - accuracy: 0.9303 - val_loss: 0.4291 - val_accuracy: 0.7766
```

Appendix A17: RNN Model Accuracy and Loss for Equity Fund



Appendix A18: RNN Model ROC Curve for Equity Fund



6 Contribution

Although this is a team project, I completed the whole project including strategy, code, and report by myself.