Dedication

*To my beloved **parents**,whose unwavering support, love, and guidance have been the cornerstone of my journey. Your sacrifices and encouragement have fueled my determination to pursue this thesis. I am forever grateful for your unwavering belief in me.*

*To my dear **brothers** and **sister**, whose presence has brought joy, laughter, and endless inspiration to my life. Your constant support and camaraderie have been a source of strength throughout this academic endeavor.*

*To my cherished **friends**, whose friendship, encouragement, and uplifting spirits have made this journey all the more meaningful and enjoyable. Your unwavering belief in my abilities has been a driving force behind my accomplishments.*

I dedicate this thesis to each and every one of you, as your love, support, and presence have shaped me into the person I am today. Thank you for being my pillars of strength and for accompanying me on this remarkable journey.

- Oussama

Dedication

*To my loving **parents**, thank you for your endless sacrifices and for always encouraging me to pursue my dreams. Your unwavering support and unconditional love have given me the strength to overcome challenges and strive for excellence. I am truly grateful for all that you have done for me.*

*To my dear **brother**, you have been my best friend and confidant. Your constant encouragement, wisdom, and camaraderie have pushed me to reach for the stars. Thank you for always being there for me, both in times of joy and in times of difficulty.*

*To my amazing **friends**, thank you for bringing color, laughter, and joy to my life. Through thick and thin, we've created countless cherished memories. Your unwavering support and positive presence mean everything to me. I'm grateful to have you in my life.*

*To all the **teachers** who shaped my education, thank you for your dedication, knowledge, and passion. Your efforts made a profound impact on my academic journey. I'm forever grateful for your guidance and invaluable lessons.*

This report is a testament to the love, support, and belief that my family, friends, and teachers have showered upon me. Without them, my accomplishments would not have been possible. I dedicate this work to all of you, as a token of my deepest gratitude and appreciation.

- Redouane

ACKNOWLEDGEMENTS

In the name of **Allah**, the Most Gracious, the Most Merciful, to whom we owe everything, we would like to express our heartfelt gratitude.

First and foremost, we extend our sincere thanks to our supervisor, **Dr. Hadj Ahmed BOUARARA** , for his invaluable guidance, unwavering support, and trust in our abilities. His benevolent advice and encouragement have inspired us to strive for excellence and meet the high expectations placed upon us.

We would also like to express our deepest appreciation to the members of the **jury** for their time and effort in evaluating this work. Your honorable presence and valuable insights are sincerely acknowledged.

Furthermore, we are grateful to all our **teachers** throughout our years of study for imparting their knowledge and nurturing our intellectual growth.

Additionally, we would like to extend our gratitude to the collaboration team of **SIC students** who have actively contributed to this research. Their dedication, teamwork, and shared expertise have been instrumental in the realization of this work.

Finally, we extend our thanks to all those who have directly or indirectly contributed to the completion of this work. Your contributions have played a significant role in its realization.

May Allah bless and reward all of you for your kindness, support, and guidance throughout this journey.

Abstract

This research study focuses on market prediction and analysis in the Algerian market and North Africa, covering a wide range of assets including stocks, precious metals, cryptocurrencies, and oil. It introduces Nadra Stocks Prediction, a novel model that utilizes deep learning techniques such as Recurrent Neural Networks (RNN) to accurately predict stock prices for any stock. In addition, the study introduces NadraGPT, a recommendation chatbot that provides personalized investment recommendations to users. The research incorporates market sentiment analysis using advanced language models like GPT-2 and BERT to assess public sentiment towards different assets, offering a comprehensive understanding of market behavior. The study aims to provide valuable insights and predictions specifically tailored to the Algerian market and the broader North African region, supporting investors, traders, and individuals in making informed decisions based on accurate price predictions, market sentiment, and community-driven insights. This research contributes to the advancement of market prediction methodologies and empowers individuals to navigate the complexities of the Algerian market and the North African financial landscape.

Keywords: Market prediction, Recurrent Neural Networks (RNN), Deep learning, Sentiment Analysis, NadraGPT, GPT-2, BERT.

Résumés

Cette étude de recherche se concentre sur la prévision et l'analyse du marché sur le marché algérien et en Afrique du Nord, couvrant un large éventail d'actifs, notamment les actions, les métaux précieux, les crypto-monnaies et le pétrole. Il présente Nadra Stocks Prediction, un nouveau modèle qui utilise des techniques Deep Learning telles que les réseaux de neurones récurrents (RNN) pour prédire avec précision les cours des actions de n'importe quelle action. De plus, l'étude présente NadraGPT, un chatbot de recommandation qui fournit des recommandations d'investissement personnalisées aux utilisateurs. La recherche intègre une analyse du sentiment du marché à l'aide de modèles de langage avancés tels que GPT-2 et BERT pour évaluer le sentiment du public envers différents actifs, offrant une compréhension complète du comportement du marché. L'étude vise à fournir des informations et des prévisions précieuses spécifiquement adaptées au marché algérien et à la région nord-africaine au sens large, en aidant les investisseurs, les commerçants et les particuliers à prendre des décisions éclairées basées sur des prévisions de prix précises, le sentiment du marché et des informations issues de la communauté. Cette recherche contribue à l'avancement des méthodologies de prévision du marché et permet aux individus de naviguer dans les complexités du marché algérien et du paysage financier nord-africain.

Mots clés: Prédiction du marché, Réseaux de neurones récurrents (RNN), Deep Learning, Analyse des sentiments, NadraGPT, GPT-2, BERT.

P l›

šAmJ¤ T§r¶ z ˜ ”ws˜ ¨• ”wsl˜ ™yl t˜ ¤ ¥bnt˜ Yl^ Ty b˜ TF Cd˜ £@¡ z•r Tnym ˜ A`m˜ ¤ œehF± –˜Ð ¨• Am šw}± Ÿ› T`F ¤ T^wm › ¨WŒ ¤ , Ayq§r• Aynq •d ts§ d§d Ðwmž w¡¤adraœehF± ‰’w •dq§ .Xfn˜ ¤ -rfKm˜ ®m`˜ ¤ .œehF ©± œehF± CA`F T’d ¥bntl˜ -Crktm˜ TybO`˜ AkbK˜ ™ › “ym`˜ œl`t˜ Ay}w •dq§ Ty}wtl˜ T A › w ¤C ¸NadraGPTTF Cd˜ •dq , –˜Ð Y˜ T•AR³A ÐAmž • d tFA ”ws˜ A§wn`› ™yl b˜ ŸmSt§ .Ÿy›d tsml˜ TOO › T§CAm tF r•w§ Am› , Tflt m˜ šw}± £A T›A`˜ r^AKm˜ œ¥¥qt¤ GPT-2™ › T›dqt› TŒ˜ A¾AOyO TmmO› Tmy’ ¥bn ¤ «¡C œ§dq Y˜ TF Cd˜ ‘dh .”ws˜ —wls˜ ¾®›AJ A r•± ¤ CA t˜ ¤ Ÿ§rm tsm˜ œ^ ¤ , ‰F¤± Ayq§r• šAmJ TqWn›¤ ©r¶ z ˜ ”wsl˜ «¡r˜ ¤ ”ws˜ A§wn`›¤ Tqy’d˜ CA`F± ¥bn Y˜ A¾ AntF -rynts› C r’ ÐA ¨• Ÿ› r•± ŸÌkm§¤ ”ws˜A ¥bnt˜ Ay hnm ¦whn˜ ¨• b˜ @¡ œ¡As§ .Ty`mt m˜ .Ayq§r• šAmJ ¨• ¨˜Am˜ dhKm˜ ¤ T§r¶ z ˜ ”ws˜ dyq` E¤A

Ty Atf› Aml•
, r^AKm˜ ™yl , “ym`˜ œl`t˜(RN,NN) -Crktm˜ TybO`˜ AkbK˜ , ”ws˜ ‰’w
.BERT , GPT-2 , NadraGPT

Contents

Dedication II

ACKNOWLEDGEMENTS IV

Abstract V

Résumés VI

P I› VII

1 General Introduction 2
1.1 Introduction 2
1.1.1 Context 2
1.1.2 Motivation and problematic 2
1.1.3 Research Goals and Objectives 3
1.1.4 Organization of The Work 4

2 Deep Learning Concepts and Analysis 7
2.1 Deep Learning 7
2.1.1 Introduction to Deep Learning 7
2.1.2 Neural Networks and their architectures 7
2.1.3 Training Neural Networks 7
2.1.4 Hyperparameter tuning and optimization 8
2.2 Arti cial Neuron 8
2.2.1 Arti cial Neuron Model and Activation Functions 8
2.2.2 Feedforward and Backpropagation 9
2.2.3 Regularization and Dropout 10
2.2.4 Optimization Algorithms 10
2.3 Convolutional Neural Network 10
2.3.1 Introduction to CNN 10
2.3.2 Architecture and Layers of CNN 11
2.3.3 ResNet 11
2.3.4 DenseNet 12
2.3.5 GoogLeNet 13
2.3.6 Inception V3, V4 13
2.3.7 AlexNet 14

2.3.8 Image Classi cation and Object Detection using CNN 14
2.4 Recurrent Neural Network 15
2.4.1 Introduction to RNN 15
2.4.2 Types of RNN 15
2.4.3 RNN models 16
2.4.3.1 Long Short-Term Memory (LSTM) Networks 16
2.4.3.2 Gated Recurrent Units (GRUs) 17
2.4.3.3 Bidirectional RNNs 18
2.5 Transformers 18
2.5.1 Introduction to Transformers 18
2.5.2 Bidirectional Encoder Representations from Transformers (BERT) . . . 19
2.5.2.1 Model Architecture 19
2.5.2.2 Types for BERT Models 19
2.5.2.3 Summary 21
2.5.3 Generative Pre-trained Transformers (GPT) 21
2.5.3.1 Introduction to GPT 21
2.5.3.2 Types of GPT Models 22
2.5.3.3 Architecture of GPT Models 22
2.5.3.4 Summary 22
2.6 Generative Adversarial Networks (GAN) 23
2.6.1 Introduction to GAN 23
2.6.2 GAN Nets 24
2.6.2.1 Vanilla GAN 24
2.6.2.2 Deep convolutional generative adversarial networks (DCGANs) 25
2.6.2.3 Progressive GAN 25
2.6.2.4 Conditional GAN (CGAN) 25
2.6.2.5 Wasserstein GAN (WGAN) 25
2.6.2.6 CycleGAN 26
2.6.2.7 StyleGAN 26
2.6.2.8 Summary 26
2.7 Deep Dream 27
2.7.1 Introduction to Dream learning 27
2.7.2 Models of DeepDream 27
2.7.3 Architecture of DeepDream 27
2.8 Conclusion 27

3 Market Prediction Mining 29
3.1 Brief Overview of The Cryptocurrency Market 29
3.1.1 Importance of Market Prediction in Cryptocurrency Investment 29
3.1.2 Explanation of The Chapter's Focus on Market Prediction Mining . . . 30
3.2 De nition and Concept of Market Prediction Mining 31
3.2.1 Different Methods and Techniques Used in Market Prediction Mining . 31
3.2.1.1 Statistical Methods 31
3.2.1.2 Summary of Methods 31
3.2.1.3 ML Methods 33
3.2.1.4 Summary of Methods 33
3.2.2 Importance of Data Analysis and Machine Learning in Market Prediction Mining 33

3.3 Data Collection and Analysis . . . 35
3.3.1 Sources of Data for Market Prediction Mining . . . 35
3.3.2 Techniques for Data Cleaning, Processing, and Normalization . . . 36
3.3.2.1 Data Cleaning . . . 36
3.3.2.2 Data Processing . . . 36
3.3.2.3 Normalization . . . 36
3.4 Role of Deep Learning in Optimizing Cryptocurrency Mining Operations . . . 37
3.4.1 Analysis of Successful Deep Learning Models Used in Cryptocurrency Mining . . . 37
3.4.2 Challenges and Limitations of Deep Learning in Cryptocurrency Mining 37
3.5 Case Studies and Real-World Applications . . . 38
3.5.1 Analysis of Successful Market Prediction Mining Projects and Their Impact on Investment Decisions . . . 38
3.5.2 Review of challenges and limitations faced by industry players in market prediction mining . . . 39
3.6 Future Directions and Opportunities . . . 39
3.6.1 Emerging Trends and Innovations in Market Prediction Mining . . . 39
3.6.2 Implications of Market Prediction Mining for The Future of Finance . . 40
3.6.3 Potential Advancements in Deep Learning Technology and Their Impact on Cryptocurrency . . . 41
3.6.4 Possible Future Applications of Deep Learning in Cryptocurrency . . . 42
3.6.5 Implications of Deep Learning for The Future of Finance . . . 42
3.6.6 Recommendations for Future Research and Development in Deep Learning and Cryptocurrency . . . 43
3.7 Conclusion . . . 44

4 Contribution 46
4.1 AI-based Services for Price Prediction and Fraud Detection in the Marketplace (Nadra) . . . 46
4.2 Predictive Models . . . 47
4.2.1 Nadra Stocks Prediction (NSP) . . . 47
4.2.1.1 Recurrent Neural Network . . . 48
4.2.1.2 Long Short-Term Memory (LSTM) . . . 48
4.2.1.3 Gated Recurrent Unit (GRU) . . . 48
4.2.1.4 Nadra-LSTM Model(NLM) Architecture . . . 48
4.2.1.5 Nadra-GRU Model (NGM) Architecture . . . 48
4.2.1.6 Nadra-LSTM+GRU Model(NLGM) Architecture . . . 49
4.2.2 Market Sentiment Analysis . . . 50
4.2.2.1 NadraGPT . . . 50
4.2.2.2 NadraBERT . . . 52
4.2.3 NFT Generation . . . 52
4.2.3.1 Generative Adversarial Network (GAN) . . . 52
4.2.3.2 Deep Convolutional Generative Adversarial Network (DCGAN) . . . 53
4.2.3.3 GAN Model Architecture . . . 53
4.3 Nadra Recommendation System (NRS) . . . 55
4.4 Nadra Self-Learning from Community Data . . . 56
4.5 Chatbot . . . 56

4.5.1 Nadra's Chatbot . . . 56
4.6 Conclusion . . . 57

5 Result Discussion and Experimentation 59
5.1 Datasets Description . . . 59
5.1.1 Stocks Datasets . . . 59
5.1.1.1 Cryptocurrencies Dataset . . . 59
5.1.1.2 Apple,Microsoft and Tesla Datasets . . . 59
5.1.1.3 Gold Dataset . . . 60
5.1.1.4 Preprocessing Steps for Stocks and cryptocurrencies Datasets 60
5.1.2 Sentiment Analysis Dataset . . . 61
5.1.2.1 Dataset 1: Reddit Crypto Sentiment . . . 61
5.1.2.2 Dataset 2: Stocktwits-crypto . . . 62
5.1.2.3 Preprocessing Steps for Reddit Crypto Sentiment and StockTwits-Crypto Datasets . . . 62
5.1.3 NFT Dataset . . . 63
5.1.3.1 Preprocessing Steps for NFT Dataset . . . 64
5.2 Implementation tools . . . 64
5.2.1 TensorFlow . . . 64
5.2.2 Keras . . . 64
5.2.3 Scikit-learn (sklearn) . . . 64
5.2.4 Torch . . . 65
5.2.5 PIL . . . 65
5.2.6 Flask . . . 65
5.3 Programming language . . . 65
5.3.1 Python . . . 65
5.3.2 HTML (Hypertext Markup Language) . . . 66
5.3.3 CSS (Cascading Style Sheets) . . . 66
5.3.4 JavaScript . . . 66
5.3.5 SQL (Structured Query Language) . . . 66
5.4 Evaluation measures . . . 67
5.4.1 Evaluation measures for cryptocurrency price prediction . . . 67
5.4.1.1 MAE (Mean Absolute Error) . . . 67
5.4.1.2 MSE (Mean Squared Error) . . . 67
5.4.1.3 R^2 score . . . 67
5.4.2 Evaluation measures for Sentiment Analysis Classi cation . . . 68
5.4.2.1 Confusion matrix . . . 68
5.4.2.2 Precision . . . 69
5.4.2.3 Recall . . . 69
5.4.2.4 F1 score . . . 69
5.4.3 Evaluation measures for NFT Generation . . . 69
5.4.3.1 Inception Score . . . 69
5.4.3.2 Frechet Inception Distance (FID) . . . 70
5.4.3.3 Kernel Inspection Distance (KID) . . . 70
5.5 Result of Price Prediction . . . 71
5.5.1 Result of Bitcoin . . . 71
5.5.1.1 LSTM+ GRU Model . . . 71
5.5.1.2 LSTM Model . . . 72

5.5.1.3 GRU Model 74
5.5.2 Testing the Models On Other Cryptocurrencies 75
5.5.3 Testing the Models On Other Stocks 80
5.6 Result of Market Sentiment Analysis 85
5.6.1 GPT-2 85
5.6.2 BERT 86
5.7 Result of NFT Generation 88
5.7.1 Generated NFT Images 89
5.8 Conclusion 91

6 Nadra 93
6.1 Web Application 93
6.1.1 Home Page 93
6.1.2 Sign Up Page 96
6.1.3 Sign In Page 97
6.1.4 Services Page 100
6.1.5 Community of Experts 106
6.1.6 Nadra Bot VS ChatGPT 107
6.2 Mobile Application 113
6.2.1 Home Page 113
6.2.2 Sign Up Page 115
6.2.3 Sign In Page 116
6.2.4 Services Page 117
6.2.5 Community of Experts 121

7 General Conclusion 126
7.1 Challenges and Limitations 127
7.2 Future work 127

Conferences and Posters 128

List of Figures

2.1 A typical Arti cial Neural Network with two hidden layers and a single output [35] 8
2.2 Feedforward neural Networks [14] 10
2.3 CNN Architecture [21] 11
2.4 ResNet Architecture [22] 12
2.5 The architecture of DenseNet Network [23] 12
2.6 GoogLeNet incarnation of the Inception architecture [57] 13
2.7 AlexNet architecture [30] 14
2.8 CNN for Object Recognition [28] 15
2.9 Recurrent Neural Network [44] 15
2.10 Long Short-Term Memory [44] 16
2.11 Gated Recurrent Units [38] 17
2.12 Transformer model architecture [60] 19
2.13 BERT Architecture [12] 20
2.14 Summary of the different BERT models 21
2.15 Summary of the different GPT models [49] [50] [9] [45] 23
2.16 Generative Adversarial Networks. [20] 23

4.1 Nadra's System Architecture Overview 47
4.2 Overview of the Stocks Prediction Model 47
4.3 LSTM Model Architecture 49
4.4 GRU Model Architecture 49
4.5 LSTM+GRU Model Architecture 49
4.6 Overview of the Market Sentiment Analysis Model 50
4.7 NadraGPT Model Architecture 51
4.8 NadraBERT Model Architecture 52
4.9 Overview of the NFT Generation Model 53

5.1 Training and Validation Loss of LSTM+GRU Model for Bitcoin 72
5.2 Training and Validation Loss of LSTM Model for Bitcoin 73
5.3 Training and Validation Loss of GRU Model for Bitcoin 74
5.4 Comparision of GRU, LSTM And GRU+LSTM Models For Bitcoin 75
5.5 Ethereum 77
5.6 Binance 77
5.7 DogeCoin 77
5.8 LiteCoin 77

5.9 Ripple . . . 77
5.10 The Performance of GRU Model . . . 77
5.11 Ethereum . . . 78
5.12 Binance . . . 78
5.13 DogeCoin . . . 78
5.14 LiteCoin . . . 78
5.15 Ripple . . . 78
5.16 The Performance of LSTM Model . . . 78
5.17 Ethereum . . . 79
5.18 Binance . . . 79
5.19 DogeCoin . . . 79
5.20 LiteCoin . . . 79
5.21 Ripple . . . 79
5.22 The Performance of LSTM+GRU Model . . . 79
5.23 Apple . . . 81
5.24 Tesla . . . 81
5.25 Microsoft . . . 82
5.26 Gold . . . 82
5.27 Silver . . . 82
5.28 The Performance of GRU Model . . . 82
5.29 Apple . . . 83
5.30 Tesla . . . 83
5.31 Microsoft . . . 83
5.32 Gold . . . 83
5.33 Silver . . . 83
5.34 The Performance of LSTM Model . . . 83
5.35 Apple . . . 84
5.36 Tesla . . . 84
5.37 Microsoft . . . 84
5.38 Gold . . . 84
5.39 Silver . . . 84
5.40 The Performance of LSTM+GRU Model . . . 84
5.41 Model Accuracy . . . 85
5.42 Model Loss . . . 85
5.43 Model Confusion matrix . . . 86
5.44 Model Accuracy . . . 87
5.45 Model Loss . . . 87
5.46 Training Loss for Generator and Discriminator . . . 89
5.47 Real NFT . . . 90
5.48 Generated NFT . . . 90

6.1 Home Page . . . 93
6.2 Our Features Page . . . 94
6.3 About Us Page . . . 94
6.4 FAQ Page . . . 95
6.5 Sign Up Page . . . 96
6.6 Sign In Page . . . 97
6.7 Sign In Page . . . 98

6.8 Sign In Page 98
6.9 Home after Sign In 99
6.10 Market prices Page 1 100
6.11 Market prices Page 2 101
6.12 Market News Page 101
6.13 Prediction Services Page 102
6.14 Nadra Precious Metals Service 102
6.15 Nadra Oil Gas Prediction Service 103
6.16 Nadra Cryptocurrency Prediction Service 103
6.17 Fraud Detection Services Page 104
6.18 Nadra Fake News Detection Service 104
6.19 Nadra Fake NFT Detection Service 105
6.20 Community of Nadra 106
6.21 Community of Nadra 106
6.22 Nadra Bot 107
6.23 Nadra Bot 108
6.24 Nadra Bot 108
6.25 Nadra Bot 109
6.26 Chatgpt 110
6.27 Chatgpt 111
6.28 Chatgpt 112
6.29 Home Page 113
6.30 Features Page 113
6.31 Our Community 114
6.32 About Us 114
6.33 Contact Us 115
6.34 Sign Up 115
6.35 Sign In 116
6.36 Dashboard 116
6.37 Market Prices Page 117
6.38 Fraud Detection Page 118
6.39 Nadra Fake NFT Detection 119
6.40 Prediction Page 120
6.41 Community Page 121
6.42 Community Page 122
6.43 Nadra Bot 123
6.44 Nadra Bot 124
6.45 Nadra Bot 125

List of Tables

2.1 An overview of frequently used activation functions for neuron models [14] . . 9
2.2 Summary GAN models 26

3.1 Market Capitalization of top 10 cryptocurrencies 2023 [1] 30
3.2 Summary of traditional statistical and econometric techniques [29] 32
3.3 Summary of ML methods (1) [29] 33
3.4 Summary of ML methods (2) [29] 34

4.1 Con guration Parameters for GPT-2 Training 50
4.2 Con guration Parameters for BERT Training 52
4.3 Nadra Generator Architecture 54
4.4 Nadra Discriminator Architecture 55

5.1 Confusion Matrix [52] 68
5.2 Hyperparameters and Evaluation Metrics of LSTM+GRU. 71
5.3 Hyperparameters and Evaluation Metrics of LSTM. 72
5.4 Hyperparameters and Evaluation Metrics of GRU. 74
5.5 Performance of GRU, LSTM, and LSTM+GRU Models on Other Cryptocurrencies 76
5.6 Performance of GRU, LSTM, and LSTM+GRU Models on Other Stocks and Precious Metals 80
5.7 Comparison between GPT-2 and BERT 86
5.8 Con guration Parameters for DCGAN Training 89

List of Acronyms

RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
GAN	Generative Adversarial Network
NN	Neural Networks
AI	Arti cial Intelligence
ML	Machine Learning
CNN	Convolutional Neural Network
FFNN	Feedforward Neural Network
D-FFNN	Deep Feedforward Neural Network
DL	Deep Learning
BRNN	Bidirectional Recurrent Neural Networks
ReLU	Recti ed Linear Unit
BERT	Bidirectional Encoder Representations from Transformers
NLP	Natural Language Processing
NSP	Next Sentence Prediction
MLM	Masked Language Modeling
GPT	Generative Pre-trained Transformers

SOP	Sentence-Order Prediction
P2P	Peer-to-Peer
DCGAN	Deep Convolutional Generative Adversarial Network
NFT	Non-Fungible Tokens
NSP	Nadra Stocks Prediction
NLM	Nadra LSTM Model
NGM	Nadra GRU Model
NLGM	Nadra LSTM+GRU Model

Chapter 1
General Introduction

Chapter 1

General Introduction

1.1 Introduction

1.1.1 Context

Accurate price prediction and sentiment analysis are crucial in the nancial markets of Algeria and North Africa. This research aims to bridge the gap by leveraging advanced computational techniques, including deep learning algorithms and state-of-the-art language models like GPT-2 and BERT. The objective is to develop comprehensive models that enhance decision-making processes and offer valuable insights for investment strategies, encompassing cryptocurrencies, stocks, precious metals, and dropshipping in the Algerian market and the broader North African region.

1.1.2 Motivation and problematic

The nancial markets in Algeria and North Africa present unique challenges and opportunities for market participants. Accurate price prediction and trend forecasting are crucial for investors, traders, and individuals seeking to navigate these markets. Traditional nancial modeling techniques often fall short in capturing the complex dynamics and volatility observed in the Algerian market, particularly in the context of stocks, precious metals, and the nascent eld of dropshipping.

Furthermore, the absence of comprehensive market sentiment analysis speci c to the Algerian market poses additional challenges. Understanding the sentiments and opinions of market participants is vital for informed decision-making and risk management. By integrating advanced computational techniques, such as deep learning algorithms, and leveraging state-of-the-art language models like GPT-2 and BERT, we can bridge this gap and provide a holistic approach to market prediction and sentiment analysis tailored to the Algerian market and the broader North African region.

The motivation behind this research study is to develop accurate price prediction models that encompass not only cryptocurrencies but also stocks, precious metals, and dropshipping. By leveraging deep learning techniques, such as Recurrent Neural Networks (RNN), and exploring market sentiment analysis, we aim to enhance decision-making processes and empower individuals in the Algerian market and North Africa as a whole. This research endeavor aims to ll the void in comprehensive market analysis and forecasting speci c to the Algerian market, contributing to the advancement of nancial research in the region.

1.1.3 Research Goals and Objectives

The primary goal of this research is to develop and evaluate advanced computational approaches for market prediction, leveraging deep learning algorithms and sentiment analysis techniques. The speci c objectives of this study are as follows:

1. The research seeks to analyze and comprehend the distinct attributes of the cryptocurrency market, encompassing its inherent volatility, decentralized structure, and the in uence of diverse factors on price uctuations.
2. To explore and apply deep learning algorithms, such as Recurrent Neural Networks (RNN), for accurate and robust market prediction.
3. To investigate the effectiveness of sentiment analysis models, employing state-of-the-art language models like BERT and GPT-2, in capturing public sentiment related to cryptocurrencies and its in uence on market behavior.
4. To develop integrated models that combine deep learning algorithms and sentiment analysis techniques to enhance the accuracy of price predictions and provide valuable insights for investment decision-making.
5. To this end, the research aims to extract valuable insights from the developed models, enabling market participants to make informed investment decisions. By analyzing and understanding the patterns and trends identi ed by the models, individuals can make more informed choices when it comes to buying, selling, or holding cryptocurrencies, stocks and other assets .

1.1.4 Organization of The Work

The thesis is carefully organized into seven separate chapters, with each chapter focusing on a particular aspect of the research in detail. The following gives a brief summary of the topics covered in each chapter:

Chapter 1: General Introduction In the rst chapter, the research highlights the importance of accurate price prediction and sentiment analysis in the nancial markets of Algeria and North Africa. By employing advanced computational techniques, including deep learning algorithms and state-of-the-art language models like BERT and GPT-2, this study aims to address the limitations of traditional modeling approaches. The primary objectives encompass analyzing market attributes, exploring deep learning algorithms, investigating sentiment analysis models, and developing integrated models to optimize informed investment decision-making in cryptocurrencies, stocks, and other assets.

Chapter 2: Deep Learning Concepts and Analysis In this chapter, a comprehensive exploration of deep learning is presented, covering a range of important topics. It begins by introducing the fundamentals of deep learning, providing readers with a solid understanding of its key concepts, architectures, and techniques. The chapter offers valuable insights into neural networks, including CNNs, RNNs, and Transformers, as well as their applications in various domains. Additionally, it delves into the intriguing eld of Generative Adversarial Networks (GANs). With its academic approach, this chapter serves as a comprehensive guide, enabling readers to navigate through the fundamental concepts, models, and architectures in deep learning.

Chapter 3: Market Prediction Mining This chapter explores market prediction in cryptocurrency investment. It covers methods and techniques used in prediction mining, data analysis, and the role of deep learning in optimizing cryptocurrency mining. The chapter presents case studies and real-world applications, discusses emerging trends in nance, and examines the implications of deep learning for risk management and decision-making. It concludes with recommendations for future research, including integrating deep learning into cryptocurrency trading and addressing security and privacy concerns.

Chapter 4: Contribution In this chapter, we will present the contributions of this study, focusing on our system architecture and the models developed. The primary objective is to provide a comprehensive and theoretical explanation of the project, emphasizing the conceptual aspects rather than the detailed implementation in code. Speci cally, we will introduce the techniques employed in each model, including price prediction, market sentiment analysis, and NFT generation. By elucidating the framework and models, we aim to showcase our unique contributions to the eld and provide insights into the methodologies utilized in this research endeavor.

Chapter 5: Result Discussion and Experimentation The concluding chapter of this research focuses on the primary objective of conducting experiments to test, evaluate, and discuss the results of the developed predictive models. The chapter provides a detailed description of the experimentation process and analyzes the performance of these models in terms of accuracy and other relevant metrics. Special attention is given to highlighting the strengths and limitations of each model, be it in terms of their individual effectiveness or the overall approach. Additionally,

the chapter serves as a comprehensive overview of the experimental results, offering valuable insights for further research in the eld of AI-based price prediction.

Chapter 6: Nadra This chapter provides an overview of the Nadra web application and mobile application, along with an exploration of the Nadra chatbot. This chapter presents a comprehensive comparison between the Nadra chatbot and ChatGPT in terms of their predictive abilities and recommendation capabilities.

Chapter 7: General Conclusion The nal chapter provides a general conclusion to this work, highlighting the research challenges encountered throughout the study and outlining future directions for further exploration. It offers a concise summary of the main ndings and contributions, emphasizing the signi cance of the research and setting the stage for future advancements in the eld.

Chapter 2
Deep Learning Concepts and Analysis

Chapter 2

Deep Learning Concepts and Analysis

2.1 Deep Learning

2.1.1 Introduction to Deep Learning

Deep learning is a sub eld of machine learning that focuses on the development of neural networks, which are inspired by the structure and function of the human brain. Deep learning models are capable of learning from large and complex data sets, and can be used for a variety of tasks such as image recognition, natural language processing, and market prediction. [14] Deep learning models are composed of layers of arti cial neurons that are connected by weights. The inputs to the model are fed into the rst layer of neurons, which perform a linear transformation of the inputs and apply an activation function. The output of the rst layer is then fed into the next layer of neurons, and so on, until the nal output is produced.

2.1.2 Neural Networks and their architectures

Neural networks are composed of multiple layers of arti cial neurons, which can be arranged in different architectures such as feedforward networks, recurrent networks, and convolutional networks. Feedforward networks are used for tasks such as image recognition and market prediction, while recurrent networks are used for tasks such as natural language processing and time series prediction [3] [58].

2.1.3 Training Neural Networks

Even in the 2000s when processor technology was advancing, training Arti cial Neural Networks was a challenging and time-consuming operation. With multi-layer structures known as deep neural networks, the number of parameters to calculate grows along with the network's layer count, necessitating additional processing power and memory. The second AI Winter persisted until 2006, when Geoffrey Hinton, a pioneer in the area, published his ground-breaking work demonstrating that Deep Belief Networks may be used to train deep networks in a more reasonable amount of time (DBN). This new invention accelerated arti cial neural network research once more, contributing to the AI boom's expansion (Hinton et al. 2006; Hinton and Salakhutdinov 2006). Training a neural network involves adjusting the weights between the neurons in order to minimize the difference between the predicted output and the actual output. This is done by backpropagating the error through the network and adjusting the weights using an optimization algorithm such as stochastic gradient descent (SGD) [58].

Figure 2.1: A typical Arti cial Neural Network with two hidden layers and a single output [35]

2.1.4 Hyperparameter tuning and optimization

Hyperparameters are parameters that are set before training the model, such as the learning rate, batch size, and number of layers. Hyper-parameter tuning involves selecting the best values for these parameters in order to optimize the performance of the model. This can be done using techniques such as grid search, random search, and Bayesian optimization [14] [13].

2.2 Arti cial Neuron

2.2.1 Arti cial Neuron Model and Activation Functions

The rst arti cial neuron was the Threshold Logic Unit (TLU), rst proposed by Warren McCulloch and Walter Pitts in 1943. The model was speci cally targeted as a computational model of the "nerve net" in the brain. The model describes a neuron as binary processing unit. The neuron is either red and not red its output signal and is controlled by a threshold logic or activation function. The model neuron, like its biological counterpart, has number of inputs and an output. At each input to the neuron, there is a weight. The weight acts like an input synaptic resister.

There are Three Types of Neural Networks Activation Functions

Binary Step Function: Binary step function depends on a threshold value that decides whether a neuron should be activated or not.

Linear Activation Function: The linear activation function, also known as "no activation," or "identity function" (multiplied x1.0), is where the activation is proportional to the input.

Non-Linear Activation Functions: Non-linear activation functions solve the

following limitations of linear activation functions:

They allow back-propagation because now the derivative function would be related to the input, and it's possible to go back and understand which weights in the input neurons can provide a better prediction.

They allow the stacking of multiple layers of neurons as the output would now be a non- linear combination of input passed through multiple layers. Any output can be represented as a functional computation in a neural network [6].

Activation Function	Formula
Linear	$f(x) = x$
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$
Tanh	$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$
ReLU	$f(x) = max(0; x)$
Softmax	$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}}$

Table 2.1: An overview of frequently used activation functions for neuron models [14]

2.2.2 Feedforward and Backpropagation

A Feedforward Neural Network, also called a Multilayer Perceptron (MLP), can use linear or non-linear activation functions. Importantly, there are no cycles in the NN that would allow a direct feedback. Equation 2.1 de nes how the output of a MLP is obtained from the input [14].

$$f(x) = \phi^{(2)}(W^{(2)}\phi^{(1)}(W^{(1)}x + b^{(1)}) + b^{(2)}) \tag{2.1}$$

Where:
A) A shallow FFNN.
B) A Deep Feedforward Neural Network (D-FFNN) with 3 hidden layers.

Backpropagation is about understanding how to change the weights and biases in a network based on the changes of the cost function (Error). Ultimately, this means computing the partial derivatives:

$$\frac{\partial E}{\partial w_{ij}^{(l)}} \text{ and } \frac{\partial E}{\partial b_i^{(l)}} \tag{2.2}$$

But to compute those, a local variable is introduced δ_j^l, which is called the local error in the j_{th} neuron in the l_{th} layer. Based on that local error Backpropagation will give the procedure to compute:

$$\frac{\partial E}{\partial w_{ij}^{(l)}} \text{ and } \frac{\partial E}{\partial b_i^{(l)}} \tag{2.3}$$

Figure 2.2: Feedforward neural Networks [14]

how the error is de ned, and the weight is updated [6].

2.2.3 Regularization and Dropout

Regularization and dropout are techniques used to prevent over tting in neural networks. Regularization involves adding a penalty term to the loss function in order to discourage the model from learning complex patterns that may not generalize well to new data. Dropout involves randomly dropping out some of the neurons in the network during training in order to prevent them from becoming too dependent on each other [6].

2.2.4 Optimization Algorithms

Optimization algorithms are used to adjust the weights in the neural network during training in order to minimize the error. Some popular optimization algorithms used in deep learning include stochastic gradient descent (SGD), Adam, and Ada-grad. These algorithms differ in the way they update the weights and in their ability to handle different types of data and architectures [13].

2.3 Convolutional Neural Network

2.3.1 Introduction to CNN

The CNN algorithm is the most well-known and often used algorithm in the eld of deep learning (DL). As compared to its forerunners, CNN's key

advantage is its ability to automatically identify the pertinent elements without human intervention. Many various domains, including computer vision, speech processing, face recognition, etc., have made substantial use of CNNs. The structure of CNNs was inspired by neurons in human and animal brains, similar to a conventional neural network.

A Convolutional Neural Network (CNN) is a special Feedforward Neural Network utilizing convolution, ReLU and pooling layers. Standard CNNs are normally composed of several Feedforward Neural Network layers including convolution, pooling, and fully- connected layers [6].

2.3.2 Architecture and Layers of CNN

Figure 2.3: CNN Architecture [21]

2.3.3 ResNet

The purpose was to construct an ultra-deep network devoid of the vanishing gradient issue, as contrasted to the prior networks. Various variants of ResNet were developed based on the amount of layers (beginning with 34 layers and extending up to 1202 layers) (starting with 34 layers and going up to 1202 layers). The most frequent form was ResNet50, which had 49 convolutional layers plus a single FC layer. The whole number of network weights was 25.5 M, while the overall number of MACs was 3.9 M. The innovative notion of ResNet is the usage of the bypass pathway concept, as shown in Fig. 5, which was applied in Highway Nets to overcome the dif culty of training a deeper network in 2015. This is demonstrated in Fig. 5, which shows the core ResNet block diagram. This is a normal feedforward network including a residual connection [6].

Figure 2.4: ResNet Architecture [22]

2.3.4 DenseNet

To solve the problem of the vanishing gradient, DenseNet was presented, following the same direction as ResNet.

DenseNet uses cross-layer connectivity in an improved technique to handle this problem It connected each layer to all layers in the network using a feed-forward mechanism.

DenseNet exhibits the in uence of cross-layer depth wise-convolutions. Consequently, the network obtains the capacity to discriminate clearly between the added and the preserved information, since DenseNet concatenates the features of the prior layers rather than adding them. Yet, due to its narrow layer structure, DenseNet becomes parametrically high-priced in addition to the increasing amount of feature mappings [6].

Figure 2.5: The architecture of DenseNet Network [23]

2.3.5 GoogLeNet

It suggested a novel inception block (module) concept in the CNN context, since it combines multiple-scale convolutional transformations by applying merge, transform, and split functions for feature extraction. Figure 7 depicts the conception block architecture. This architecture incorporates lters of varying sizes (5×5,3×3,and1×1) to capture channel information combined with spatial information at diverse ranges of spatial resolution. The common convolutional layer of GoogLeNet is substituted by small blocks utilizing the same concept of network-in-network (NIN) architecture [94], which replaced each layer with a micro-neural network. The GoogLeNet ideas of merge, transform, and split were applied, backed by attention to an issue linked with distinct learning types of variants existing in a comparable class of multiple photos. The objective of GoogLeNet was to improve the ef ciency of CNN parameters, as well as to raise the learning capacity [6].

Figure 2.6: GoogLeNet incarnation of the Inception architecture [57]

2.3.6 Inception V3, V4

Inception v3 enhances the performance of a network with a minimal computational cost. It is favorable for applications when processing power and memory are constrained. The experimental results reveal that Inception v3 is a strong network for detecting cassava illnesses using transfer learning that attained an overall accuracy of 93%. One further version of GoogLeNet, Inception v4, replaces the lter concatenation stage. It merges residual connections with Inception architecture to speed up the training of Inception networks. The accuracy attained by this model is 98% [6].

2.3.7 AlexNet

AlexNet is considered as the first deep CNN architecture, which showed groundbreaking results for image classification and recognition tasks. AlexNet was proposed by Krizhevesky et al., which enhanced the learning capacity of the CNN by making it deeper and by applying several parameter optimizations strategies. The basic architectural design of AlexNet is shown in Fig. 8. In early 2000, hardware limitations curtailed the learning capacity of deep CNN architectures by restricting them to small size. In order to get the benefit of the representational capacity of deep CNNs, Alexnet was trained in parallel on two NVIDIA GTX 580 GPUs to overcome shortcomings of the hardware [6].

Figure 2.7: AlexNet architecture [30]

2.3.8 Image Classification and Object Detection using CNN

One of the hottest areas of research in the field of computer vision is image classification, which is also the fundamental system for other image application fields. Image classification systems are typically composed of three crucial components: image preprocessing, image feature extraction, and classifier. Object detection is the procedure of determining the instance of the class to which the object belongs and estimating the location of the object by outputting the bounding box around the object. Detecting single instance of class from image is called as single class object detection, whereas detecting the classes of all objects present in the image is known as multi class object detection [6].

Figure 2.8: CNN for Object Recognition [28]

2.4 Recurrent Neural Network

2.4.1 Introduction to RNN

RNN are specialized in the processing of sequential data such as time series or texts. at each time step, a recurrent layer receives the input that corresponds to that time step as well as its own output from the previous time step. As a consequence, the output at a given time step is a function of all the inputs from the previous time steps. Hence, the network is said to have a memory [19]. Widely used variants of RNN that provide better long-term memory include Long Short-Term Memory (LSTMs) and Gated Recurrent Units (GRU) networks.

Figure 2.9: Recurrent Neural Network [44]

2.4.2 Types of RNN

Many-to-many:

A sequence of inputs is mapped to a sequence of output. So it can be used sequence labeling, machine translation.

Many-to-one:

A sequence of inputs is mapped to a single output. An example application is sentiment analysis, where the RNN classi es the sentiment of a text.

One-to-many:

A single input is mapped to a sequence of outputs. An example application is image captioning, where the RNN generates a sequence of words to describe an image.

One-to-one:

A single input is mapped to a single output, and the RNN is used as a regular feed forward neural network.

2.4.3 RNN models

2.4.3.1 Long Short-Term Memory (LSTM) Networks

These are a type of RNN model that address the vanishing gradient problem by introducing a gating mechanism that controls the ow of information through the network [53]. LSTM networks have been shown to be effective at learning long-term dependencies.

Figure 2.10: Long Short-Term Memory [44]

The symbols and tanh represent the sigmoid function and the hyperbolic tangent function , respectively, although other activation functions are possible. Each box and tanh applies the function to the input which is an af ne combination (the weights are not indicated on the diagram).

Each + gate corresponds to the addition of vectors

Each gate corresponds to the Hadamard matrix product (term-to-term product).

The initial values are $c_0 = 0$ and $h_0 = 0$ In the following equations, the operator symbolizes the Hadamard matrix product (term-to-term product). The

following equations precisely describe the calculations shown in the diagram above:

$$
\begin{aligned}
F_t &= (W_F x_t + U_F h_t \quad 1 + b_F) \text{ (Forget gate)} \\
I_t &= (W_I x_t + U_I h_t \quad 1 + b_I) \text{ (Input gate)} \\
O_t &= (W_O x_t + U_O h_t \quad 1 + b_O) \text{ (Output gate)} \\
c_t &= F_t \quad c_t \quad 1 + I_t \quad \tanh(W_c x_t + U_c h_t \quad 1 + b_c) \\
h_t &= O_t \quad \tanh(c_t) \\
o_t &= f(W_o h_t + b_o)
\end{aligned}
\tag{2.4}
$$

Vectors F_t; I_t and O_t for respectively the gates of entry, exit and forgetting at time t are obtained in the same way: the application of the sigmoid function to a fine combinations from the input vector at time t and the vector for the hidden layer at time $t \quad 1$. These vectors then make it possible to calculate the vector c_t and the vector h_t of the hidden layer at time t.

2.4.3.2 Gated Recurrent Units (GRUs)

These are similar to LSTM networks in that they also use gating mechanisms to control the flow of information, but they are simpler and faster to compute [11].

Figure 2.11: Gated Recurrent Units [38]

The computations involved are given bellow:

$$
\begin{aligned}
Z_t &= (W_Z x_t + U_Z h_t \quad 1 + b_Z) \text{ (update gate)} \\
R_t &= (W_R x_t + U_R h_t \quad 1 + b_R) \text{ (reset gate)} \\
h_t &= Z_t \quad h_t \quad 1 + (1 \quad Z_t) \quad \tanh(W_h x_t + U_h(R_t \quad h_t \quad 1) + b_h)
\end{aligned}
\tag{2.5}
$$

2.4.3.3 Bidirectional RNNs

These are RNN models that process input data in both forward and backward directions. By considering the context both before and after each input, bidirectional RNNs are able to capture more complex dependencies in the input data.A bidirectional recurrent neural network (BRNN) was proposed by Schuster and Paliwal in 1997 [56].

2.5 Transformers

2.5.1 Introduction to Transformers

The Transformer model as described by Vaswani et al [60]. has been found to perform well for numerous applications, such as machine translation or language modeling. With the rise of pretrained language models. Transformer models have become even more popular. As a result they are at the core of several state-of-the-art natural language processing models. A Transformer model consists of numerous layers, or blocks. Each layer is a self-attention module followed by a feed-forward layer. Layer normalization and residual connections are additional components of a layer.

Figure 2.12: Transformer model architecture [60]

2.5.2 Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT) is a powerful pre-trained neural network model that has achieved state-of-the-art performance on various natural language processing (NLP) tasks. BERT is a type of transformer-based model that has been pre-trained on a large amount of text data, allowing it to learn rich, context-aware representations of text that can be ne-tuned for a wide range of downstream NLP tasks [12]. There are several types of BERT models available, each with a different architecture and pre-training objective.

2.5.2.1 Model Architecture

2.5.2.2 Types for BERT Models

BERT Base The original BERT model , has 12 transformer layers and 110 million parameters [12].

Figure 2.13: BERT Architecture [12]

BERT Large is a larger version of the model, with 24 transformer layers and 340 million parameters [12]. Both models were pre-trained on the BooksCorpus and English Wikipedia dataset using two tasks: masked language modeling (MLM) and next sentence prediction (NSP). MLM involves masking a random set of words in a sentence and training the model to predict the masked words based on the surrounding context. NSP involves predicting whether two sentences are contiguous in a document.

RoBERTa is a variation of the BERT model developed by Facebook AI researchers [33]. It uses a similar architecture as BERT Base, but with a larger training corpus and some modi cations to the pre-training procedure. RoBERTa was pre-trained on a larger corpus of text data, which includes the BooksCorpus, English Wikipedia, and CommonCrawl. The pre-training objectives were also changed to remove the NSP task and use dynamic masking, where the model is trained to predict a subset of the tokens in a sentence that are randomly masked each time the sentence is processed.

ALBERT is alite version of the BERT model developed by Google researchers [32]. It uses parameter-sharing techniques to reduce the number of parameters while maintaining or improving performance. ALBERT was pre-trained on the same BooksCorpus and English Wikipedia dataset as BERT, but with a different pre-training objective called sentence-order prediction (SOP). SOP involves randomly permuting the order of sentences in a document and training the model to predict the original order.

DistilBERT is a distilled version of the BERT model developed by Hugging Face [54]. It uses a smaller architecture and a more ef cient training proce-

dure to reduce the number of parameters and make the model more lightweight. DistilBERT was pre-trained on the same BooksCorpus and English Wikipedia dataset as BERT Base, but with a simpli ed version of the MLM task where only half of the tokens in a sentence are masked.

2.5.2.3 Summary

summary of the different BERT models, including parameters, and pre-training objectives, as well as the name of the author(s) of each paper and the year of publication.

Figure 2.14: Summary of the different BERT models

2.5.3 Generative Pre-trained Transformers (GPT)

2.5.3.1 Introduction to GPT

Generative Pre-trained Transformers (GPT) are a family of language models developed by OpenAI [49] that use deep learning to generate human-like text. These models have gained signi cant attention in recent years due to their impressive performance on various natural language processing (NLP) tasks, such as language translation, text summarization, and question-answering. In this chapter, we will discuss the different types of GPT models, their architecture, and their applications in NLP.

2.5.3.2 Types of GPT Models

OpenAI has developed several GPT models, each with different sizes and capabilities. The following are the main types of GPT models:

GPT-1 : This model was released by OpenAI in 2018 and has 117 million parameters. It was trained on a dataset of web pages and was the first of the GPT models [49].

GPT-2 : This is a larger version of GPT-1 with 1.5 billion parameters. It was trained on a much larger dataset and is capable of generating high-quality text that is often difficult to distinguish from text written by a human [50].

GPT-3 : This is the most recent and largest GPT model, with up to 175 billion parameters. It has been trained on a massive dataset that includes a wide range of texts and is capable of performing a variety of language tasks, such as translation, summarization, and question-answering [9].

GPT-4 : is a large multimodal model (accepting image and text inputs, emitting text outputs) that exhibits human-level performance on various professional and academic benchmarks [45]. OpenAI spent 6 months making GPT-4 safer and more aligned. GPT-4 is 82% less likely to respond to requests for disallowed content and 40% more likely to produce factual responses than GPT-3.5 on their internal evaluations.

2.5.3.3 Architecture of GPT Models

The architecture of each GPT model is based on the transformer architecture, which was introduced in the paper by Vaswani et al [60]. The transformer architecture uses self-attention mechanisms to compute a representation of each word in a sequence, which allows the model to consider the context of each word when generating text.

2.5.3.4 Summary

summary of the different GPT models, including the number of parameters, as well as the name of the author(s) of each paper and the year of publication.

Figure 2.15: Summary of the different GPT models [49] [50] [9] [45]

2.6 Generative Adversarial Networks (GAN)

2.6.1 Introduction to GAN

GAN are constructed of two competing neural networks, a discriminator and a generator. The role of the generator is to produce observations as close as feasible to the instances existent in the training set in order to trick the discriminator. The role of the discriminator is to predict whether a particular instance originates from the training set or has been produced by the generator [20]. The GAN is trained by alternatively training the generator and the discriminator. After a suf cient number of training repetitions the generator is competent enough to produce instances that the discriminator cannot discern from genuine instances anymore.

Figure 2.16: Generative Adversarial Networks. [20]

Objective function: Various objective functions can be used in GANs

$$\min_G \max_D V(D;G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

This expression represents value(V), which is a function of both, discriminator D and generator G. The goal is to maximize the discriminator (D) loss and minimize the generator (G) loss. Value V is the sum of expected log likelihood for real and generated data. Likelihoods (probabilities) are the discriminator outputs for real or generated images. Note that the discriminator output for a generated image is subtracted from 1 before taking the log. Maximizing the resulting values leads to optimization of the discriminator parameters such that it learns to correctly identify both real and fake data. It is also necessary to explain that:

p_{data} : Represents the distribution of real data.

p_z : Represents the distribution of noise (usually a Gaussian distribution) from which we can generate a fake image.

x and z : Represent the samples from each corresponding space.

E_x and E_z : Represent the expected log likelihood from the different outputs of both real and generated images.

D function outputs a real number ranged between 0 and 1 representing the probability for data being real(1) or fake(0). On the other hand, G function outputs a generated sample or instance Besides, in order to train generator and discriminator, Errors on their outputs are propagated back into the models. These errors are propagated as gradients of the following loss functions.

Update rule for the discriminator:

$$r_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D(x^{(i)}) + \log(1 - D(G(z^{(i)}))) \right]$$

Update rule for the generator:

$$r_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \left[\log(1 - D(G(z^{(i)}))) \right]$$

where m represents the total number of samples tested in batch before updating both models, and $_d$ and $_g$ represents the weights of each model.

2.6.2 GAN Nets

2.6.2.1 Vanilla GAN

The Vanilla GAN is the original GAN architecture proposed by Goodfellow et al. in their seminal paper [20]. The model consists of two neural networks, a generator and a discriminator, that are trained in an adversarial game. The generator takes random noise as input and generates fake samples, while the

discriminator tries to distinguish between the real and fake samples. The generator is trained to fool the discriminator, while the discriminator is trained to correctly identify the real samples.

2.6.2.2 Deep convolutional generative adversarial networks (DCGANs)

MLP de nes G and D in original GANs. Since CNNs perform better than MLPs on pictures, G and D are determined by deep convolutional neural networks (DCNNs) in DCGANs, which perform better [48]. The architecture of most modern GANs is at least somewhat based on the DCGANs design. The following are three fundamental elements of the DCGANs architecture:

First, the general architecture is mostly based on the all-convolutional net. This design lacks both pooling and "unpooling" layers. When G wants to expand the spatial dimensionality of the representation, it uses transposed convolution (deconvolution) with a stride greater than 1.

Second, use batch normalization for the majority of G and D layers. In order for the neural network to learn the right mean and scale of the data distribution, the last layer of G and the rst layer of D are not batch normalized.

Third , instead of SGD with momentum, use the Adam optimizer.

2.6.2.3 Progressive GAN

A new training mechanism for GAN is proposed in Progressive GAN (PGGAN). Progressive GAN is built on progressive neural networks. The key idea of Progressive GAN is to gradually increase both the generator and discriminator starting with a low resolution, adding new layers that model increasingly ne details as training progresses [25].

2.6.2.4 Conditional GAN (CGAN)

The Conditional GAN (CGAN) is a variant of the GAN architecture that incorporates additional information into the training process. Speci cally, the generator is conditioned on some additional input, such as class labels or attribute vectors, to generate more speci c and controlled outputs [40]. This makes the CGAN well-suited for tasks such as image-to-image translation, where the goal is to generate a target image based on a given input image.

2.6.2.5 Wasserstein GAN (WGAN)

The Wasserstein GAN (WGAN) is a variant of the GAN architecture that uses a different loss function, called the Wasserstein distance, to measure the distance between the real and fake samples. This helps to stabilize the training

process and produce more realistic images, particularly in cases where the traditional GAN loss function is ineffective [7]. The WGAN also includes several other modi cations, such as using weight clipping and a gradient penalty, to further improve the stability of the training process.

2.6.2.6 CycleGAN

The CycleGAN is a type of GAN that can be used for image-to-image translation tasks, such as converting images from one domain to another. Unlike the CGAN, the CycleGAN does not require paired training data, which makes it more exible and easier to use in practice. [64] The CycleGAN architecture includes several modi cations, such as cycle consistency loss and residual blocks, to improve the quality and realism of the generated images.

2.6.2.7 StyleGAN

The StyleGAN is a variant of the GAN architecture that includes a novel generator network that is capable of controlling the style and appearance of the generated images in a ne-grained manner. This enables the model to generate high-quality images that are not only visually appealing but also semantically meaningful [26]. The StyleGAN architecture includes several other modi cations, such as adaptive instance normalization and truncation trick, to further improve the quality and diversity of the generated images.

2.6.2.8 Summary

summary of the different GAN models.

GAN Model	Year	Reference
Vanilla GAN	2014	Goodfellow et al [20].
Conditional GAN (CGAN)	2014	Mirza and Osindero [40]
Deep Convolutional GAN (DCGAN)	2015	Radford et al [48].
Wasserstein GAN (WGAN)	2017	Arjovsky et al [7].
CycleGAN	2017	Zhu et al [64].
Progressive Growing of GANs (PGGAN)	2018	Karras et al [25].
SNGAN	2018	Miyato et al [41].
Self-Attention GAN (SAGAN)	2018	Zhang et al [62].
BigGAN	2019	Brock et al [8].
StyleGAN	2019	Karras et al [26].
GauGAN	2019	Park et al [46].
StyleGAN2	2020	Karras et al [27].

Table 2.2: Summary GAN models

2.7 Deep Dream

2.7.1 Introduction to Dream learning

DeepDream is an image synthesis technique that utilizes pre-trained convolutional neural networks to generate surrealistic and dream-like images. The DeepDream algorithm was developed by Google in 2015 [42] and has since gained popularity for its ability to reveal the features and patterns learned by neural networks through image enhancement.

2.7.2 Models of DeepDream

Pre-trained convolutional neural networks, such as the Inception and VGG networks, are commonly used as the base model for the DeepDream algorithm. These networks have been trained on large datasets of images and are capable of identifying complex features and patterns in images.

2.7.3 Architecture of DeepDream

The original DeepDream algorithm was developed using the Inception V1 neural network architecture, which consists of multiple convolutional layers and pooling layers. The network was modi ed to create the DeepDream algorithm by adding feedback loops that allow the network to enhance and amplify certain patterns in the input image.

2.8 Conclusion

In conclusion, deep learning has become a rapidly evolving eld that has transformed various areas of research and industry. With fundamental concepts such as neural networks, training, optimization, and popular architectures like CNNs, RNNs, GANs, and transformers, deep learning has revolutionized machine learning.

One of the most signi cant contributions to the eld of NLP has been made by models such as GPT and BERT. These models have achieved remarkable performance on various NLP tasks, and they are based on transformers that employ self-attention mechanisms to capture long-range dependencies in sequence data.

Chapter 3
Market Prediction Mining

Chapter 3

Market Prediction Mining

3.1 Brief Overview of The Cryptocurrency Market

Cryptocurrencies refer to virtual currencies that facilitate exchanges between individuals or groups in a network-based system. These currencies employ cryptographic algorithms to secure transactions, inheriting the decentralization, transparency, and immutability properties of blockchain technology [16]. Unlike traditional systems, cryptocurrencies operate without central authority controlling their transactions, relying instead on consensus algorithms to solve trust issues among stakeholders. The inception of Bitcoin in 2008 by an anonymous researcher named Satoshi Nakamoto marked the birth of cryptocurrency. Based on the P2P cash transfer concept, this electronic cash system aimed to facilitate online payments without the intervention of any intermediary nancial institution [43]. Since then, other cryptocurrencies have emerged, including Ethereum and others, which have become popular alternatives to Bitcoin for diverse applications.

The present tabular exposition exhibits the current market capitalization of the foremost 10 cryptocurrencies. The tabular presentation furnishes data concerning the nomenclature, prevailing market value, and market capitalization of the cryptocurrencies.

3.1.1 Importance of Market Prediction in Cryptocurrency Investment

The valuation of cryptocurrencies has become a topic of great interest to investors and researchers worldwide. The emergence of cryptocurrencies has opened up new investment opportunities, and many investors have been attracted to this digital asset. However, the volatility of cryptocurrency prices and the unpredictable nature of the market have made cryptocurrency investment a risky venture.Therefore, the importance of market prediction in cryptocurrency

Coin Name	Price	Market Cap
Bitcoin BTC	$22,522.33	$434,907,517,337.
Ethereum ETH	$1,574.69	$192,704,855,281
Tether USDT	$1.00	$71,481,385,635
BNB BNB	$288.85	$45,608,458,666
USD Coin USDC	$1.00	$43,751,617,754
XRP XRP	$0.3732	$19,014,363,256
Cardano ADA	$0.3352	$11,622,621,749
Polygon MATIC	$1.14	$9,934,863,650
Dogecoin DOGE	$0.07494	$9,930,594,952
Binance USD BUSD	$1.00	$8,692,517,193

Table 3.1: Market Capitalization of top 10 cryptocurrencies 2023 [1]

investment cannot be overstated.

As mentioned earlier, the prices of cryptocurrencies are in uenced by various factors, making their prices unstable and dif cult to predict. Investors must understand the market dynamics and the impact of these factors to make informed investment decisions. Market prediction can provide investors with valuable insights into the future movement of prices and help them determine the best time to buy or sell their digital assets.

Moreover, accurate market prediction can also help investors mitigate risks associated with cryptocurrency investment [29]. By identifying potential market trends, investors can make strategic investment decisions that align with their investment goals and risk tolerance. Additionally, market prediction can help investors avoid signi cant losses by exiting the market before a signi cant price drop.

3.1.2 Explanation of The Chapter's Focus on Market Prediction Mining

The integration of arti cial intelligence (AI) has become essential in resolving various issues encountered in the cryptocurrency market, including security, fraud detection, automated trading, market prediction, privacy, and mining. One of the initial steps towards pro table trade is price forecasting, which is crucial for both traders and market observers. The application of AI can be advantageous in achieving this step. Typically, traders analyze historical price and technical indicators to predict the future price, which becomes a daunting task given the massive amount of market data that needs to be analyzed. Furthermore, the cryptocurrency market price is in uenced by various factors, such as public sentiment, supply, demand, and news from authorities. Machine learning (ML) offers a practical solution to capture the correlation between these numerous factors and the crypto market price. ML models can identify signi cant

features and patterns in the cryptocurrency market from vast amounts of data generated daily. Additionally, they can reduce trading loss risk and speed up the trading process with minimal error. These ML models are essential in developing automated trading strategies and trading bots. Therefore, this chapter aims to examine the use of ML in the prediction of the cryptocurrency market.

3.2 De nition and Concept of Market Prediction Mining

Market prediction refers to the process of using information and analysis to make forecasts about the future performance of nancial markets, such as the stock market, currency market, or commodity market. The goal of market prediction is to identify trends, patterns, and other factors that may in uence market behavior in order to make informed investment decisions.

3.2.1 Different Methods and Techniques Used in Market Prediction Mining

3.2.1.1 Statistical Methods

The domain of cryptocurrency price prediction has witnessed the emergence of novel techniques that rely on statistical and machine learning technologies. Statistical models are commonly utilized to encode the information extracted from data using mathematical equations [29]. On the other hand, traditional approaches in this eld have relied on statistical and econometric models. The econometric methodology combines statistical and economic theories to estimate and predict the values of diverse economic variables.

3.2.1.2 Summary of Methods

Technique	Cryptocurrency	Reference	Data frequency	Time range
GARCH–MIDAS framework	Bitcoin, ethereum, litecoin, ripple, and stellar, the cryptocurrency index CRIX	Walther et al., (2019)	Daily, weekly, and monthly	3-1-2013, 7-31-2019
Kapetanios unit-root test, Maki cointegration analysis and Markov regime switching regression analysis	Bitcoin, ethereum, litecoin, and ripple	Gunay (2019)	Daily	8-7-2015, 1-13-2018
FCVAR model	Bitcoin	Dos Santos Maciel and Ballini (2019)	Daily	1-2012, 2-2018
Dynamic ordinary least square method	Bitcoin, ethereum, dash, litecoin, monero, and dash	Bhambhwani et al., (2019)	Weekly average	8-7-2015, 1-25-2019
MA method	Bitcoin	Abu Bakar et al., (2019)	Daily	10-1-2019, 12-20-2019
Markov-switching GARCH models	Bitcoin, ethereum, ripple, and litecoin	Caporale and Zekokh (2019)	Daily	4-30-2018
MSGARCH mode	Bitcoin	Ardia et al., (2019)	Daily	8-18-2011, 3-3-2018
Granger-causality method	Bitcoin, ethereum	Bartolucci et al., (2020)	—–	12-2010, 8-2017

Table 3.2: Summary of traditional statistical and econometric techniques [29]

3.2.1.3 ML Methods

The advent of big data technology and arti cial intelligence has opened up new avenues for the application of machine learning models in classi cation and prediction tasks. As a result, numerous research studies have been undertaken with a speci c focus on leveraging these innovative techniques for market prediction.

3.2.1.4 Summary of Methods

Author(s)	Paper Title	Year
T. G. Fischer, C. Krauss, A. Deinert	Statistical arbitrage in cryptocurrency markets	2019
D. Zhao, A. Rinaldo, C. Brookins	Cryptocurrency price prediction and trading strategies using support vector machines	2019
M. S. S. G. T. R. N. E. S. Rama Devi Ponnaganti, Sreeja Killamsetty	Cryptocurrency price prediction using deep learning	n.d.
O. Sattarov, H. S. Jeon, R. Oh, J. D. Lee	Forecasting bitcoin price uctuation by twitter sentiment analysis	2020
S. Raju, A. M. Tarif	Real-time prediction of bitcoin price using machine learning techniques and public sentiment analysis	2020
N. H. Huy, B. Dao, T.-T. Mai, K. Nguyen-An, et al.	Predicting cryptocurrency price movements based on social media	2019

Table 3.3: Summary of ML methods (1) [29]

3.2.2 Importance of Data Analysis and Machine Learning in Market Prediction Mining

In contemporary fast-paced and multifaceted markets, data analysis and machine learning have emerged as crucial tools for market prediction. The pro-

Author(s)	Paper Title	Year
S. Galeshchuk, O. Vasylchyshyn, A. Kryso-vatyy	Bitcoin response to twitter senti-ments	2018
K. Ceyhan, E. Kurtul-maz, O. C. Sert, T. Özyer	Bitcoin movement prediction with text mining	2018
S. McNally, J. Roche, S. Caton	Predicting the price of bitcoin using machine learning	2018
F. Fang, W. Chung, C. Ventre, M. Basios, L. Kanthan, L. Li, F. Wu	Ascertaining price formation in cryptocurrency markets with deeplearning	2020
F. Valencia, A. Gómez-Espinosa, B. Valdés-Aguirre	Price movement prediction of cryp-tocurrencies using sentiment analy-sis and machine learning	2019
D.-H. Kwon, J.-B. Kim, J.-S. Heo, C.-M. Kim, Y.-H. Han	Time series classi cation of cryp-tocurrency price trend based on a recurrent lstm neural network	2019

Table 3.4: Summary of ML methods (2) [29]

liferation of big data and the development of advanced machine learning algorithms have enabled investors and traders to analyze large and intricate datasets in real-time, resulting in more informed and nancially rewarding decisions. The ef cacy of machine learning algorithms in detecting intricate patterns and correlations, which may not be readily discernible to human analysts, has enabled investors and traders to keep abreast of the ever-evolving market dynamics. In addition, machine learning can dynamically adapt to changing market conditions and identify potential risks and opportunities, contributing to the development of more accurate and comprehensive predictions. Thus, data analysis and machine learning represent a sophisticated and accurate approach to analyz-

ing market data, facilitating well-informed and nancially rewarding decision-making for investors and traders.

3.3 Data Collection and Analysis

The process of collecting and analyzing data for cryptocurrencies involves gathering information from various sources.Once the data is collected, it can be cleaned and preprocessed to remove any errors or inconsistencies. Analyzing the data can involve using statistical methods and machine learning algorithms to identify patterns and trends in the cryptocurrency market.

3.3.1 Sources of Data for Market Prediction Mining

One crucial aspect of cryptocurrency market prediction is the availability and quality of data used in the analysis. There are various sources of data that can be used for market prediction, and the choice of data source(s) can signi cantly impact the accuracy and validity of the results.

One primary source of data for cryptocurrency market prediction is the historical price data of different cryptocurrencies. This data can be obtained from various cryptocurrency exchanges, such as Coinbase[1], Binance[2], Bit nex[3], and others. The price data typically includes information such as the opening price, closing price, high price, low price, and trading volume for a particular time interval (e.g., hourly, daily, weekly, monthly).

In addition to price data, other data sources that can be used for market prediction include social media sentiment analysis, news articles, and trading volumes. Social media sentiment analysis involves analyzing the sentiment of social media posts (e.g., Twitter, Reddit) related to a particular cryptocurrency to gauge the overall market sentiment towards the cryptocurrency. News articles can provide valuable insights into the events and developments that may impact the cryptocurrency market. Trading volumes can also provide useful information on market activity and the level of interest in a particular cryptocurrency.

Furthermore, there are various data providers and analytics platforms that specialize in providing cryptocurrency market data, such as CoinMarketCap[4], CryptoCompare[5], Kaiko[6], and BitInfoCharts[7]. These platforms offer access

[1] https://www.coinbase.com/
[2] https://www.binance.com/
[3] https://www.bitfinex.com/
[4] https://coinmarketcap.com/
[5] https://www.cryptocompare.com/
[6] https://www.kaiko.com/
[7] https://bitinfocharts.com/

to real-time and historical market data for various cryptocurrencies, including price data, trading volumes, market capitalization, and other metrics.

3.3.2 Techniques for Data Cleaning, Processing, and Normalization

Data cleaning, processing, and normalization are critical steps in preparing data for analysis. Here are some common techniques for each of these steps:

3.3.2.1 Data Cleaning

Removing Duplicates Check for and remove any duplicated data points in the dataset [4].

Handling Missing Values Decide how to handle missing data points, such as replacing them with the mean or median value or dropping them altogether [4].

Removing Outliers Identify any data points that are signi cantly different from the others and remove them if they are errors or irrelevant to the analysis [4].

Standardizing Data Formats Ensure that all data is in a consistent format, such as converting dates to a standard format or converting text data to lowercase [4].

3.3.2.2 Data Processing

Feature Engineering Create new features or variables that are relevant to the analysis, such as calculating moving averages or adding categorical variables [4].

Dimensionality Reduction Reduce the number of variables in the dataset by identifying those that are highly correlated or irrelevant to the analysis [4].

Encoding Categorical Data Convert categorical data into numerical form for easier analysis using techniques such as one-hot encoding or label encoding [4].

3.3.2.3 Normalization

Scaling data Scale numerical data to have a similar range or distribution, such as using standardization (z-score) or min-max scaling [4].

Logarithmic transformation Apply a logarithmic transformation to skewed numerical data to normalize its distribution.

Discretization Convert continuous data into discrete categories for easier analysis, such as grouping age data into age ranges [4].

The speci c techniques utilized for data cleaning, processing, and normalization are subject to the dataset in question and the objectives of the analysis. Hence, it is of utmost importance that the precise necessities of the analysis are meticulously assessed, and appropriate techniques are applied to ensure the accuracy and validity of the results.

3.4 Role of Deep Learning in Optimizing Cryptocurrency Mining Operations

3.4.1 Analysis of Successful Deep Learning Models Used in Cryptocurrency Mining

Deep learning models such as recurrent neural networks and convolutional neural networks and LSTM model have been used successfully in cryptocurrency mining for market prediction and optimization of mining operations.

The deep learning algorithms considered include two variations of Convolutional Neural Networks, three variations of Long–Short Term Memory Neural Networks and a Gated Recurrent Neural Network. These algorithms have been tested on Ethereum stock data from Poloniex [2] and their performance is estimated.

Convolutional Neural Network (CNN) has shown greatly advance on the performance of many computer vision and image classi cation tasks. CNN has been used and tested in a continuous variable predictor.

One of the earlier studies was conducted by McNally et al. (2018), who evaluated RNN and LSTM models and discovered that RNN performs best when given a shorter timestep while LSTM is better at recognizing the longer-term connections between data [65] [39].

3.4.2 Challenges and Limitations of Deep Learning in Cryptocurrency Mining

The dif culty in predicting Bitcoin, according to Pintelas et al [34]. (2020), comes from the fact that the price of cryptocurrencies follows a nearly random walk process, making it dif cult for an intelligent system to effectively recognize a pattern and predict the future values or market trend. According to Ji et al. (2019), even though some deep learning models, like LSTM and DNN, can accurately predict the direction of the bitcoin price, their performance is still

insuf cient for algorithmic bitcoin trading because, even after accounting for the time and cost involved, it is still not worth it.

The accurate prediction of cryptocurrency prices is generally considered a signi cantly complex and challenging task, mainly due to its chaotic nature. This problem is traditionally addressed by the investor's personal experience and consistent watching of exchanges prices [39].

3.5 Case Studies and Real-World Applications

3.5.1 Analysis of Successful Market Prediction Mining Projects and Their Impact on Investment Decisions

Successful market prediction mining projects such as Google Trends analysis and sentiment analysis have had a signi cant impact on investment decisions in the cryptocurrency market.

Sentiment Analysis Sentiment analysis automatically extracts sentiments from a text using text mining, natural language processing, and computational methods. It seeks to categorize a text's polarity at the phrase or class level, determining whether it expresses a positive, negative, or neutral viewpoint. Two key text sources are used in the stock market prediction task: either social media, primarily using data from Twitter, or online nancial news articles.

Twitter Sentiment Many researchers have looked into how Twitter's data relates to changes in the stock market because it is a key source of information. Despite the fact that each tweet is limited to 140 characters, it is thought that the data can accurately re ect the mood of the general audience.

Online Financial News Sentiment Articles in the nancial news are thought to be a more constant and trustworthy source of information. Many studies have revealed a high correlation between stock market uctuations and nancial news stories, as a result, understanding nancial news reports can help in forecasting stock market changes.

Machine Learning for Market Prediction Recently, many research studies used machine learning via text mining innovation methods to successfully predict the stock market changes, and the most signi cant ones are Support vector machines [5].

3.5.2 Review of challenges and limitations faced by industry players in market prediction mining

Because of the various factors involved in stock market prediction, including interest rates, politics, and economic growth, which make the stock market unpredictable and highly dif cult to anticipate effectively, it is one of the most dif cult problems to solve. A key driving force behind study in this eld is the large potential for pro t from share predictions; knowing stock movements by just a fraction of a second can result in signi cant gains. Since investing in stocks is a signi cant nancial market activity, a lack of precise knowledge and thorough information will inevitably result in investment loss. Stock market forecasting is challenging because market movements are highly uncertain [5].

3.6 Future Directions and Opportunities

3.6.1 Emerging Trends and Innovations in Market Prediction Mining

According to the standard approach of industrial economics, the size of the rm and the level of market competition are the two main factors that affect innovation.

Market prediction mining is the process of using data analysis techniques to identify patterns and trends in nancial markets. By analyzing historical data, traders and investors can gain insights into market behavior and make informed decisions about buying and selling assets. In recent years, there have been several emerging trends and innovations in market prediction mining that are changing the way traders approach the markets.

Arti cial Intelligence and Machine Learning Arti cial intelligence (AI) and machine learning (ML) are revolutionizing the way nancial data is analyzed. With the vast amount of data available, AI and ML algorithms can quickly identify patterns and trends that would be dif cult for humans to discern. These algorithms can be used to analyze large datasets, identify correlations between different markets, and make predictions about future market behavior.

Big Data Analytics Big data analytics is another emerging trend in market prediction mining. With the explosion of digital data, traders and investors can use big data analytics tools to gain insights into market behavior. These tools can analyze large amounts of data from multiple sources, including social media, news articles, and nancial reports, to identify trends and patterns that may not be visible using traditional data analysis techniques.

Blockchain Technology Blockchain technology is another innovation that is changing the way nancial markets operate. Blockchain is a distributed ledger technology that provides a secure and transparent way to record transactions. By using blockchain technology, traders and investors can reduce transaction costs, eliminate intermediaries, and increase transparency in the markets.

Sentiment Analysis Sentiment analysis is the process of using natural language processing (NLP) and machine learning techniques to analyze text data and identify the sentiment behind it. Traders and investors can use sentiment analysis to gain insights into market sentiment, which can be used to make informed decisions about buying and selling assets.

Predictive Analytics Predictive analytics is the process of using statistical algorithms and machine learning techniques to analyze historical data and make predictions about future market behavior. By analyzing patterns and trends in historical data, traders and investors can make informed decisions about buying and selling assets [5] [36].

3.6.2 Implications of Market Prediction Mining for The Future of Finance

Market prediction mining, which involves using data analysis techniques to identify patterns and trends in nancial markets, has signi cant implications for the future of nance. Here are some of the implications:

Improved Decision-making Market prediction mining provides investors with valuable insights that can inform their investment decisions. By analyzing historical data, traders and investors can identify patterns and trends in market behavior, which can help them make more informed investment decisions. This can lead to improved returns and reduced risks.

Increased Automation With the help of arti cial intelligence and machine learning algorithms, market prediction mining can be automated. This means that investors can rely on machines to analyze large amounts of data and make predictions about market behavior. As automation becomes more prevalent in nance, it will likely reduce the need for human analysts and traders.

Democratization of Finance Market prediction mining has the potential to make nance more accessible to a wider range of people. By providing investors with valuable insights, it can help level the playing eld between institutional

investors and retail investors. As technology continues to advance, it will likely become easier and cheaper for individuals to access nancial data and insights.

Greater Transparency Market prediction mining can provide greater transparency in nancial markets. By analyzing data from multiple sources, including social media, news articles, and nancial reports, it can help investors better understand the factors that in uence market behavior. This can lead to more informed investment decisions and greater trust in nancial markets.

New Investment Opportunities Market prediction mining can help identify new investment opportunities. By analyzing market trends and patterns, it can help investors identify emerging markets and assets that may have strong growth potential. This can lead to the development of new investment products and strategies [34].

3.6.3 Potential Advancements in Deep Learning Technology and Their Impact on Cryptocurrency

Improved accuracy in price prediction: Deep learning models are already being used to predict cryptocurrency prices, but as the technology advances, we can expect even greater accuracy in these predictions. This could be particularly valuable for investors looking to make informed decisions about when to buy and sell.

Increased Ef ciency in Mining Mining cryptocurrency requires a signi cant amount of computing power, and deep learning models could help to optimize the mining process, making it more ef cient and cost-effective. Enhanced security: As cryptocurrency becomes more mainstream, security threats are becoming more prevalent. Deep learning algorithms can help to identify potential threats and prevent security breaches, protecting user data and assets.

Integration with Blockchain Technology Deep learning algorithms can be integrated with blockchain technology to create more sophisticated smart contracts and other applications. This could help to streamline transactions and make the cryptocurrency market more ef cient.

Improved User Experience Deep learning algorithms can be used to analyze user behavior and make recommendations based on individual preferences. This could help to improve the user experience of cryptocurrency platforms and encourage more widespread adoption [39] [5].

3.6.4 Possible Future Applications of Deep Learning in Cryptocurrency

Deep learning, has demonstrated impressive capabilities in a wide range of elds, from computer vision to natural language processing. The combination of the two has the potential to bring about even more transformative changes to the world of cryptocurrency.

Price Prediction Cryptocurrencies are notoriously volatile, with prices changing rapidly and often unpredictably. Deep learning models can analyze historical data, market trends, news, and social media sentiment to make accurate predictions about future price movements, which can be valuable for investors.

Sentiment Analysis Deep learning algorithms can be trained to analyze social media conversations and news articles to identify trends in public sentiment toward cryptocurrencies. This information can help investors make informed decisions about when to buy or sell, based on the prevailing market sentiment [34] [5].

3.6.5 Implications of Deep Learning for The Future of Finance

The implications of deep learning for the future of nance are vast and far-reaching. As the technology continues to evolve and improve, we can expect to see more innovative applications that could transform the nance industry and improve outcomes for customers and investors alike.
Some of implications of deep learning:

Enhanced Risk Management Deep learning algorithms can analyze vast amounts of data to identify patterns and predict potential risks. This could help nancial institutions to improve their risk management practices and reduce the likelihood of nancial losses.

More Accurate Decision-making Deep learning algorithms can analyze complex data sets to provide insights that would be dif cult or impossible for humans to identify. This could help nancial institutions to make more informed decisions, leading to better outcomes for investors and customers.

More Ef cient Trading Deep learning algorithms can be used to analyze market trends and make predictions about future price movements. This could help traders to make more informed decisions and execute trades more ef ciently.

Personalized Financial Services Deep learning algorithms can analyze customer data to provide personalized nancial advice and services. This could help to improve customer satisfaction and loyalty, as well as increase revenue for nancial institutions.

Automation of Financial Tasks Deep learning algorithms can be used to automate tasks such as loan underwriting and credit scoring. This could help to streamline nancial processes and reduce the need for manual labor.

3.6.6 Recommendations for Future Research and Development in Deep Learning and Cryptocurrency

Integration of Deep Learning into Cryptocurrency Trading Researchers can investigate how deep learning can be integrated into cryptocurrency trading. This can include developing advanced algorithms that can accurately predict cryptocurrency prices and trends.

Security and Privacy in Cryptocurrency Transactions As cryptocurrency transactions become more popular, there is a need to ensure the security and privacy of these transactions. Future research can explore how deep learning can be used to improve the security and privacy of cryptocurrency transactions.

Smart Contract Optimization Smart contracts are a critical part of the cryptocurrency ecosystem, but they can be complex and dif cult to optimize. Researchers can explore how deep learning can be used to optimize smart contracts and make them more ef cient.

Decentralized Deep Learning Deep learning algorithms often require large amounts of data, which can be dif cult to gather and store. Decentralized deep learning can be used to train algorithms on data that is distributed across many nodes, making the process more ef cient and secure.

Cryptocurrency Fraud Detection Cryptocurrency fraud is a signi cant problem, and deep learning can be used to detect and prevent fraud. Future research can investigate how deep learning algorithms can be used to detect fraudulent transactions and identify potential scammers.

Understanding Cryptocurrency Market Dynamics The cryptocurrency market is highly volatile and can be dif cult to predict. Deep learning algorithms can be used to analyze market trends and predict future price movements. Future research can

explore how deep learning can be used to better understand the dynamics of the cryptocurrency market.

3.7 Conclusion

In conclusion, the cryptocurrency market has become a fascinating area of research and investment due to its decentralized nature and the potential for signi cant returns. However, due to the volatility and unpredictability of the market, it is crucial to employ effective market prediction mining techniques to make informed investment decisions. Machine learning algorithms and data analysis techniques have proven to be valuable tools in this regard, as they allow investors to analyze vast amounts of data and identify patterns and trends that can in uence market behavior. While deep learning models have shown promise in optimizing cryptocurrency mining operations and predicting market trends, challenges and limitations still exist. Further research and development in this area will undoubtedly lead to more accurate and reliable market prediction mining techniques, ultimately bene ting both investors and the cryptocurrency market as a whole.

Chapter 4
Contribution

Chapter 4

Contribution

4.1 AI-based Services for Price Prediction and Fraud Detection in the Marketplace (Nadra)

The development of AI-based services for detecting fraud and predicting trends in the marketplace has become increasingly important in today's business landscape. With the rise of big data and the increasing complexity of financial transactions, traditional methods of fraud detection and trend analysis have become inadequate. AI-based solutions offer the promise of more accurate and efficient fraud detection, as well as the ability to predict trends and make data-driven decisions in real-time. These services leverage deep learning algorithms to analyze large amounts of data, identify patterns and anomalies, and provide insights that can help businesses stay ahead of the competition. The use of AI-based services for fraud detection and trend analysis is expected to continue to grow in the coming years, as businesses seek to stay competitive in an ever-changing marketplace.

The objective of this project is to develop a comprehensive platform for detecting fraudulent activities and predicting market trends. This platform aims to provide investors and e-commerce users with valuable insights to make informed decisions and minimize financial loss. The thesis will primarily focus on price prediction for cryptocurrencies and other stocks, market sentiment analysis, and the exploration of non-fungible tokens (NFTs). Through collaboration with SIC students, the project aims to combat various fraudulent activities, including the identification of fake news, counterfeit gold, fake reviews, and fraudulent NFTs.

Figure 4.1: Nadra's System Architecture Overview

4.2 Predictive Models

4.2.1 Nadra Stocks Prediction (NSP)

In this approach, the utilization of historical stocks data as input for the prediction model enables the generation of accurate forecasts and valuable insights into future market trends. By analyzing the historical data, the prediction model identi es patterns, trends, and relationships that play a crucial role in informing the decision-making process.

Figure 4.2: Overview of the Stocks Prediction Model

4.2.1.1 Recurrent Neural Network

RNN stands for Recurrent Neural Network, a type of neural network designed for sequential data processing. It uses a feedback loop to send the output of a previous time step as an input to the current time step. This enables the network to maintain a sort of memory or context of past inputs, which can be useful for tasks like time series prediction.

4.2.1.2 Long Short-Term Memory (LSTM)

LSTM stands for Long Short-Term Memory, a type of RNN architecture. It was designed to address the issue of vanishing gradients in traditional RNNs, which can make it dif cult for the network to learn long-term dependencies. LSTM includes a memory cell that can store and access information over long periods of time, as well as ”gates” that control the ow of information in and out of the cell. This makes it well-suited for tasks that require modeling long-term dependencies.

4.2.1.3 Gated Recurrent Unit (GRU)

GRU stands for Gated Recurrent Unit, another type of RNN architecture. Like LSTM, it was designed to address the vanishing gradient problem in traditional RNNs. GRU has fewer parameters than LSTM, making it faster to train and less likely to over t the data. It uses a similar gating mechanism as LSTM to control the ow of information, but with fewer gates. GRU is often used for similar tasks as LSTM.

4.2.1.4 Nadra-LSTM Model(NLM) Architecture

The proposed architecture utilizes three LSTM layers with increasing numbers of units (64, 128, and 256) to capture intricate patterns in the historical price data. Dropout layers are inserted between each LSTM layer to enhance regularization and prevent over tting. The nal layer in the architecture is a dense layer with a single unit, responsible for producing the predicted price as the output.

4.2.1.5 Nadra-GRU Model (NGM) Architecture

This architecture bears similarities to the previous one, with the main difference being the utilization of GRU layers instead of LSTM layers.

4.2.1.6 Nadra-LSTM+GRU Model(NLGM) Architecture

This architecture incorporates a hybrid approach by combining GRU and LSTM layers.

Figure 4.3: LSTM Model Architecture

Figure 4.4: GRU Model Architecture

Figure 4.5: LSTM+GRU Model Architecture

4.2.2 Market Sentiment Analysis

Real-time social media data scraping involves the automated and continuous collection of data from diverse social media platforms. This encompasses a wide range of content, including posts, comments, tweets, and other user-generated information. The extracted data is then utilized as input for a sophisticated sentiment analysis model, which leverages advanced natural language processing and machine learning techniques. By effectively classifying the sentiment expressed in the collected text, the model assigns labels such as positive, negative, or neutral to each piece of social media data. This enables valuable insights into public opinion and sentiment trends, thereby empowering decision-makers to make informed choices that are attuned to the prevailing sentiment on social media platforms.

Figure 4.6: Overview of the Market Sentiment Analysis Model

4.2.2.1 NadraGPT

The training of the GPT-2 model in this study involved optimizing its performance through the selection of speci c hyperparameters. To maintain consistent input length, text sequences were modi ed to have a maximum of 60 tokens by padding and truncating them. The training process spanned three epochs and utilized a batch size of 32. The AdamW optimizer was employed with a learning rate of 2e-5 and a weight decay coef cient of 0.04. The optimizer's betas were set to 0.92 and 0.999.

Optimizer	Lr	Batch Size	Epochs	Max Length	Weight Decay	Betas
AdamW	2e-5	32	3	60	0.04	(0.92, 0.999)

Table 4.1: Con guration Parameters for GPT-2 Training

Figure 4.7: NadraGPT Model Architecture

4.2.2.2 NadraBERT

The training procedure of the BERT model in this research aimed at enhancing its performance through the careful selection of hyperparameters. To ensure uniformity in input length, text sequences were adjusted to a maximum of 160 tokens using padding and truncation techniques. The training phase encompassed three epochs, employing a batch size of 64. The AdamW optimizer was employed, with a learning rate of 1e-5 and a weight decay coef cient of 0.04.

Figure 4.8: NadraBERT Model Architecture

Optimizer	Lr	Batch Size	Epochs	Max Length	Weight Decay	Betas
AdamW	1e-5	64	3	160	0.04	(0.9, 0.999)

Table 4.2: Con guration Parameters for BERT Training

4.2.3 NFT Generation

4.2.3.1 Generative Adversarial Network (GAN)

GAN, short for Generative Adversarial Network, is a powerful deep learning model that excels in generating synthetic data samples. It comprises two main components: a generator and a discriminator, working in opposition to each other. The generator generates new data instances, such as images, while the discriminator tries to distinguish between real and generated data.

Figure 4.9: Overview of the NFT Generation Model

4.2.3.2 Deep Convolutional Generative Adversarial Network (DCGAN)

Deep Convolutional Generative Adversarial Network, commonly known as DCGAN, is a specialized variant of the Generative Adversarial Network (GAN) architecture that focuses on generating realistic images. It leverages deep convolutional neural networks (CNNs) to enhance the generator and discriminator networks' capabilities.

4.2.3.3 GAN Model Architecture

Nadra Generator Architecture The generator architecture consists of a series of transposed convolutional layers, batch normalization layers, and leaky ReLU activation functions. It takes a latent vector of size 100 and gradually upsamples it to generate an output image of size 64x64x3.

• The first layer is a transposed convolutional layer with 100 input channels and 512 output channels. It uses a kernel size of 4x4 and stride 1. The resulting feature maps have dimensions of 4x4x512. Batch normalization is applied to normalize the activations, and a leaky ReLU activation function with a negative slope of 0.2 introduces non-linearity.

• The second layer is another transposed convolutional layer that takes the 512 input channels and produces 256 output channels. It uses a kernel size of 4x4, a stride of 2, and padding of 1. The output feature maps have dimensions of 8x8x256. Batch normalization and leaky ReLU activation are applied again.

• The third layer is a transposed convolutional layer with 256 input channels and 128 output channels. It uses a kernel size of 4x4, a stride of 2, and padding of 1. The resulting feature maps have dimensions of 16x16x128. Batch normalization and leaky ReLU activation follow.

• The fourth layer is a transposed convolutional layer that takes the 128 input

channels and produces 64 output channels. It uses a kernel size of 4x4, a stride of 2, and padding of 1. The output feature maps have dimensions of 32x32x64. Batch normalization and leaky ReLU activation are applied.

• Finally, the fth and last layer is a transposed convolutional layer with 64 input channels and 3 output channels (representing the RGB image). It uses a kernel size of 4x4, a stride of 2, and padding of 1. The resulting feature maps have dimensions of 64x64x3. The Tanh activation function is used to ensure the output values are in the range [-1, 1], representing the pixel intensities of the generated image.

Layer (type)	Output Shape	Param #
ConvTranspose2d-1	[-1, 512, 4, 4]	819,200
BatchNorm2d-2	[-1, 512, 4, 4]	1,024
LeakyReLU-3	[-1, 512, 4, 4]	0
ConvTranspose2d-4	[-1, 256, 8, 8]	2,097,152
BatchNorm2d-5	[-1, 256, 8, 8]	512
LeakyReLU-6	[-1, 256, 8, 8]	0
ConvTranspose2d-7	[-1, 128, 16, 16]	524,288
BatchNorm2d-8	[-1, 128, 16, 16]	256
LeakyReLU-9	[-1, 128, 16, 16]	0
ConvTranspose2d-10	[-1, 64, 32, 32]	131,072
BatchNorm2d-11	[-1, 64, 32, 32]	128
LeakyReLU-12	[-1, 64, 32, 32]	0
ConvTranspose2d-13	[-1, 3, 64, 64]	3,072
Tanh-14	[-1, 3, 64, 64]	0

Table 4.3: Nadra Generator Architecture

Nadra Discriminator Architecture The discriminator architecture consists of a series of convolutional layers followed by batch normalization and LeakyReLU activation functions.

• The input image, which is an RGB image with a depth of 3, is passed through the rst convolutional layer. This layer has 64 feature kernel lters with a size of 4x4 and a stride of 2. The resulting output has dimensions of 32x32x64.

• The output of the rst convolutional layer is then processed by batch normalization, which normalizes the activations, and followed by a LeakyReLU activation function with a negative slope of 0.2.

• Next, the output is fed into the second convolutional layer, which has 128 feature kernel lters with a size of 4x4 and a stride of 2. This layer reduces the spatial dimensions to 16x16 while increasing the number of feature maps.

• Similarly, batch normalization and LeakyReLU activation are applied to the

output of the second convolutional layer. • The third convolutional layer operates on the output of the previous layer, using 256 feature kernel filters with a size of 4x4 and a stride of 2. This layer further reduces the spatial dimensions to 8x8.
• Batch normalization and LeakyReLU activation functions are applied again to the output of the third convolutional layer. • The fourth and final convolutional layer takes the output of the third layer and applies 512 feature kernel filters with a size of 4x4 and a stride of 2. This layer reduces the spatial dimensions to 4x4.
• Lastly, a single convolutional layer with a kernel size of 4x4 and a stride of 2 is used to obtain the final output of the discriminator. This layer has a single output channel, representing the probability of the input being real or fake.

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 32, 32]	3,072
BatchNorm2d-2	[-1, 64, 32, 32]	128
LeakyReLU-3	[-1, 64, 32, 32]	0
Conv2d-4	[-1, 128, 16, 16]	131,072
BatchNorm2d-5	[-1, 128, 16, 16]	256
LeakyReLU-6	[-1, 128, 16, 16]	0
Conv2d-7	[-1, 256, 8, 8]	524,288
BatchNorm2d-8	[-1, 256, 8, 8]	512
LeakyReLU-9	[-1, 256, 8, 8]	0
Conv2d-10	[-1, 512, 4, 4]	2,097,152
BatchNorm2d-11	[-1, 512, 4, 4]	1,024
LeakyReLU-12	[-1, 512, 4, 4]	0
Conv2d-13	[-1, 1, 1, 1]	8,192

Table 4.4: Nadra Discriminator Architecture

4.3 Nadra Recommendation System (NRS)

Nadra Recommendation System (NRS) is a sophisticated platform that offers market recommendations and safeguards against scams in investment transactions. With NRS, users can receive personalized recommendations on where to invest and which stocks can yield profitable revenue. Leveraging advanced algorithms and data analysis, NRS analyzes market trends, historical data, and user preferences to provide tailored investment suggestions. Moreover, NRS acts as a trusted intermediary by recommending reputable buyers and sellers, ensuring secure transactions and mitigating the risks of fraud. By integrating market analysis, personalized recommendations, and a focus on security, NRS

aims to empower users, enhance their investment experience, and facilitate successful and trustworthy transactions.

4.4 Nadra Self-Learning from Community Data

Nadra Self-Learning from Community Data revolutionizes the way our platform operates by tapping into the vast expertise of our user community. Through an interactive and collaborative environment, Nadra collects and analyzes the wealth of data generated by our community members, harnessing their insights and experiences. This data serves as a powerful resource for Nadra's self-learning capabilities, enabling the platform to constantly re ne its understanding and knowledge of the market. By leveraging the collective intelligence of our community, Nadra aims to provide users with unparalleled accuracy in predictions, valuable insights, and well-informed recommendations, empowering them to make con dent investment decisions. Our commitment to leveraging community data ensures that Nadra remains at the forefront of innovation, continuously improving and evolving to meet the diverse needs of our users.

4.5 Chatbot

Chatbots have emerged as powerful tools in the realm of arti cial intelligence and natural language processing. These intelligent conversational agents are designed to simulate human-like interactions and provide automated responses to user queries. Powered by advanced machine learning algorithms, chatbots can understand natural language inputs, interpret user intents, and generate contextually relevant and accurate responses [2]. They nd applications in various industries, including customer service, healthcare, education, and e-commerce, facilitating ef cient communication and improving user experiences. By leveraging sophisticated techniques such as natural language understanding and generation, chatbots have the potential to revolutionize the way humans interact with technology and streamline various processes.

4.5.1 Nadra's Chatbot

In our study, we propose the utilization of a chatbot based on GPT-3.5, a state-of-the-art language model, to enhance the interaction with our developed prediction models. While it's important to note that ChatGPT, as a language model, cannot provide predictions for future events beyond the knowledge it has been trained on, it can serve as a recommendation system. By integrating our models with this chatbot, we enable seamless communication between users

and our prediction algorithms. Users can input their queries, and the chatbot, leveraging advanced natural language processing capabilities, can provide recommendations related to stocks and cryptocurrency price movements based on the available data. This interactive approach offers a user-friendly and intuitive experience, empowering users to obtain insights and make informed decisions based on our models' recommendations. The integration of the chatbot serves as a bridge between users and our models, facilitating a streamlined and accessible means of accessing accurate and up-to-date information. This interactive dimension adds value to our research, allowing users to directly engage with the prediction models and harness their recommendation capabilities in a conversational manner.

4.6 Conclusion

This chapter presents the signi cant contributions of the study, focusing on AI-based services dedicated to price prediction and fraud detection in the marketplace. The primary objective is to develop ef cient predictive models using recurrent neural network (RNN) architectures, speci cally Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), to achieve accurate price prediction and trend analysis. Moreover, advanced natural language processing techniques are employed for market sentiment analysis. The performance of GPT-2 and BERT language models is ne-tuned within the platform to optimize their effectiveness. Additionally, the platform explores the generation of non-fungible tokens (NFTs) and integrates a GPT-3.5-based chatbot to enhance user interaction. Finally, the collaboration team of SIC students thoroughly explores fraud detection models in their thesis, offering comprehensive insights into the subject matter.

Chapter 5
Result Discussion and Experimentation

Chapter 5

Result Discussion and Experimentation

This chapter shifts focus from theory to presenting the results of evaluating our solution's effectiveness in cryptocurrency and stocks price prediction, sentiment analysis, and NFT generation. It includes a detailed analysis of our models' performance, limitations, conclusions, and potential applications in market prediction.

5.1 Datasets Description

5.1.1 Stocks Datasets

5.1.1.1 Cryptocurrencies Dataset

Date: Date of observation in YYYY-MM-DD format

High: Highest price of the cryptocurrency on the given date

Low: Lowest price of the cryptocurrency on the given date

Open: Opening price of the cryptocurrency on the given date

Close: Closing price of the cryptocurrency on the given date

Volume: Volume of transactions for the cryptocurrency on the given date

Marketcap: Market capitalization in USD for the cryptocurrency on the given date

The dataset is provided in CSV format.

5.1.1.2 Apple,Microsoft and Tesla Datasets

Our datasetss was collected using the Yahoo Finance, API[1] with a maximum period and 24h interval. It includes the following features:

[1]https://finance.yahoo.com/

Date: Date of observation in YYYY-MM-DD format

High: Highest price of the cryptocurrency on the given date

Low: Lowest price of the cryptocurrency on the given date

Open: Opening price of the cryptocurrency on the given date

Close: Closing price of the cryptocurrency on the given date

Volume: Volume of transactions for the cryptocurrency on the given date

The dataset is provided in CSV format.

5.1.1.3 Gold Dataset

Our dataset for Gold analysis was collected using the Yahoo Finance API[2], with a maximum period and 24h interval. It includes the following features:

Date: Date of observation in YYYY-MM-DD format

High: Highest price of the cryptocurrency on the given date

Low: Lowest price of the cryptocurrency on the given date

Open: Opening price of the cryptocurrency on the given date

Close: Closing price of the cryptocurrency on the given date

Volume: Volume of transactions for the cryptocurrency on the given date

The dataset is provided in CSV format.

5.1.1.4 Preprocessing Steps for Stocks and cryptocurrencies Datasets

The pre-processing steps for the financial datasets, such as silver, gold, and cryptocurrencies, involve a series of actions to prepare the data for further analysis. One of the primary pre-processing steps is to remove all other irrelevant columns except for the `Close` column that indicates the closing price of the financial instrument on the given date.

After removing the unnecessary columns, the next step is to create a matrix of time steps for the financial instrument, such as 100 days. Each time step contains 100 features, where the `Close` price of the instrument is recorded for each day. The next day's `Close` price is labeled as the class for the 100-day sequence.

[2] https://finance.yahoo.com/

This pre-processing step is crucial in preparing the financial data for machine learning models, particularly those that use recurrent neural networks (RNNs) to predict future prices. The time series matrix allows the model to learn the historical trends and patterns of the financial instrument, which can then be used to make predictions on future prices or trends.

5.1.2 Sentiment Analysis Dataset

5.1.2.1 Dataset 1: Reddit Crypto Sentiment

The Reddit Crypto Sentiment dataset[3] is a collection of sentiment analysis results on comments related to cryptocurrency on Reddit. The dataset indicates that 54% of the comments were classified as positive, while 46% were classified as negative.

Dataset characteristics: The dataset consists of 562 samples, stored in a CSV file format. Each sample is represented by 6 features: worker-id, task-id, task-response-id, Comment Text, Sentiment, and Reddit URL.

Features:

Worker ID: a unique identifier assigned to the worker who performed the sentiment analysis task.

Task ID: a unique identifier assigned to each task, which likely refers to the analysis of a single comment.

Task Response ID: a unique identifier assigned to each worker's response to a task, indicating their sentiment analysis result for the given comment.

Comment Text: the text of the comment that was analyzed.

Sentiment: the sentiment analysis result, which is likely categorized as positive, negative.

Reddit URL: the URL of the Reddit post where the comment was made.

Source of the dataset: The dataset is sourced from Surge AI, a platform that provides access to various datasets for machine learning projects.

[3] https://www.surgehq.ai/datasets/crypto-sentiment-dataset

5.1.2.2 Dataset 2: Stocktwits-crypto

StockTwits-crypto dataset:

The StockTwits-crypto dataset[4] is a collection of cryptocurrency-related posts from the StockTwits website, covering the period from November 1, 2021 to June 15, 2022.

Dataset characteristics:

The dataset consists of 600,000 samples, stored in a CSV file format. Each sample is represented by two features: `text` and `labels`.

Features:

The `text` feature contains the text of each post, while the `labels` feature is used to classify the sentiment of each post as negative (0), natural (1), or positive (2).

Source of the dataset:

The StockTwits-crypto dataset was collected and made available by ElKulako, and can be found on the Hugging Face datasets platform[5]

5.1.2.3 Preprocessing Steps for Reddit Crypto Sentiment and StockTwits-Crypto Datasets

The pre-processing steps are essential for ensuring that the data is in a format suitable for analysis. In this study, we conducted pre-processing on two datasets, namely the Reddit Crypto Sentiment dataset and the StockTwits-Crypto dataset, to prepare them for sentiment analysis.

For the Reddit Crypto Sentiment dataset, we removed all the columns except for CommentText and Sentiment, as these were the only features relevant to our study.

In the case of the StockTwits-Crypto dataset, we removed all the samples with a neutral label and only kept those with positive and negative labels. We then concatenated the two datasets into a single dataset, which enabled us to increase the number of samples available for analysis.

After concatenating the datasets, we removed any missing values and performed other necessary cleaning operations to ensure the quality of the data. This pre-processing step is critical as it ensures that the data is accurate, complete, and consistent, which is crucial for obtaining meaningful results.

[4] https://huggingface.co/datasets/ElKulako/stocktwits-crypto
[5] https://huggingface.co/

5.1.3 NFT Dataset

Our dataset for NFT is a comprehensive collection of all the transactions that have occurred in the CryptoPunks[6] market since its inception in 2017. This dataset includes PNG images of the 10,000 Punks, along with information on each transaction that has taken place. These transactions consist of the initial Punk Claim in 2017, sales between users, bids, and offers.
The dataset contains valuable information on each transaction, including:

txn-type: the type of transaction(Bid-Withrawn).

from: the sender.

to: the recipient.

date: the date of transaction.

timestamp: the timestamp of each transaction.

Additionally, the dataset includes information on:

source: the source of the transaction.

eth :the value of the transaction in ETH and USD equivalent.

One of the unique features of the CryptoPunks dataset is:

punk-id: the of cial Punk "attributes" that are included.

These attributes, such as Alien, Hoodie and Earring are used to generate the computer-generated Punk images. The dataset includes information on the attributes associated with each Punk, allowing researchers to analyze the distribution of attributes across the Punk population.
The dataset also includes additional information such as:

from-wallet-address: the wallet addresses of the sender.

to-wallet-address: the wallet addresses of the recipient.

type: the type of the cryptopunk (e.g., male or female).

accessories: the accessories associated with each Punk(e.g., Green Eye Shadow, Earring, Blonde Bob).

[6] https://www.kaggle.com/datasets/tunguz/cryptopunks

5.1.3.1 Preprocessing Steps for NFT Dataset

Keep only the necessary columns for analysis, such as `transaction_type`, `date`, `eth`, `punk-id`, `type`, and `accessories`.
Convert the `type` column from a list to individual rows using the `explode()` method of the pandas DataFrame.
Transform the images into a format suitable for deep learning using the `torchvision.transforms.Compose()` method. This should include resizing the images, converting them to tensors, and normalizing the pixel values.

5.2 Implementation tools

5.2.1 TensorFlow

TensorFlow[7] is an open-source software library for data flow and differentiable programming across a range of tasks. It is widely used for deep learning applications such as neural network training, natural language processing, and computer vision.

5.2.2 Keras

Keras[8] is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, Theano, or Microsoft Cognitive Toolkit. It is designed to enable fast experimentation with deep neural networks and can be used for image and text classification, among other tasks.

5.2.3 Scikit-learn (sklearn)

Sklearn[9] is a machine learning library in Python that provides simple and efficient tools for data mining and data analysis. It provides various supervised and unsupervised learning algorithms and tools for model selection, preprocessing, evaluation, and visualization. It's based on familiar technologies such as NumPy, pandas, and Matplotlib. Scikit-learn provides many functionalities, including regression (such as Linear Regression), classification (like K-Nearest Neighbors), clustering (including K-Means and K-Means++), model selection, and preprocessing (such as Min-Max Normalization).

[7] https://www.tensorflow.org/
[8] https://keras.io/
[9] https://scikit-learn.org/stable/

5.2.4 Torch

Torch[10] is a widely-used open-source machine learning framework based on the Lua programming language. It is particularly well-suited for developing deep learning models, and provides support for a wide range of model types, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers. Torch also includes a number of useful tools for model training and evaluation, such as support for parallel training and optimization algorithms like stochastic gradient descent (SGD).

5.2.5 PIL

PIL[11] is a popular library for working with images in Python. It provides a wide range of image processing functionality, including support for reading and writing various image formats, manipulating images through operations such as resizing, cropping, and rotation, and applying various filters and effects. PIL also includes tools for working with image metadata and color profiles, and provides support for working with images in different color spaces.

5.2.6 Flask

Flask[12] is a popular web framework for building Python web applications. It is designed to be simple and lightweight, and provides a range of tools for building and deploying web applications quickly and easily. Flask includes support for routing and request handling, template rendering, and database integration, and provides a range of extensions that can be used to add additional functionality to the framework, such as support for user authentication and authorization, caching, and API development. Flask is particularly well-suited for building microservices and other small-scale web applications.

5.3 Programming language

5.3.1 Python

Python[13] is a popular interpreted, high-level, object-oriented programming language with dynamic semantics. Its built-in data structures, combined with dynamic typing and dynamic binding, make it an attractive choice for Rapid Application Development and scripting. Python emphasizes readability with

[10] https://pytorch.org/
[11] https://pypi.org/project/Pillow/
[12] https://pypi.org/project/Flask/
[13] https://www.python.org/

its simple, easy-to-learn syntax, reducing the cost of program maintenance. It supports modules and packages, encouraging program modularity and code reuse. The Python interpreter and extensive standard library are available without charge in source or binary form for all major platforms and can be freely distributed.

5.3.2 HTML (Hypertext Markup Language)

HTML[14] is a standard markup language used to create web pages and applications. It provides a set of tags and attributes that can be used to de ne the structure and content of a web page, such as headings, paragraphs, lists, and links. HTML also provides support for multimedia elements like images and videos, and can be used to create interactive forms and other user interfaces.

5.3.3 CSS (Cascading Style Sheets)

CSS[15] is a stylesheet language used to describe the presentation of a web page or application. It provides a range of style rules that can be used to control the appearance of HTML elements, such as fonts, colors, margins, and borders. CSS allows developers to separate the presentation of a web page from its content, making it easier to maintain and update.

5.3.4 JavaScript

JavaScript[16] is a popular scripting language used to create dynamic web pages and applications. It provides a range of tools for manipulating the content and behavior of a web page in response to user interactions or other events. JavaScript can be used to create animations, validate form input, and make HTTP requests to web servers, among other things. It is also used as the primary language for many popular front-end frameworks and libraries, such as React and Vue.

5.3.5 SQL (Structured Query Language)

SQL[17] is a domain-speci c language used to manage and manipulate relational databases. It provides a set of commands and syntax for creating and modifying database structures, inserting and retrieving data, and performing complex queries and analyses. SQL is widely used in enterprise applications

[14] https://developer.mozilla.org/en-US/docs/Web/HTML
[15] https://developer.mozilla.org/en-US/docs/Web/CSS
[16] https://www.javascript.com/
[17] https://www.w3schools.com/SQL/

and data-driven web applications, and is supported by a wide range of relational database management systems, including MySQL, PostgreSQL, and Oracle.

5.4 Evaluation measures

5.4.1 Evaluation measures for cryptocurrency price prediction

5.4.1.1 MAE (Mean Absolute Error)

It measures the average magnitude of the errors in a set of predictions. It is calculated as the average absolute difference between the actual values and the predicted values [47]. Lower values of MAE indicate better performance.

The formula for MAE is:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{5.1}$$

where:
n is the number of samples.
y_i is the true value.
$\hat{y}_i$ is the predicted value.

5.4.1.2 MSE (Mean Squared Error)

It measures the average of the squared differences between the actual values and the predicted values. It is a more sensitive metric than MAE to larger errors as it squares the difference [47]. Lower values of MSE indicate better performance.

The formula for MSE is:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{5.2}$$

where:
n is the number of samples.
y_i is the true value.
$\hat{y}_i$ is the predicted value.

5.4.1.3 R^2 score

It measures the proportion of variance in the dependent variable (cryptocurrency price) that is predictable from the independent variables (features used for

prediction). It ranges from 0 to 1, with higher values indicating better performance. A value of 1 indicates a perfect prediction, while a value of 0 indicates that the model does not explain any variability in the target variable [47].

The formula for R^2 score is:

$$R^2 = 1 \quad \frac{SS_{res}}{SS_{tot}} \tag{5.3}$$

where

$$SS_{res} = \sum_{i=1}^{n} (y_i \quad \hat{y}_i)^2 \tag{5.4}$$

$$SS_{tot} = \sum_{i=1}^{n} (y_i \quad y)^2 \tag{5.5}$$

where

y_i is the actual value of the target variable for the i-th observatio.
$\hat{y}_i$ is the predicted value of the target variable for the i-th observation.
y is the mean of the actual values of the target variable.

5.4.2 Evaluation measures for Sentiment Analysis Classi cation

5.4.2.1 Confusion matrix

	Actual Positive	Actual Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Table 5.1: Confusion Matrix [52]

A confusion matrix is a table that is commonly used to evaluate the performance of a classi cation model. The matrix summarizes the classi cation results and shows the number of correctly and incorrectly classi ed instances, also known as true positives, true negatives, false positives, and false negatives.

In a binary classi cation problem,the confusion matrix will have four possible outcomes:

True Positive (TP): The number of samples that are correctly predicted as positive.

True Negative (TN): The number of samples that are correctly predicted as negative.

False Positive (FP): The number of samples that are incorrectly predicted as positive, but are actually negative.

False Negative (FN): The number of samples that are incorrectly predicted as negative, but are actually positive.

5.4.2.2 Precision

Precision measures the proportion of true positive predictions among all positive predictions made by the model. It is calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{5.6}$$

5.4.2.3 Recall

Recall measures the proportion of true positive predictions among all actual positive cases in the dataset. It is calculated as:

$$Recall = \frac{TP}{TP + FN} \tag{5.7}$$

5.4.2.4 F1 score

F1 score is the harmonic mean of precision and recall, providing a balanced measure of the two metrics. It is calculated as:

$$F1 = \frac{2 \quad precision \quad recall}{precision + recall} \tag{5.8}$$

5.4.3 Evaluation measures for NFT Generation

5.4.3.1 Inception Score

The Inception Score measures how well the generated images capture the essence of a given dataset. It does this by comparing the statistics of the generated images to those of the real images in the dataset. Specifically, the Inception Score uses the output of a pre-trained Inception network to calculate the scores. The Inception network is a convolutional neural network (CNN) that is trained on the ImageNet dataset and is capable of recognizing a wide variety of objects. The formula for Inception Score is as follows:

$$IS(G) = \exp(E[KL(P(y|x) \,||\, P(y))]) \tag{5.9}$$

G: is the generator that generates the images.

x: is the noise input to the generator.

y: is the label predicted by the Inception network.

$P(y|x)$: is the conditional distribution of the Inception network's predictions given the generated images.

$P(y)$: is the marginal distribution of the Inception network's predictions over the real dataset.

KL: is the Kullback-Leibler divergence, which measures the difference between two probability distributions.

$E[:::]$: is the expected value over the generated images.

5.4.3.2 Frechet Inception Distance (FID)

FID measures the distance between the distributions of the real images in a dataset and the generated images. It uses the output of a pre-trained Inception network to calculate the distance between the two distributions. The Inception network is a convolutional neural network (CNN) that is trained on the ImageNet dataset and is capable of recognizing a wide variety of objects.
The FID metric is calculated using the following formula:

$$FID = k_1 \quad {}_2k^2 + Tr(\quad_1 + \quad_2 \quad 2(\ _1\ _2)^{\frac{1}{2}}) \qquad (5.10)$$

where:

$_1$ and $_2$ are the mean vectors of the activations of the Inception network on the real images and the generated images, respectively.

$_1$ and $_2$ are the covariance matrices of the activations of the Inception network on the real images and the generated images, respectively.

$Tr()$ is the trace operator, which sums the diagonal elements of a matrix.

5.4.3.3 Kernel Inspection Distance (KID)

KID measures the distance between the distributions of the real images in a dataset and the generated images in the feature space of a pre-trained Inception network. It uses the kernel trick to map the activations of the Inception network to an in nite-dimensional feature space, where the distance between the distributions can be calculated more accurately.
The KID metric is calculated using the following formula:

$$KID = jj_1 \quad {}_2jj^2 + Tr(\ K_1 + K_2 \quad 2(K_1K_2)^{\frac{1}{2}}) \qquad (5.11)$$

where:

μ_1 and μ_2 are the mean vectors of the activations of the Inception network on the real images and the generated images, respectively.

K_1 and K_2 are the kernel matrices of the activations of the Inception network on the real images and the generated images, respectively.

$Tr()$ is the trace operator, which sums the diagonal elements of a matrix.

5.5 Result of Price Prediction

5.5.1 Result of Bitcoin

To improve the accuracy of the price prediction model, we conducted several experiments using recurrent neural network (RNN) models such as LSTM and GRU.
The dataset was divided into training, validation, and test sets with an 80:10:10 ratio.
We evaluated the impact of different hyperparameters, including loss functions (Huber, MAE, MSE), optimizers (Adam, SGD, RMSProp), batch sizes (12, 32, 64, 128, 256), and learning rates (0.001, 0.002, 0.003) on the model accuracy.
The results of the experiments were analyzed, and the model with the best performance was selected. To measure the performance of the models, we used metrics such as mean squared error (MSE), mean absolute error (MAE), and R^2 score.

5.5.1.1 LSTM+ GRU Model

Our results showed that the RMSProp optimizer with the MAE loss function and a batch size of 64 with a learning rate of 0.001 gave the best results. For LSTM+GRU model

Model	LSTM+GRU
Optimizer	RMSProp
Loss Function	MAE
Batch Size	64
Learning Rate	0.001
MSE	8.8031e-05
MAE	0.00598
R^2 score	0.96

Table 5.2: Hyperparameters and Evaluation Metrics of LSTM+GRU.

Figure 5.1: Training and Validation Loss of LSTM+GRU Model for Bitcoin

5.5.1.2 LSTM Model

Our results showed that the Adam optimizer with the MAE loss function and a batch size of 12 with a learning rate of 0.001 gave the best results. For LSTM model

Model	LSTM
Optimizer	Adam
Loss Function	MAE
Batch Size	12
Learning Rate	0.001
MSE	0.000128
MAE	0.00750
R2 score	0.97

Table 5.3: Hyperparameters and Evaluation Metrics of LSTM.

Figure 5.2: Training and Validation Loss of LSTM Model for Bitcoin

5.5.1.3 GRU Model

Our results showed that the RMSProp optimizer with the Huber loss function and a batch size of 64 with a learning rate of 0.001 gave the best results. For GRU model

Model	GRU
Optimizer	RMSProp
Loss Function	Huber
Batch Size	64
Learning Rate	0.001
MSE	8.1091e-05
MAE	0.00587
R2 score	0.97

Table 5.4: Hyperparameters and Evaluation Metrics of GRU.

Figure 5.3: Training and Validation Loss of GRU Model for Bitcoin

Figure 5.4: Comparision of GRU, LSTM And GRU+LSTM Models For Bitcoin

5.5.2 Testing the Models On Other Cryptocurrencies

In addition to analyzing the performance of the models on Bitcoin, we also extended our evaluation to other prominent cryptocurrencies. This subsection focuses on testing the effectiveness of the model on cryptocurrencies such as Ethereum, Binance Coin, Ripple, Doge-coin, USD Coin, and Litecoin. By assessing the model's performance across multiple cryptocurrencies, we aim to gain insights into its generalizability and robustness. The results obtained from these experiments provide valuable information on the model's ability to predict price across different cryptocurrency markets.
The results of the models performance on these additional cryptocurrencies are summarized in the following table:

Model	Coin Name	Metrics		
		MSE	MAE	R2 Score
GRU	Ethereum	0.00030	0.010	0.99
	Binance	0.00037	0.010	0.99
	XRP (Ripple)	0.00020	0.0088	0.97
	Dogecoin	0.00033	0.0070	0.98
	USD Coin	0.0019	0.019	0.55
	Litecoin	0.00031	0.0087	0.98
LSTM	Ethereum	0.00085	0.017	0.98
	Binance	0.00088	0.016	0.98
	XRP (Ripple)	0.00037	0.014	0.95
	Dogecoin	0.00048	0.010	0.97
	USD Coin	0.0017	0.018	0.60
	Litecoin	0.00051	0.013	0.98
LSTM+GRU	Ethereum	0.00037	0.012	0.99
	Binance	0.00044	0.012	0.99
	XRP (Ripple)	0.00019	0.0098	0.97
	Dogecoin	0.00033	0.0064	0.98
	USD Coin	0.0017	0.018	0.59
	Litecoin	0.00033	0.0089	0.98

Table 5.5: Performance of GRU, LSTM, and LSTM+GRU Models on Other Cryptocurrencies

The gures display a comparison between the predicted and original values of cryptocurrencies. It provides a visual representation of the model's performance in forecasting cryptocurrency prices. The predicted values, represented by the orange curve, indicate the estimated price movements based on the trained model's predictions. On the other hand, the original values, represented by the blue curve, depict the actual price data obtained from historical records. By examining the gure, one can assess the model's accuracy in capturing the trends and patterns of the cryptocurrency market.

Figure 5.5: Ethereum

Figure 5.6: Binance

Figure 5.7: DogeCoin

Figure 5.8: LiteCoin

Figure 5.9: Ripple

Figure 5.10: The Performance of GRU Model

Figure 5.11: Ethereum

Figure 5.12: Binance

Figure 5.13: DogeCoin

Figure 5.14: LiteCoin

Figure 5.15: Ripple

Figure 5.16: The Performance of LSTM Model

Figure 5.17: Ethereum

Figure 5.18: Binance

Figure 5.19: DogeCoin

Figure 5.20: LiteCoin

Figure 5.21: Ripple

Figure 5.22: The Performance of LSTM+GRU Model

5.5.3 Testing the Models On Other Stocks

Furthermore, besides assessing the performance of the models on diverse cryptocurrencies, we extended our evaluation to encompass prominent stocks like Apple, Microsoft, and Tesla, as well as precious metals such as gold and silver. This broader scope allowed us to gauge the models' effectiveness in capturing market dynamics beyond the cryptocurrency realm. Through these supplementary experiments, we gained valuable insights into the models' versatility and their potential for broader nancial market analysis. The results derived from testing on these varied assets contribute to a more comprehensive comprehension of the models' predictive capabilities across multiple investment instruments.
The results of the models' performance on these additional assets are summarized in the following tables:

Model	Stock Name	Metrics		
		MSE	MAE	R2 Score
GRU	Apple	3.64e-05	0.0039	0.99
	Tesla	0.00015	0.0075	0.99
	Microsoft	3.58e-05	0.0037	0.99
	Silver	0.00015	0.0081	0.99
	Gold	0.00011	0.0076	0.99
LSTM	Apple	7.054e-05	0.0043	0.99
	Tesla	0.00017	0.0065	0.99
	Microsoft	7.043e-05	0.0047	0.99
	Silver	0.00018	0.0094	0.99
	Gold	0.00027	0.011	0.99
LSTM+GRU	Apple	6.95e-05	0.0044	0.99
	Tesla	0.00020	0.0093	0.99
	Microsoft	8.33e-05	0.0060	0.99
	Silver	0.00016	0.0083	0.99
	Gold	0.00022	0.010	0.99

Table 5.6: Performance of GRU, LSTM, and LSTM+GRU Models on Other Stocks and Precious Metals

The gures present a side-by-side comparison of predicted and original stock values, offering a visual depiction of the model's effectiveness in forecasting stock prices. The orange curve represents the predicted values, re ecting the estimated price movements derived from the model's trained predictions. In contrast, the blue curve represents the original values, showcasing the actual price data obtained from historical records. Through careful examination of these gures, one can evaluate the model's accuracy in capturing the intricate trends and patterns within the stock market.

Figure 5.23: Apple

Figure 5.24: Tesla

Figure 5.25: Microsoft

Figure 5.26: Gold

Figure 5.27: Silver

Figure 5.28: The Performance of GRU Model

Figure 5.29: Apple

Figure 5.30: Tesla

Figure 5.31: Microsoft

Figure 5.32: Gold

Figure 5.33: Silver

Figure 5.34: The Performance of LSTM Model

Figure 5.35: Apple

Figure 5.36: Tesla

Figure 5.37: Microsoft

Figure 5.38: Gold

Figure 5.39: Silver

Figure 5.40: The Performance of LSTM+GRU Model

5.6 Result of Market Sentiment Analysis

5.6.1 GPT-2

The market sentiment analysis using the GPT-2 model yielded promising results. The accuracy of the model was evaluated on both the training and validation sets. The training set achieved an accuracy of 85.0%, indicating the model's ability to capture sentiment patterns effectively. The validation set also performed well with an accuracy of 85.2%, demonstrating the model's ability to generalize to new data. The loss metrics were analyzed as well, with the training set showing a loss of 0.36589 and the validation set showing a loss of 0.35903. These low loss values indicate the model's capability to minimize errors and t the data well.

Figure 5.41: Model Accuracy

Figure 5.42: Model Loss

The confusion matrix shows that the model accurately predicted positive labels in 97% of the cases (true positive) but incorrectly predicted negative labels

Figure 5.43: Model Confusion matrix

in 3% of the cases (false negative). However, the model had a higher error rate in predicting negative labels, as it correctly identi ed negative labels in only 22% of the cases (true negative) but incorrectly predicted positive labels in 78% of the cases (false positive).

5.6.2 BERT

The evaluation results of the BERT model demonstrate promising performance. During the training phase, the model achieved a relatively low training loss of 0.1999 and a high accuracy of 0.9232, indicating its effectiveness in learning from the training data and making accurate predictions. Similarly, the model's performance on the validation set was notable, with a validation loss of 0.2924 and an accuracy of 0.8921. These results suggest that the BERT model successfully generalized its learned knowledge to unseen data, exhibiting strong predictive capabilities.

Model	Validation Loss	Validation Accuracy	Training Loss	Training Accuracy
GPT-2	0.35	85.2%	0.36	85.0%
BERT	0.29	89%	0.19	92%

Table 5.7: Comparison between GPT-2 and BERT

Figure 5.44: Model Accuracy

Figure 5.45: Model Loss

5.7 Result of NFT Generation

In our study, we conducted a series of experiments to train our Generative Adversarial Network (GAN) model for NFT generation. The experiments focused on exploring different con gurations, including varying the batch size, learning rate, and optimizers. The details of these experiments and their results are as follows:

To optimize the performance of our GAN model, we implemented the Adam optimizer. Adam has been chosen for its advantages, such as faster convergence and reduced computational time. The decision to utilize Adam as the optimizer is justi ed by its ability to yield satisfactory results in our experimentation with the GAN model.

The learning rate is a crucial factor in training machine learning models, including the Deep Convolutional Generative Adversarial Network (DCGAN) employed in our study. We selected a learning rate of 0.0002, consistent with the original DCGAN paper by Goodfellow et al. The learning rate determines the magnitude of parameter updates during training. An incorrect learning rate can lead to issues like mode collapse, where the generator fails to capture the full diversity of the target distribution, resulting in limited variation and quality in the generated images.

Additionally, we set the z-dim variable to 100, representing the dimensionality of the noise vector used as input to the generator network. This choice ensures that the noise vector has 100 elements, allowing for the generation of diverse and high-quality images.

To enhance the performance of the DCGAN model and mitigate mode collapse, we experimented with different values of Batch Normalization, a critical parameter in uencing the model's learning speed. Through our experiments, we determined the speci c value of Batch Normalization that yielded the best outcome.

Furthermore, we investigated various variations of the LeakyReLU Activation Function and compared their results to identify the con guration that produced the most desirable outcome.

In our experimentation, we also implemented Convolutional and Transpose Convolutional Layers to assess their impact on the model's performance. By analyzing the results, we gained insights into the effectiveness of these layers in the context of NFT generation.

Figure 5.46: Training Loss for Generator and Discriminator

In the case of GANs, accuracy is not commonly employed as a performance metric. Instead, GANs are evaluated based on the quality and diversity of the generated samples and their ability to capture the underlying data distribution. As GANs lack explicit labels or ground truth, accuracy is not a meaningful metric in this context. To assess the quality and diversity of the generated images, our model was evaluated using the Inception Score (IS), resulting in a score of 3.40.

Based on the experimentation, we have determined the parameter con guration:

Model	Optimizer	Learning Rate	Batch Size	Epochs
DCGAN	Adam	0.0002	128	400

Table 5.8: Con guration Parameters for DCGAN Training

5.7.1 Generated NFT Images

Figure 5.47: Real NFT

Figure 5.48: Generated NFT

5.8 Conclusion

This chapter presents a comprehensive overview of the evaluation measures utilized in our research, providing a detailed discussion and analysis of the results obtained from our experimental investigations. The implications and signi cance of these ndings are thoroughly examined and elucidated. Furthermore, the implementation stages of our work are extensively explored, highlighting the key methodologies and techniques employed. Additionally, a comparative study is conducted to evaluate the performance of various prediction models and determine the most effective approach. Moreover, the BERT and GPT-2 language models are compared, with a focus on their strengths and weaknesses in the context of sentiment classi cation. Lastly, the effectiveness of the GAN model in generating images is assessed.

Chapter 6
Nadra Web Application

Chapter 6

Nadra

6.1 Web Application

6.1.1 Home Page

Figure 6.1: Home Page

Figure 6.2: Our Features Page

Figure 6.3: About Us Page

Figure 6.4: FAQ Page

6.1.2 Sign Up Page

Figure 6.5: Sign Up Page

6.1.3 Sign In Page

Figure 6.6: Sign In Page

Figure 6.7: Sign In Page

Figure 6.8: Sign In Page

Figure 6.9: Home after Sign In

6.1.4 Services Page

Figure 6.10: Market prices Page 1

Figure 6.11: Market prices Page 2

Figure 6.12: Market News Page

Figure 6.13: Prediction Services Page

Figure 6.14: Nadra Precious Metals Service

Figure 6.15: Nadra Oil Gas Prediction Service

Figure 6.16: Nadra Cryptocurrency Prediction Service

Figure 6.17: Fraud Detection Services Page

Figure 6.18: Nadra Fake News Detection Service

Figure 6.19: Nadra Fake NFT Detection Service

6.1.5 Community of Experts

Figure 6.20: Community of Nadra

Figure 6.21: Community of Nadra

6.1.6 Nadra Bot VS ChatGPT

Figure 6.22: Nadra Bot

Figure 6.23: Nadra Bot

Figure 6.24: Nadra Bot

Figure 6.25: Nadra Bot

Figure 6.26: Chatgpt

Figure 6.27: Chatgpt

Figure 6.28: Chatgpt

6.2 Mobile Application

6.2.1 Home Page

Figure 6.29: Home Page

Figure 6.30: Features Page

Figure 6.31: Our Community

Figure 6.32: About Us

6.2.2 Sign Up Page

Figure 6.33: Contact Us

Figure 6.34: Sign Up

6.2.3 Sign In Page

Figure 6.35: Sign In

Figure 6.36: Dashboard

6.2.4 Services Page

Figure 6.37: Market Prices Page

Figure 6.38: Fraud Detection Page

Figure 6.39: Nadra Fake NFT Detection