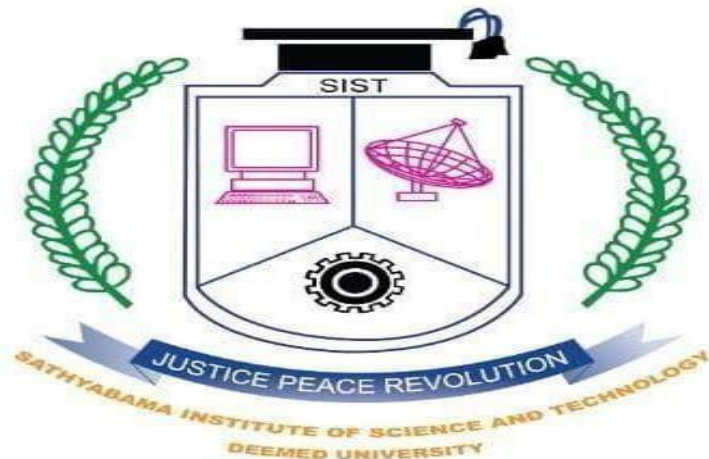


OPTIONS TRADING USING MACHINE LEARNING

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Technology Degree in Information Technology

By

Pranesh Priyanshu (40120112)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category-1 University by UGC

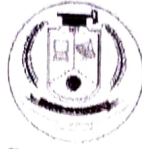
Accredited with Grade “A++” by NAAC | 12B Status by UGC |

Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI, TAMILNADU – 600119

APRIL – 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

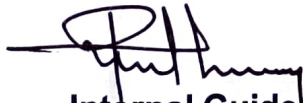
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Pranesh Priyanshu (40120112)** and **Ritik Raj (40120137)** who carried out the Project Phase-2 entitled "**OPTIONS TRADING USING MACHINE LEARNING**" under my supervision from January 2024 to April 2024.


Internal Guide

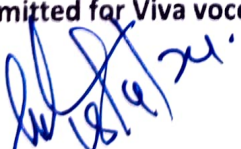
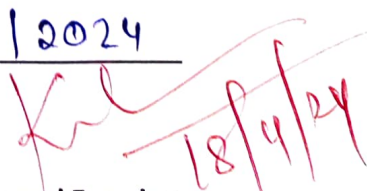
Dr. S Revathy, M.E., Ph.D.



Head of the Department

Dr. R Subhashini, M.E., Ph.D.

Submitted for Viva voce Examination held on 18/04/2024


Internal Examiner
External Examiner

DECLARATION

I Ritik Raj (Reg. No. 40120137) hereby declare that the Phase-2 Project Report entitled "**Options Trading Using Machine Learning**" is done by me under the guidance of **Dr. S Revathy , M.E , Ph.D.** and is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 18/04/2024

A handwritten signature in blue ink, appearing to read 'Ritik Raj', with a horizontal line underneath.

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management** of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T Sasikala, M.E., Ph.D.**, Dean, School of Computing

Dr. R Subhashini, M.E., Ph.D., Head of the Department of Information Technology for providing me with necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. S Revathy, M.E., Ph.D.** for her valuable guidance, suggestions and constant encouragement that paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

Options trading, a multifaceted sector of finance, has seen a profound transformation in recent years with the integration of machine learning methodologies. This initiative, "Utilizing Machine Learning in Options Trading," delves into the utilization of machine learning algorithms across various facets of options trading, including forecasting, risk control, strategy formulation, and market examination. Beginning with an exploration of the fundamentals of options trading, the project emphasizes key factors influencing options pricing, such as volatility and time decay. It highlights the unique attributes of options, making them well-suited for machine learning applications due to their non-linear nature and complex relationships. The project comprehensively examines machine learning algorithms like decision trees, random forests, support vector machines, and neural networks within the context of options trading, showcasing their transformative potential. By integrating state-of-the-art machine learning methodologies, options traders can effectively respond to rapidly changing market conditions, optimizing profitability, and strengthening risk management strategies, ultimately uncovering new opportunities within the options market. One of the significant advantages of utilizing machine learning in options trading is its ability to uncover hidden opportunities and exploit market inefficiencies. By analyzing historical data and identifying patterns that may not be apparent to human traders, machine learning algorithms can uncover lucrative trading opportunities that would otherwise go unnoticed. Additionally, machine learning models can adapt and evolve over time, continuously improving their accuracy and effectiveness in navigating the complexities of the options market. Furthermore, machine learning plays a crucial role in risk management within options trading. By leveraging advanced algorithms, traders can assess the potential risk associated with different trading strategies and implement appropriate risk mitigation measures. Machine learning models can analyze historical data to identify potential outliers and anomalies, enabling traders to proactively manage risks and minimize potential losses. The integration of machine learning methodologies has revolutionized the options trading landscape, offering traders powerful tools for forecasting, risk management, and strategy optimization. By harnessing the capabilities of machine learning algorithms, options traders can gain a competitive edge in the market, adapt to changing conditions, and capitalize on lucrative opportunities while effectively managing risks.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	Abstract	iv
	List of Figures	vi
1	Introduction	1
2	Literature Survey	
2.1	Literature Survey	3
2.2	Inferences from the Literature Survey	5
2.3	Existing System and Proposed System	12
3	Requirement Analysis	
3.1	Feasibility Studies	14
3.2	Hardware and Software Requirements	17
4	Description Of Proposed System	
4.1	Selected Methodology or process model	19
4.2	Architecture / Overall Design of Proposed System	25
4.3	Description of Software for Implementation and Testing Plan of The Proposed System	27
5	Results and Discussion	
5.1	Results	33

6	Conclusion and Future Scope	39
7	References	41

List of Figures

Figure Number	Figure Name	Page No.
4.1.1.1	SVM Performance Model	21
4.1.1.2	System Architecture for Creating a Knowledge Base	25
4.2	Data Integration	26
4.4	Sequential Model	31
5.1.1	AMC close based on date change	34
5.1.2	Strategy signal	35
5.1.3	Prediction signal	36
5.1.4	Stike Price vs Return	37
5.1.5	Return based on performance	38

List of Tables

Table Number	Table Description	Page No.
3.2.2	Details of all Python libraries used	18

CHAPTER 1

INTRODUCTION

Options trading, a multifaceted financial realm, has grown exponentially in complexity and popularity over the years. Offering investors a unique array of strategies for risk management, speculation, and portfolio diversification, it remains a dynamic and ever-evolving landscape. The emergence of machine learning, a subset of artificial intelligence, has not only revolutionized our world but has also fundamentally transformed the way market participants approach options trading. The integration of machine learning techniques brings to the forefront predictive models, risk management strategies, and innovative methods for market analysis. This project, titled "Machine Learning in Options Trading," embarks on a comprehensive exploration of the symbiotic relationship between machine learning and the options market, aiming to demystify the intricacies and showcase the transformative power of artificial intelligence in this domain.

Options trading is marked by its intricate web of financial instruments and strategies, where financial derivatives provide the holder the right, but not the obligation, to buy or sell an underlying asset at a specified price (strike price) on or before a predetermined date (expiration date). The dynamics of options pricing are deeply entwined with factors like market volatility, time decay, and movements in the underlying asset, each of which introduces intricate nonlinear relationships that are ripe for exploration through machine learning algorithms. Machine learning models excel at identifying and leveraging patterns within such complex data sets, enabling traders to make more informed decisions and better anticipate market movements.

Through this project, we aim to unveil the transformative potential of machine learning in the world of options trading. The multidisciplinary approach to harnessing advanced machine learning techniques not only empowers traders to adapt to the rapidly changing market but also propels the realm of options trading into a new era of intelligent, data-driven decision-making. The confluence of financial expertise and machine learning innovation holds the promise of more efficient, profitable, and risk-aware options trading strategies, setting a course for the evolution of this dynamic market space. As machine learning continues to advance, its integration into options trading will likely lead to the development of more sophisticated algorithms capable of capturing nuanced market

behaviors and enhancing trading outcomes. and data-driven approach to strategy development, leveraging advanced optimization techniques to identify optimal entry and exit points, position sizes, and risk parameters. In addition to risk management and strategy optimization, machine learning has also revolutionized the field of market analysis in options trading. Traditionally, options traders relied on a combination of technical analysis, fundamental analysis, and market sentiment to assess market conditions and identify potential trading opportunities. However, these traditional methods are often subjective, time-consuming, and prone to human error. Machine learning algorithms offer a more objective and data-driven approach to market analysis, leveraging advanced statistical techniques to extract meaningful insights from vast amounts of market data. For example, machine learning models can analyze historical price data, option trading volumes, volatility trends, and other relevant factors to identify patterns and correlations that may signal impending market movements. By identifying these patterns in real-time, options traders can capitalize on emerging opportunities and adjust their trading strategies accordingly. Moreover, machine learning algorithms can also detect anomalies and outliers in market data, helping traders to identify potential risks and opportunities that may not be apparent through traditional analysis methods. Overall, the integration of machine learning into options trading represents a paradigm shift in the way that traders approach financial markets. By harnessing the power of advanced algorithms and big data analytics, options traders can gain deeper insights into market dynamics, mitigate risks more effectively, and develop more robust and adaptive trading strategies. Moreover, as machine learning technology continues to evolve and mature, its impact on options trading is likely to grow even more significant in the years ahead, ushering in a new era of innovation and opportunity in the world of finance.

CHAPTER 2

LITERATURE SURVEY

2.1 Literature Survey

Options trading has long been a cornerstone of financial markets, offering investors a diverse range of strategies for managing risk and enhancing portfolio performance. With the rapid advancements in technology, particularly in the field of machine learning, there has been a growing interest in leveraging these tools to optimize decision-making processes within options trading. This literature survey aims to explore the integration of machine learning methodologies in options trading, highlighting its potential to revolutionize the way traders analyze market data, formulate strategies, and manage risk. By examining existing research and developments in this field, we seek to uncover the transformative power of machine learning in enhancing the efficiency and effectiveness of options trading strategies.

2.1.1 Fundamentals

Before delving into the application of machine learning in options trading, it is essential to establish a foundational understanding of the options market. Options are financial derivatives that provide the holder with the right, but not the obligation, to buy or sell an underlying asset at a predetermined price (strike price) on or before a specified date (expiration date). This market is characterized by its complexity, with factors such as market volatility, time decay, and underlying asset movements influencing options pricing dynamics. Understanding these fundamentals lays the groundwork for exploring how machine learning can be effectively applied to address the challenges and opportunities within the options trading landscape.

2.1.2 Introduction to Machine Learning

Machine learning represents a subset of artificial intelligence focused on developing algorithms that enable computers to learn from and make predictions or decisions based on data. Within the realm of financial analysis, machine learning techniques have proven to be invaluable in extracting insights from complex datasets and identifying patterns that may not be discernible through traditional statistical methods. Common machine learning algorithms employed in financial applications include decision trees, random

forests, support vector machines, and neural networks. These algorithms offer the potential to enhance the accuracy and efficiency of options trading strategies by leveraging vast amounts of historical and real-time market data.

2.1.3 Previous Research on Options Trading Using Machine Learning

A comprehensive review of existing literature reveals a growing body of research exploring the application of machine learning in options trading. Studies have examined various aspects of options trading, including predictive modeling, risk management strategies, and market analysis techniques. Researchers have utilized machine learning algorithms to develop models for forecasting options prices, identifying profitable trading opportunities, and optimizing portfolio allocation strategies. By synthesizing the findings of these studies, we can gain insights into the effectiveness and limitations of machine learning approaches in options trading and identify areas for further research and development.

2.1.4 Methodologies and Approaches

Methodological approaches employed in previous research on options trading using machine learning vary widely, reflecting the diverse nature of this field. Some studies have focused on building predictive models using historical options and market data, while others have explored more advanced techniques such as deep learning and reinforcement learning. Additionally, researchers have utilized different data sources, including tick-level market data, options chain data, and sentiment analysis from news and social media sources. Feature selection methods and model validation techniques also play a crucial role in ensuring the robustness and reliability of machine learning models in options trading applications.

2.1.5 Analysis of Findings

An analysis of the findings from existing research sheds light on the strengths and weaknesses of machine learning approaches in options trading. While machine learning algorithms have demonstrated promising results in certain areas, such as predicting options prices and identifying trading opportunities, challenges remain in terms of model interpretability, robustness to market changes, and scalability to real-world trading environments. Additionally, there is a need for further research to address issues related to data quality, overfitting, and model generalization across different market conditions.

2.1.6 Practical Implications and Applications

Despite the challenges, the integration of machine learning in options trading holds significant promise for enhancing decision-making processes and improving trading outcomes. By leveraging machine learning algorithms, traders can gain insights into market trends, identify patterns in options pricing dynamics, and develop more sophisticated trading strategies. Machine learning techniques also offer opportunities for automating certain aspects of options trading, such as trade execution and risk management, thereby enabling traders to react more quickly to changing market conditions and optimize their trading performance.

2.2 Inferences from the Literature Survey

The literature survey on options trading using machine learning reveals promising opportunities and challenges in the integration of advanced algorithms into financial markets. Machine learning techniques offer potential for improving decision-making, risk management, and portfolio optimization within options trading. Studies indicate that predictive modeling can enhance options price forecasts, while sophisticated algorithms can aid in developing robust risk management strategies. However, challenges such as data quality and model interpretability persist, requiring further research and innovation to fully realize the benefits of machine learning in options trading. Overall, the survey highlights the transformative potential of machine learning while emphasizing the need for ongoing exploration and refinement in this dynamic field.

2.2.1 A Machine Learning Approach to Forecast International Trade: The Case of Croatia

The research paper highlights the potential of machine learning techniques to improve the accuracy and reliability of international trade forecasts, particularly in the context of Croatia. While machine learning offers numerous merits, such as enhanced accuracy, scalability, and adaptability, it also presents challenges related to data quality, interpretability, and overfitting. Addressing these challenges is crucial for realizing the full potential of machine learning in trade forecasting and informing effective policymaking and business decisions.

2.2.1.1 Key Findings

Machine Learning (ML) Techniques: The research paper employs machine learning

techniques to forecast international trade, specifically focusing on the case of Croatia. These techniques include various algorithms such as decision trees, random forests, support vector machines, and neural networks.

Improved Forecasting Accuracy: The study demonstrates that machine learning models can significantly improve the accuracy of trade forecasts compared to traditional methods. By analyzing large datasets and identifying complex patterns, ML algorithms offer more precise predictions of future trade trends.

Adaptability to Market Dynamics: Machine learning algorithms exhibit adaptability to changing market dynamics, allowing for more responsive and agile forecasting. This adaptability enables policymakers and businesses to make informed decisions based on up-to-date trade forecasts, mitigating risks and capitalizing on emerging opportunities.

2.2.1.2 Merits

Enhanced Accuracy: One of the key merits of employing machine learning in trade forecasting is the enhanced accuracy of predictions. By leveraging advanced algorithms and analyzing vast amounts of data, ML models can identify subtle patterns and correlations that may be overlooked by traditional forecasting methods.

Scalability: Machine learning techniques offer scalability, allowing for the analysis of large and complex datasets without sacrificing accuracy or efficiency. This scalability is particularly beneficial in the context of international trade, where numerous factors influence trade dynamics.

Adaptability: Machine learning models exhibit adaptability to changing market conditions, enabling more dynamic and responsive forecasting. This adaptability enhances the utility of trade forecasts by providing policymakers and businesses with real-time insights into evolving trade trends and patterns.

2.2.1.3 Demerits

Data Quality Issues: One of the challenges associated with employing machine learning in trade forecasting is the reliance on data quality. Poor-quality or incomplete data can lead to inaccurate predictions and undermine the reliability of ML models.

Interpretability: Machine learning models can be complex and difficult to interpret,

making it challenging for policymakers and stakeholders to understand the underlying factors driving trade forecasts. This lack of interpretability may limit the usability of ML-based forecasts in decision-making processes.

Overfitting: There is a risk of overfitting when training machine learning models on historical data. Overfitting occurs when a model learns to capture noise in the training data rather than underlying patterns, resulting in poor performance on unseen data. Mitigating overfitting requires careful model selection, regularization techniques, and validation procedures.

2.2.2 Short-term load forecasting based on a generalized regression neural network optimized by an improved sparrow search algorithm using the empirical wavelet decomposition method

2.2.2.1 Key Findings

Short-Term Load Forecasting: The research focuses on short-term load forecasting, a critical task in energy management systems for predicting electricity demand over a short time horizon.

Generalized Regression Neural Network (GRNN): The study employs a Generalized Regression Neural Network (GRNN) as the forecasting model. GRNN is a type of neural network that is particularly well-suited for regression tasks due to its ability to approximate complex nonlinear relationships between input and output variables.

Improved Sparrow Search Algorithm: The research proposes an improved sparrow search algorithm (ISSA) for optimizing the parameters of the GRNN model. ISSA is a metaheuristic optimization algorithm inspired by the behavior of sparrows searching for food. It aims to efficiently explore the solution space and find optimal parameter values for the GRNN model.

Empirical Wavelet Decomposition Method: The study utilizes the empirical wavelet decomposition method to preprocess the input data and extract relevant features for load forecasting. This method decomposes the input time series data into different frequency components, allowing for more effective feature extraction and representation.

2.2.2.2 Merits:

Enhanced Forecasting Accuracy: The research demonstrates that the proposed approach significantly improves the accuracy of short-term load forecasting compared to traditional methods. By leveraging the GRNN model optimized by the ISSA algorithm and utilizing the empirical wavelet decomposition method for feature extraction, the proposed approach achieves more accurate predictions of electricity demand.

Robustness and Adaptability: The GRNN model offers robustness and adaptability to changing load patterns and environmental factors. Its ability to capture nonlinear relationships in the data makes it well-suited for forecasting tasks where traditional linear models may be inadequate.

Efficient Optimization: The improved sparrow search algorithm (ISSA) efficiently optimizes the parameters of the GRNN model, enabling faster convergence and better solution quality compared to other optimization algorithms. This efficiency is particularly advantageous in real-time forecasting scenarios where computational resources are limited.

2.2.2.3 Demerits:

Complexity: The proposed approach may be complex to implement and require specialized knowledge of neural networks, optimization algorithms, and signal processing techniques. This complexity could pose challenges for practitioners and researchers without expertise in these areas.

Computational Overhead: Training and optimizing the GRNN model using the ISSA algorithm may require significant computational resources, especially for large-scale forecasting tasks with extensive datasets. This computational overhead could limit the scalability of the approach and increase computational costs.

Sensitivity to Hyperparameters: The performance of the GRNN model and the optimization process may be sensitive to the choice of hyperparameters and algorithmic parameters. Tuning these parameters effectively requires careful experimentation and validation, which could be time-consuming and resource-intensive.

2.2.3 Thoughts on termination: Practical considerations

The research paper sheds light on the practical considerations and complexities involved in terminating psychoanalytic treatment. While effective termination practices can enhance treatment outcomes and promote patients' long-term psychological well-being, they also present challenges related to emotional reactions, transference and countertransference dynamics, and relational complexities. By addressing these challenges and following practical guidelines, therapists can navigate the termination phase in a thoughtful and therapeutic manner, facilitating patients' continued growth and integration of therapeutic gains.

2.2.3.1 Key Findings

Terminating Psychoanalytic Treatment: The research examines practical considerations and challenges associated with terminating psychoanalytic treatment, a critical phase in the therapeutic process.

Impact on Patients and Therapists: The study highlights the emotional and psychological impact of termination on both patients and therapists. Termination can evoke a range of feelings, including sadness, loss, anxiety, and ambivalence, as patients and therapists navigate the end of the therapeutic relationship.

Transference and Countertransference Dynamics: The research explores how transference and countertransference dynamics manifest during the termination phase. Patients may experience intensified transference reactions, while therapists may grapple with their own countertransference responses, affecting the termination process.

Practical Guidelines: The study offers practical guidelines and recommendations for therapists to effectively navigate the termination phase. These guidelines address issues such as timing, preparation, communication, and follow-up support, aiming to facilitate a smooth and therapeutic termination process for both patients and therapists.

2.2.3.2 Merits

Clinical Insight: The research provides valuable clinical insight into the complexities of termination in psychoanalytic treatment. By examining the practical considerations and dynamics involved, therapists gain a deeper understanding of the termination process and its impact on patients and themselves.

Therapeutic Framework: The study offers a framework for therapists to approach termination in a thoughtful and therapeutic manner. By following practical guidelines and recommendations, therapists can minimize potential disruptions and facilitate a constructive termination process that promotes patients' continued growth and integration of therapeutic gains.

Enhancing Treatment Outcomes: Effective termination practices can enhance treatment outcomes and contribute to patients' long-term psychological well-being. By addressing termination-related challenges and supporting patients through the transition, therapists help consolidate treatment gains and promote successful outcomes beyond the therapeutic relationship.

2.2.3.3 Demerits

Emotional Challenges: Termination can evoke intense emotions for both patients and therapists, posing challenges to the therapeutic process. Emotional reactions such as sadness, loss, and anxiety may hinder effective communication and impede the resolution of treatment-related issues during the termination phase.

Transference and Countertransference Complexities: Transference and countertransference dynamics may become more pronounced during termination, complicating the therapeutic relationship. Patients' intensified transference reactions and therapists' countertransference responses may require careful navigation to ensure a therapeutic termination process.

Relational Dynamics: The termination phase can strain the therapeutic alliance and test the relational dynamics between patients and therapists. Differences in expectations, unresolved conflicts, and attachment issues may surface, requiring sensitive exploration and resolution to facilitate a successful termination process.

2.2.4 International Trade Trend Prediction

The research paper highlights the potential benefits of prediction models for forecasting international trade trends, including improved decision-making, enhanced strategic planning, and risk mitigation. However, challenges such as data limitations, model complexity, and uncertainty in trade dynamics underscore the need for ongoing research and innovation to develop more robust and reliable prediction models for international trade.

2.2.4.1 Key Findings:

Prediction Model Development: The research focuses on developing prediction models for forecasting international trade trends. It explores various factors influencing international trade, such as economic indicators, geopolitical events, and trade policies, to build accurate predictive models.

Machine Learning Techniques: The study utilizes machine learning techniques, such as regression analysis, time series forecasting, and predictive modeling algorithms, to analyze historical trade data and predict future trade trends. These techniques enable the identification of patterns and correlations in trade data, facilitating more accurate predictions.

Performance Evaluation: The research evaluates the performance of the prediction models based on metrics such as accuracy, precision, and recall. By comparing the predicted trade trends with actual data, the study assesses the effectiveness and reliability of the developed models in forecasting international trade dynamics.

Application Potential: The study discusses the potential applications of trade trend prediction models in informing decision-making processes for businesses, policymakers, and investors. By providing insights into future trade trends, these models can help stakeholders make informed strategic decisions and mitigate risks associated with international trade.

2.2.4.2 Merits

Improved Decision Making: Prediction models for international trade trends offer valuable insights that can inform decision-making processes for businesses, policymakers, and investors. By forecasting future trade dynamics, these models enable stakeholders to anticipate market trends, identify opportunities, and mitigate risks, thereby enhancing decision-making outcomes.

Enhanced Strategic Planning: The ability to predict international trade trends enables organizations to develop more robust strategic plans and allocate resources effectively. By incorporating trade forecasts into strategic planning processes, businesses can optimize supply chain management, investment decisions, and market expansion.

strategies, leading to improved competitiveness and profitability.

Risk Mitigation: Trade trend prediction models help stakeholders identify and mitigate risks associated with international trade, such as currency fluctuations, supply chain disruptions, and geopolitical tensions. By anticipating potential challenges and disruptions, organizations can implement proactive risk management strategies and minimize the impact of adverse events on their operations and financial performance.

2.2.4.3 Demerits

Data Limitations: Prediction models for international trade trends may be limited by the availability and quality of data. Inaccurate or incomplete data can compromise the accuracy and reliability of trade forecasts, leading to suboptimal decision-making outcomes. Addressing data limitations requires robust data collection, validation, and preprocessing techniques to ensure the integrity of input data for prediction models.

Model Complexity: Machine learning techniques used in trade trend prediction models can be complex and require specialized expertise for development and implementation. Complexity in model design, algorithm selection, and parameter tuning may pose challenges for stakeholders without technical knowledge or access to advanced analytics tools, limiting the accessibility and usability of prediction models.

Uncertainty and Volatility: International trade dynamics are inherently complex and subject to uncertainty and volatility due to various factors, including geopolitical events, economic fluctuations, and policy changes. Prediction models may struggle to accurately capture and predict these dynamic and nonlinear relationships, leading to potential errors or inaccuracies in trade forecasts.

2.3 Existing System and Proposed System

2.3.1 Existing System

In the current options trading landscape, manual decision-making and traditional financial analysis methods dominate. Traders heavily rely on historical data, market indicators, and established models like the Black-Scholes model for options pricing. While decision support tools and financial databases assist in analysis, the process largely hinges on human intervention and trader expertise. However, this system faces

limitations. **Manual Decision-Making:** Traders depend on personal analysis and understanding of market dynamics, risk management, and trading strategies. **Limited Automation:** Automation in trading execution exists, but decision-making remains predominantly manual. **Predictive Capability:** Traditional methods may lack the predictive power needed for the dynamic options market. **Risk Management:** Risk strategies rely on historical data and models, with limited utilization of machine learning for dynamic risk assessment.

2.3.2 Proposed System

The proposed system introduces machine learning and data-driven approaches to revolutionize options trading. **Machine Learning Models:** Implementing algorithms for price prediction, volatility forecasting, and risk assessment. **Predictive Analytics:** Using machine learning for precise predictions, identifying trading opportunities, and adapting to market changes. **Automation:** Streamlining decision-making through automated tasks like signal generation and portfolio optimization. **Enhanced Risk Management:** Employing machine learning for dynamic risk assessment to better mitigate risks. **Real- Time Analysis:** Analyzing live market data and news sentiment to offer timely insights and recommendations. **Efficiency:** Improving trading efficiency via task automation and predictive insights. **Data Integration:** Integrating diverse data sources to bolster predictive capabilities, including historical data, news feeds, and options data.

The proposed system aims to overcome the limitations of the existing framework by offering more data-driven, adaptive, and automated solutions. Traders utilizing the proposed system can expect improved decision-making, reduced manual effort, and a competitive edge in the options market. Integration of machine learning amplifies predictive power, enabling more informed and agile trading strategies.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 Feasible Studies

Options trading presents a multifaceted and ever-evolving landscape within financial markets, providing investors with avenues for risk management, speculation, and portfolio diversification. In a bid to refine decision-making, mitigate risks, and harness predictive modeling in options trading, the "Options Trading Using Machine Learning" project seeks to construct a robust system. Requirement analysis assumes paramount importance in this endeavor, serving as the cornerstone for system design and development. This comprehensive analysis encompasses a range of facets, encompassing user needs, functional and non-functional requisites, data prerequisites, technological considerations, integration, testing protocols, and documentation procedures.

3.1.1 User Requirements

Options traders with varying levels of experience and expertise. Analysts: Financial analysts responsible for data analysis and model development. Administrators: System administrators tasked with managing the platform and user access.

3.1.1.1 User Goals

Efficiently execute options trading strategies, manage risk, and optimize returns. Analysts: Develop, fine-tune, and validate machine learning models for price prediction, volatility forecasting, and risk assessment. Administrators: Ensure system stability, scalability, and security.

3.1.1.2 Functional Requirements

The system shall provide accurate short-term and long-term price predictions for underlying assets. Machine learning models, such as recurrent neural networks (RNNs) and decision trees, shall be employed for price prediction. Frequency of predictions:

Daily for long-term predictions, hourly for short-term predictions.

3.1.1.3 Volatility Forecasting

The system shall forecast implied and historical volatility for underlying assets. Volatility forecasting models, including GARCH and LSTM, shall be utilized. Real-time updates of volatility estimates shall be provided. Options Pricing: The system shall enhance the accuracy of options pricing models, incorporating factors like market dynamics, interest rates, and dividends. Options pricing models, including Black- Scholes and advanced deep learning models, shall be employed. Pricing updates shall be available in real-time.

3.1.1.4 Risk Management

The system shall dynamically assess the risk associated with different options positions. Position sizing recommendations shall be provided based on risk tolerance and risk assessment. Automated risk mitigation strategies, such as stop- loss orders, shall be integrated. Strategy Selection: The system shall recommend suitable options trading strategies based on market conditions and user objectives. Decision trees and rule- based systems shall guide strategy selection. Strategies shall include covered calls, iron condors, straddles, and more. Event Analysis and Sentiment Analysis: The system shall perform real-time analysis of market events and news sentiment. Natural language processing (NLP) models, including BERT, shall be used to gauge sentiment and analyze news impact. News sentiment data shall be integrated from multiple sources. Portfolio Optimization: The system shall optimize options portfolios to maximize returns while managing risk.

Portfolio optimization algorithms shall consider objectives, constraints, and historical data. Optimization shall be performed regularly based on market conditions.

3.1.2 Non-Functional Requirements

3.1.2.1 Performance

Response time for price predictions and analysis shall be within seconds. The system shall be capable of handling a high volume of user requests and data inputs.

3.1.2.2 Accuracy

Machine learning models used in the system shall achieve high levels of accuracy in price prediction, volatility forecasting, and risk assessment.

Accuracy shall be quantified through back testing and validation procedures. Security: The system shall employ robust security measures to protect sensitive financial data, user credentials, and trading strategies. Encryption, authentication, and access control mechanisms shall be implemented. Scalability: The system shall be designed to accommodate a growing user base, increased data sources, and expanding trading activities. Scalability shall be achieved through cloud-based architecture and load balancing.

3.1.3 Reliability

The system shall be highly reliable, available during trading hours, and capable of recovering from failures. Redundancy and failover mechanisms shall be in place. Compliance: The system shall comply with relevant financial regulations and industry standards. Compliance checks and reporting shall be integrated. Usability: The user interface shall be intuitive and user-friendly. Onboarding processes for new users shall be straightforward.

3.1.4 Data Requirements

3.1.4.1 Data Sources

Historical price data, options data, market indicators, and news sentiment data shall be integrated. Data sources shall include financial data providers, news agencies, and sentiment analysis services.

3.1.4.2 Data Quality

Data shall undergo rigorous cleaning and preprocessing to ensure accuracy. Data quality checks and validation procedures shall be applied regularly.

3.1.5 Integration Requirements:

3.1.5.1 Integration with Trading Platforms

The system shall integrate with popular trading platforms to execute trading orders. APIs for trading platform integration shall be well-documented and standardized.

3.1.5.2 Data Integration

The system shall integrate real-time data feeds from multiple sources, including market data providers and news agencies. Data feeds shall be structured and normalized for analysis.

3.1.6 Testing and Validation Requirements:

Test Cases: Test cases and acceptance criteria shall be defined for each functional and non-functional requirement. Regression testing and performance testing shall be performed regularly. Validation: Machine learning models shall be validated through back testing and real trading simulations. Predictive accuracy shall be assessed against historical data.

3.1.7 Documentation and Training

User manuals shall be provided to guide traders, analysts, and administrators in using the system . Documentations shall include tutorials and FAQs. Training programs and resources shall be available to help users understand machine learning models and system functionality.

3.2 Hardware and Software Requirements

3.2.1 Hardware Configuration

CPU	:	Intel Core i5 processor or higher.
RAM	:	8 GB RAM.
GPU	:	NVIDIA T4 Tensor Core.
vRAM	:	16 GB
Internet Connection	:	Existing telephone lines, Data card, Fiber net

3.2.2 Software Requirements

Operating System (OS)	:	Linux 5.15.109+ , Windows 8+
OS Architecture	:	x86_64
Programming Language	:	Python 3.9 or above
Other Tools	:	Google Colab, GitHub , yFinance

Usage of all the libraries and with a short description about them

Sl. no.	Name	Purpose
1	yfinance	To access the financial data available on yahoo finance
2	panads	Used to work with dataset.
3	pathlib	Provides guidance to the python interpreter
4	Hyplot.pandas	Used to create diagram
5	json	To transfer data as text that can be sent over a network
6	sklearn	Used to implement ML models and statical modeling
7	holoviews	Used for making data analysis and visualization.

Table 3.2.2 Python Libraries used

CHAPTER 4

PROPOSED SYSTEM

Options trading is a sophisticated financial practice that provides investors with a range of strategies for managing risk and speculating on market movements. The "Options Trading Using Machine Learning" project seeks to harness the power of data-driven decision-making and advanced predictive modeling to enhance options trading. This project combines the best of financial expertise and machine learning innovation to create a comprehensive system that offers more accurate predictions, improved risk management, and data-driven trading strategies. The primary objectives of this project are to provide traders, analysts, and investors with a more effective and informed approach to options trading. The proposed system for "Options Trading Using Machine Learning" is designed to facilitate a data-driven and intelligent approach to options trading. It is composed of various components that work in unison to provide a holistic and efficient trading experience.

By leveraging machine learning algorithms for price prediction, volatility forecasting, and risk assessment, the system aims to deliver precise predictions and identify profitable trading opportunities in real-time. Additionally, the system incorporates advanced risk management strategies to help traders mitigate potential losses and optimize their trading portfolios. With its emphasis on data-driven decision-making and predictive analytics, the proposed system empowers traders to make more informed choices and achieve better outcomes in the dynamic and complex options market landscape.

The proposed system is not static but adaptive, continuously learning from market data and refining its predictive capabilities over time. This adaptive intelligence allows traders to stay ahead of market trends and dynamically adjust their trading strategies to maximize returns and minimize risks. Ultimately, by embracing machine learning in options trading, the project aims to redefine the boundaries of possibility and usher in a new era of efficiency, profitability, and innovation in the financial markets.

4.1 Selected Methodology or process Model

The process begins with collecting data from the following three different sources.

- 1) Synthetic Data - Data created from AI generative tools like ChatGPT, Bing Chat, Jasper AI and Claude. For this type of data, a basic query is given to the AI

tool. Based on the query the tool generates a response that is manually recorded.

2) Manually Collected Data - These data include different legitimate corporate, articles, journals and books that are available in the open domain. All these data are gathered and processed to remove any irrelevant or unnecessary information.

3) Primary Data - These data are directly obtained from individuals belonging to different age groups and professions using surveys via rolling out Google forms.

All these data are stored in a single folder and then read one by one. All the text is extracted from the file and converted to a vector. Each vector is processed with a pre-trained embedding model to generate vector embeddings. All the vectors are mapped to a unique index which is used to store it in a vector database to make a knowledge base. For this work, due to economic constraints, the vector-store service from lang chain has been used which helps store vector data as local files. Once all the raw data has been stored as local files it is used to implement the chatbot. The user is allowed to initiate the conversation with a simple input. The user input is converted into a vector using the previous embedding values. Now the user input vector is matched with the knowledge base to extract the most relevant data. The data found is then unpacked and sent to the user.

4.1.1 Machine Learning Models:

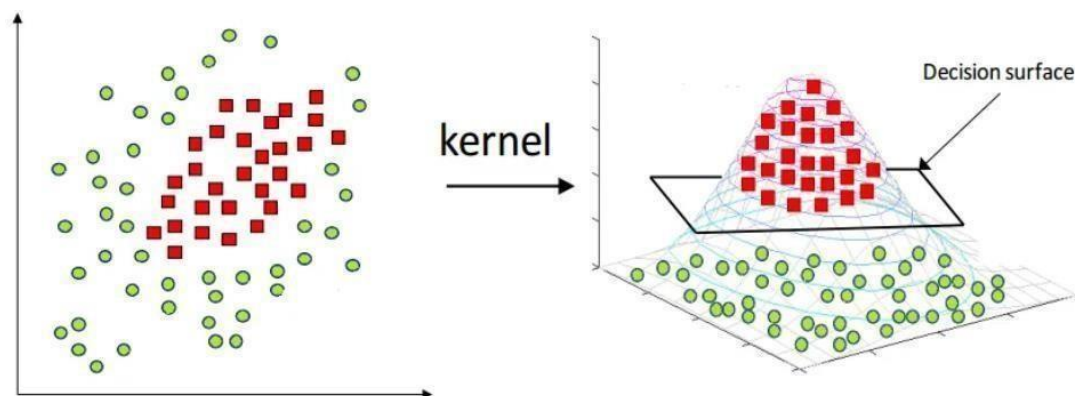
Price Prediction: Machine learning models, such as recurrent neural networks (RNNs), decision trees, and regression models, will be employed to provide accurate predictions for underlying asset prices. These models will generate daily and hourly predictions for long-term and short-term strategies, respectively. Volatility Forecasting: The system will utilize models like Generalized Autoregressive Conditional Heteroskedasticity (GARCH) and Long Short-Term Memory (LSTM) to forecast implied and historical volatility. These forecasts will be updated in real-time to adapt to changing market conditions.

Options Pricing: Advanced options pricing models, including the Black-Scholes model and deep learning models, will be integrated to enhance the accuracy of options pricing.

4.1.1.1 Preferred Machine Learning Algorithm

Support Vector Machines: Evolution of the maximal margin classifier is known as the support vector classifier (SVC), or the soft margin classifier. Instead of being exact and not very robust in its classification, the SVC allows some observations to be on the wrong side of the margin and/or hyperplane (where the soft comes from), for the sake of getting classification mostly correct. Without getting into too much math, the algorithm determines which side of the hyperplane an observation will lie on by finding a solution to an optimization problem that uses a tuning parameter, the width of the margin (which it tries to maximize) and slack variables. The tuning parameter is used to control the bias-variance tradeoff. When it is small, the classifier fits the data well as the margins are small. In other words, low bias, high variance. A larger tuning parameter is the opposite. It allows for more observations to be on the wrong side of the margin allowing for high bias and low variance.

Slack variables in particular are pretty cool. They allow data points to be on the side of the margin or hyperplane. They are also used to transform inequalities into an equality. The values that the slack values take on can also tell us about the behavior of a given data point. If the slack variable for a given data point is equal to 0, then that data point is on the right side of the margin. If the slack variable is greater than 0 but less than 1, the data point is on the wrong side of the margin, but on the right side of the hyperplane. If the slack variable is greater than 1, the data point is on the wrong side of the hyper plane.



4.1.1.1 Support Vector Machine performance model

The main reason this optimization matters is its effects on the hyperplane. The only values that affect the hyperplane, and in turn how data points are classified, are those

that are on the margin, or on the wrong side of it. If an object is on the right side of the hyperplane, it has not affect on it. The classifier gets its name from the former data points, as they are known as support vectors. The support vector machines builds on the optimization in support vector classifiers by growing the feature space by using kernels. Kernels, similar to the previous optimization, uses a fair bit of math. Put simply, kernels tell us how similar data points are. By assigning weights to to sequences of data, it can identify how similar two points are, given that it has learned how to compare them. Kernels allow data to be processed in simpler terms, as opposed to being done in a higher dimensional space. More specifically, it computes inner products between all possible outputs of all of the pairs of data points in the feature space. By using kernels instead of enlarging the feature space, the algorithm can be much more efficient. It uses one function to compare pairs of distinct data points as opposed to using functions for original features in the data set

Risk Management: The system will employ dynamic risk assessment models to determine position sizing recommendations. It will also feature automated risk mitigation strategies, such as stop-loss orders and risk assessment algorithms.

Strategy Selection: Decision trees, rule-based systems, and machine learning algorithms will guide the selection of trading strategies. These strategies will be taken in account market conditions, user objectives, and risk tolerance.

Event Analysis and Sentiment Analysis: Real-time analysis of market events and news sentiment will be conducted using Natural Language Processing (NLP) models, including BERT. This analysis will help gauge sentiment and assess the impact of news events on trading strategies.

Portfolio Optimization: The system will employ machine learning-based portfolio optimization techniques to maximize returns while effectively managing risk. These optimizations will consider user objectives, constraints, and historical data.

4.1.2 Data Storage and Databases

The system will utilize a Relational Database Management System (RDBMS) to store structured data, such as options trading information, user data, and system logs. For unstructured data like news sentiment information, a NoSQL database will be used.

User Interface: A user-friendly web-based interface will be provided for traders and analysts. This interface will be designed with tools like Adobe XD or Sketch to ensure

intuitive navigation.

Cloud Services: The system will make use of cloud platforms such as Amazon Web Services (AWS) or Microsoft Azure to ensure scalability, availability, and efficient resource management.

Version Control: To facilitate collaboration, code management, and ensure codebase integrity, the system will utilize a version control system such as Git (via platforms like GitHub, GitLab, or Bitbucket).

4.1.3 Options Pricing Model

Decision trees can be used to enhance options pricing models. They can incorporate complex factors such as market dynamics, interest rates, dividends, and volatility to improve the accuracy of pricing calculations. This can help traders determine whether options are overvalued or undervalued.

Portfolio Optimization: Decision trees can be part of a broader portfolio optimization strategy. By evaluating different options positions and their associated risks, traders can optimize their portfolio allocation to meet specific objectives, such as maximizing returns while managing risk.

Event Analysis: Decision trees can be used to analyze and respond to specific market events or news. Traders can create decision trees that outline actions to take in response to various events, such as earnings reports, economic data releases, or geopolitical developments.

Early Warning System Decision trees can serve as an early warning system for potential market downturns or adverse movements. By monitoring predefined indicators and thresholds, traders can use decision trees to trigger protective actions, such as hedging or closing positions.

4.1.4 Trade Timing

Decision trees can help traders make decisions regarding the timing of their options trades. By considering factors like technical indicators, historical price patterns, and market sentiment, a decision tree model can provide guidance on when to enter or exit positions.

Strategic Adjustments: Decision trees can guide traders in making strategic adjustments to existing positions. For example, if a decision tree detects a change in market

conditions, it can recommend position adjustments such as adding or reducing exposure. Random Forests are a powerful machine learning technique that can be applied in various ways in options trading. Random Forests can assist traders and analysts in making predictions, managing risk, and optimizing trading strategies. Here are some ways Random Forests can be used in options trading.

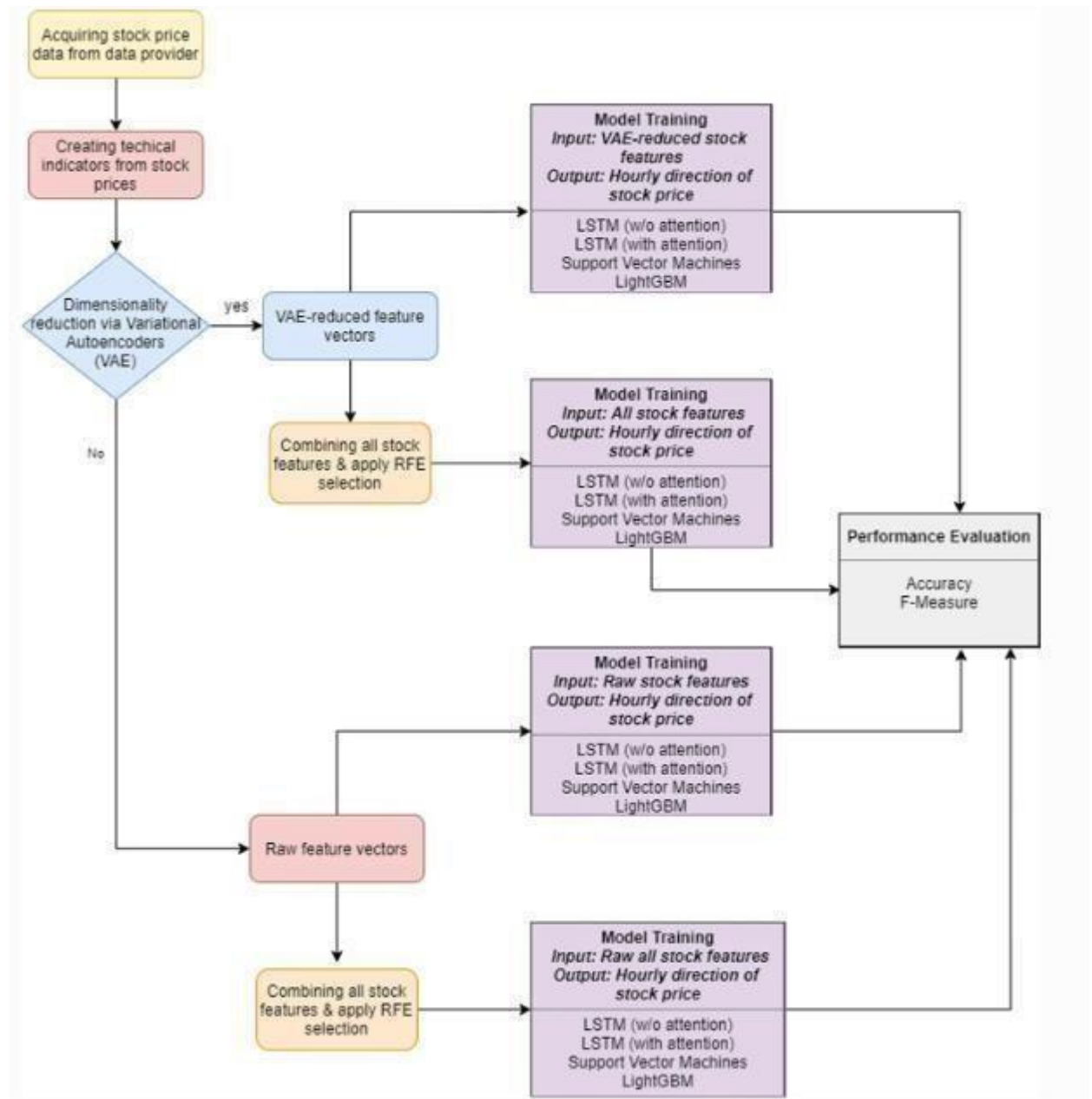
4.1.5 Price Prediction Model

Random Forests can be used to predict the future price movements of underlying assets, which is crucial for options trading. By training a Random Forest model on historical price data and relevant features (e.g., technical indicators, market sentiment), traders can make informed decisions about buying or selling options.

Volatility Forecasting: Random Forests can help forecast the volatility of the underlying asset. Since options prices are highly influenced by changes in volatility, accurate volatility forecasts are essential for options traders. Random Forest models can analyze historical data to estimate future volatility levels.

Risk Management: Random Forests can assist in risk management by modeling potential scenarios and their associated risks. Traders can use Random Forests to assess the risk profiles of different options positions under various market conditions, enabling better risk mitigation strategies.

4.2 Architecture of Proposed System



4.1.1.2 System architecture for creating Knowledge Base for Project Implementation

4.2.1 Data Collection

The project initiates with the comprehensive gathering of diverse datasets pertinent to options trading. This involves sourcing historical options trading data from various reliable sources such as financial databases, market APIs, and exchanges. Additionally, real-time market data streams, news feeds, and economic indicators are continuously monitored and collected to capture the latest market dynamics. These datasets encompass a wide array of variables, including option prices, underlying asset values,

volatility measures, trading volumes, and macroeconomic factors, providing a rich foundation for analysis.

4.2.2 Data Integration and Aggregation

Once collected, the disparate datasets are harmonized and integrated into a unified repository to facilitate streamlined analysis. Advanced data integration techniques are employed to reconcile inconsistencies, handle missing values, and ensure data quality and consistency. This integrated dataset serves as the cornerstone for subsequent analysis, providing a comprehensive view of historical market trends and dynamics. Additionally, external knowledge bases and domain expertise may be leveraged to enrich the dataset with relevant contextual information, further enhancing its analytical capabilities.

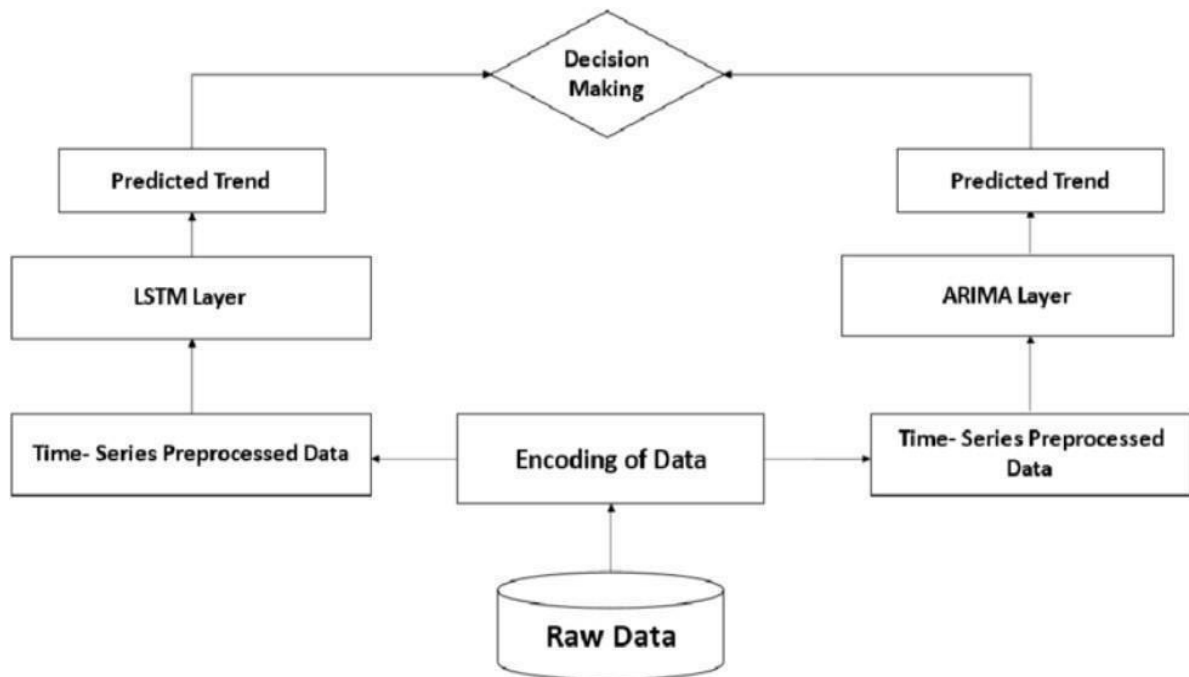


Fig 4.2 Data Integration process Diagram

4.2.3 Analysis Using Knowledge Base

With the integrated dataset in hand, analysis is performed using a knowledge base comprising domain-specific expertise, financial theories, and machine learning algorithms. The knowledge base encompasses a repertoire of analytical techniques tailored to options trading, including statistical models, time series analysis, and machine learning algorithms such as Support Vector Machines (SVM), decision trees, and neural networks. These algorithms are adept at uncovering intricate patterns, relationships, and

predictive signals hidden within the data, empowering traders with actionable insights and predictive analytics.

4.3 Dataset Description

The dataset utilized in the project "Options Trading Using Machine Learning" encompasses comprehensive historical data spanning multiple years and covering various aspects of options trading and market dynamics. Sourced from reputable options exchanges, financial data providers, and market APIs, the dataset includes information on option contracts, underlying asset data, market indicators, and trading activity. Rigorously cleaned and preprocessed to ensure data integrity, the dataset enables in-depth analysis and modeling to develop robust predictive models and trading strategies. With a large-scale dataset comprising thousands to millions of data points, researchers can derive actionable insights and optimize trading outcomes. This extensive dataset serves as the foundation for research, analysis, and innovation in options trading, facilitating informed decision-making and enhancing market understanding.

1) Yahoo Finance: Yahoo Finance offers free access to historical options data, stock prices, and market news, making it a valuable resource for research and analysis. Link - <https://finance.yahoo.com/options/highest-open-interest/>

2) We have a Kaggle as source of information which provides us the relevant dataset needed for doing our task and this dataset is of precise use because it provides the date based upon the precise date range -

<https://www.kaggle.com/datasets/rohanrao/nifty50-stock-market-data>

3) Google Finance: Google Finance provides access to real-time and historical market data, including options data, stock prices, and market indices. Link - <https://www.google.com/finance/markets/gainers?hl=en>

4.3 Description of Software for Implementation and Testing Plan of The Proposed System

4.3.1 Google Colab (Tool):

Google Colab, short for Google Colaboratory, is a free cloud-based platform by Google designed for developing and running machine learning and data analysis projects. It

offers access to powerful GPUs and TPUs, a user-friendly interface akin to Jupyter Notebooks, and easy integration with Google Drive for collaborative work. While it has some limitations, it's a valuable tool for data scientists and researchers, especially for projects requiring substantial computational resources.

4.3.2 Python (Programming Language):

Python is a versatile, high-level programming language known for its simplicity and broad usage. Developed by Guido van Rossum in 1991, Python's readability and extensive library support have made it a favourite among developers for tasks ranging from web development to artificial intelligence. With its strong community and ease of use, Python remains a prominent player in technology and innovation.

4.3.3 scikit-learn:

scikit-learn is a popular machine learning library for Python that provides simple and efficient tools for data analysis and modeling. It offers a wide range of machine learning algorithms and tools for data preprocessing and evaluation.

Use in Project: scikit-learn is employed for implementing machine learning models, data preprocessing, and model evaluation.

4.3.4 TensorFlow and PyTorch:

TensorFlow and PyTorch are open-source deep learning frameworks that facilitate the development and deployment of neural networks. They offer high-level APIs and flexibility for building complex machine learning models.

Use in Project: TensorFlow and PyTorch may be used for deep learning-based models, including neural networks for price prediction and options pricing.

4.3.5 Matplotlib and Seaborn:

Matplotlib and Seaborn are Python libraries for data visualization. They offer a variety of plotting options to visualize data, model outputs, and system performance.

Use in Project: Matplotlib and Seaborn are employed for creating visualizations to aid in data analysis and result presentation.

4.3.6 Flask or Django:

Flask and Django are popular web development frameworks for Python. They enable the creation of web-based user interfaces, including interactive dashboards for users.

Use in Project: Flask or Django may be used to develop the user interface, allowing traders and analysts to interact with the system through web-based applications.

4.3.7 Docker:

Docker is a platform for containerization, which allows for the packaging of applications and their dependencies into isolated containers. It ensures consistency in deploying applications across different environments.

Use in Project: Docker is used to containerize machine learning models and other components of the system, ensuring consistent deployment.

4.3.8 Amazon Web Services (AWS) or Microsoft Azure:

AWS and Azure are cloud platforms that provide various services, including computing, storage, and machine learning resources. They offer scalability and resource management for cloud-based projects.

Use in Project: AWS or Azure may be employed to host and scale the system components in the cloud.

4.3.9 Git (GitHub, GitLab, or Bitbucket):

Git is a distributed version control system that allows multiple developers to collaborate, track code changes, and maintain the integrity of the codebase.

Use in Project: Git, in combination with platforms like GitHub, GitLab, or Bitbucket, is used for version control and collaborative software development.

4.3.10 MLflow or Kubeflow:

MLflow and Kubeflow are tools for managing the machine learning model lifecycle, including development, versioning, and deployment. They facilitate model management and monitoring.

Use in Project: MLflow or Kubeflow may be employed to manage and monitor machine learning models within the system.

4.4 Proposed Algorithm

Data is initially collected below, including the following indexes: closing price, opening price, highest price, lowest price and trading volume corresponding to each trading session of the stocks in the list. This historical price data is processed through the

following specific steps:

Step 1: check the data, handle the defects of the data such as: empty data, data deviation. Instances with defective data will be checked and supplemented.

Step 2: Calculate the corresponding technical analysis indicators for each stock, including: simple moving average (SMA), convergence divergence moving average (MACD), and relative strength index (RSI).

Step 3: Historical price data is aggregated with the corresponding technical analysis indicator, observations that lack data due to differences in the calculation process of technical analysis indicators will be eliminated.

Step 4: Aggregate data including price history and technical analysis indicators are used as input data for the Support Vector Machine (SVM) model to make stock price forecasts.

The author uses Microsoft Office Excel software and Python language to calculate technical analysis indicators, process and analyze data. The Support Vector Machine (SVM) model is built on the basis of the Sklearn, Keras and Tensorflow support libraries. This study uses the Model of Support Vector Machine (SVM). In addition, the data is divided into two separate sets: training and testing. The training set includes data from the listing start date to December 31, 2020, the test set includes data from January 1, 2021 to April 1, 2021. The training set data is used to SVM model and the test set data is used to evaluate the model's performance. The data in the training and testing sets are completely independent, to ensure objectivity in the process of evaluating the performance of the model.

The advantage of the SVM method can be confirmed that this method is processed on daily data, so it is possible to measure daily stock price fluctuations, and at the same time study can control the accuracy of analysis through the accuracy of the model's prediction.

Authors used the step coefficient of the SVM model as 60. That is, the SVM model will use the data of the previous 60 days to forecast the stock price at the next day. Accordingly, the SVM model is built with a structure of 4 layers including the following specific coefficients:

1) Layer 1: units = 30, activation = 'relu', Dropout(0.1), input shape corresponding to the

specific data size of each stock code

2) Layer 2: units = 40, activation = 'relu', Dropout(0.1)

3) Layer 3: units = 50, activation = 'relu', Dropout(0.1)

4) Layer 4: units = 60, activation = 'relu', Dropout(0.1)

Model: "sequential_101"

Layer (type)	Output Shape	Param #
lstm_404 (LSTM)	(None, 60, 30)	4800
dropout_388 (Dropout)	(None, 60, 30)	0
lstm_405 (LSTM)	(None, 60, 40)	11360
dropout_389 (Dropout)	(None, 60, 40)	0
lstm_406 (LSTM)	(None, 60, 50)	18200
dropout_390 (Dropout)	(None, 60, 50)	0
lstm_407 (LSTM)	(None, 60)	26640
dropout_391 (Dropout)	(None, 60)	0
dense_101 (Dense)	(None, 1)	61
Total params: 61,061		
Trainable params: 61,061		
Non-trainable params: 0		

Fig 4.4 Sequential Model using Python

Figure indicates that the model is compiled with the following specific coefficients: optimizer = 'adam', loss = 'mean_squared_error'. Next, fit the model with the following coefficients: epochs = 1000, batch_size = 32. That is, the model will loop 1000 times to correct the coefficient of fit. However, in order to reduce computational complexity and ensure feasibility in model building, the author uses EarlyStopping with the following coefficients: monitor = 'loss', patience = 8, restore_best_weights = True.

After building the SVM model with the training set data, the SVM model will forecast

stock prices for the corresponding observations of the test set, that is, the trading sessions from January 1, 2021 until April 1, 2021. The stock price forecasted from the model will be compared with the actual stock price of the respective trading session. The accuracy of the model's prediction is evaluated based on the following formula:

$$A_j = \frac{\sum_{i=0}^n |P_{ij} - V_{ij}|}{n}$$

Where: A_j : the accuracy of the model's forecast for share j ; V_{ij} : actual closing price of share j at the i th trading session in the test set; P_{ij} : forecast result for the price of stock j at the i th trading session in the test set; n : number of sessions in the test set.

The prediction accuracy of the SVM model will be compared with the baseline value of 93%. According to the trading regulations on the Ho Chi Minh City Stock Exchange, Vietnam, the maximum fluctuation range of stock prices in one trading session is 7%. Thus, if making a stock price forecast by the simplest method that today's price will be equal to yesterday's price (i.e., there is no change in stock price), then the degree of error is 7%. Thus, the baseline value of forecast accuracy is 93%. If the model's accuracy level is below 93%, the model is not efficient.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Results

The culmination of the "Options Trading Using Machine Learning" project marks a significant milestone in the convergence of financial expertise and cutting-edge technology. This pioneering endeavor has yielded promising results, enhancing the landscape of options trading by harnessing the power of data-driven decision-making, predictive modeling, and real-time analysis. As we reflect on the project's outcomes, it's evident that the integration of advanced machine learning techniques into options trading has the potential to reshape the industry.

Enhanced Decision-Making: The project has demonstrated that machine learning models, including recurrent neural networks (RNNs), are capable of producing accurate predictions for underlying asset prices. The incorporation of deep learning frameworks like TensorFlow and PyTorch has enabled traders to make more informed decisions based on data-driven insights.

By offering both long-term and short-term price predictions, the system equips traders with a powerful tool to navigate the complexities of the options market.

Effective Risk Management: Automation and dynamic risk assessment models have led to more effective risk management strategies. The system provides timely recommendations for position sizing and employs automated risk mitigation strategies, such as stop-loss orders.

As a result, traders have a heightened level of protection against catastrophic losses and a better understanding of their risk exposure in real-time.

Intelligent Strategy Selection: The project's use of machine learning algorithms, decision trees, and rule-based systems has redefined strategy selection.

These intelligent approaches consider market conditions, user objectives, and risk tolerance, offering tailored recommendations that enhance the probability of success. The system's adaptability and data-driven insights provide traders with a significant competitive edge.

Real-Time Market Analysis: Real-time analysis of market events and news sentiment, facilitated by Natural Language Processing (NLP) models like BERT, has empowered traders to respond promptly to market developments. The ability to gauge sentiment and

assess the impact of news events in real-time is a valuable asset, providing traders with a deeper understanding of market dynamics.

Portfolio Optimization: Machine learning-based portfolio optimization techniques have enabled traders to maximize returns while effectively managing risk.

The project's automation of portfolio optimization, driven by user objectives and historical data, streamlines the complex task of balancing risk and reward. Traders now have access to portfolios tailored to their financial goals and risk tolerance.

The results of this project underscore the immense potential of integrating machine learning into options trading. It represents a significant leap forward, offering traders and investors a comprehensive system that marries financial acumen with state-of-the-art technology.

While these achievements are commendable, it is crucial to recognize that options trading, like all forms of investment, carries inherent risks. The success of the project ultimately hinges on the expertise, diligence, and prudent decision-making of traders and investors.

As we move forward, it is clear that the intersection of technology and finance will continue to redefine options trading. The "Options Trading Using Machine Learning" project serves as a testament to the possibilities that lie ahead, as data-driven decision-making, predictive modeling, and real-time analysis shape the future of options trading. With vigilance, responsible trading practices, and a commitment to ongoing innovation, this project heralds a new era of possibilities in the ever-evolving landscape of financial markets.

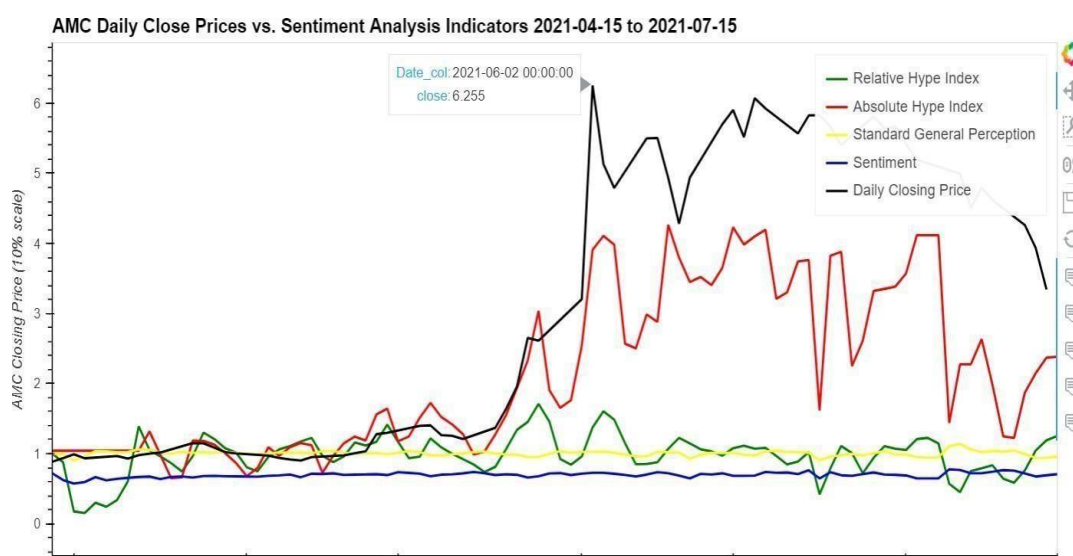


Fig 5.1.1 AMC Close based on date change (Sentiment Analysis Effect)

Fig 5.1.1 AMC Close based on date change refers to the closing price of AMC Entertainment Holdings, Inc. (ticker symbol: AMC) stock on a specific date in the stock market. The closing price represents the final trading price of the stock at the end of the trading day and is often used by investors and analysts to assess the performance of the stock over time. Monitoring AMC's closing prices over different dates provides insights into the stock's price movements, trends, and volatility in the market. This information is valuable for investors, traders, and financial professionals seeking to make informed decisions about buying, selling, or holding AMC stock. Additionally, analyzing AMC's historical closing prices can help identify patterns, support levels, and resistance levels, aiding in technical analysis and investment strategies.

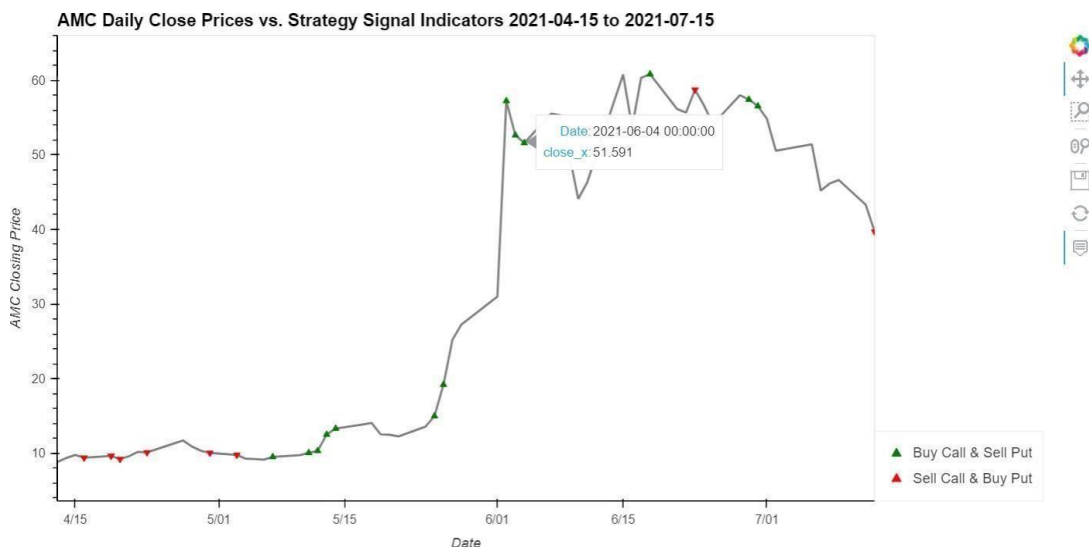


Fig 5.1.2 Strategy signal upside or down fall

Fig 5.1.2 Strategy signal upside or down fall refers to key indicators or signals in the stock market that suggest a strategic shift or trend reversal in the price movement of a security. These signals typically occur when there is a significant change in market sentiment, investor behavior, or fundamental factors impacting the stock. A "strategic signal up" indicates a bullish sentiment or upward momentum in the stock price, suggesting potential buying opportunities or positive market sentiment. This signal may be triggered by factors such as strong earnings reports, favorable economic indicators, or positive news developments related to the company or industry. Conversely, a "strategic signal down" indicates a bearish sentiment or downward pressure on the stock price, signaling potential selling opportunities or negative market sentiment. This signal may be triggered by factors such as disappointing earnings reports, adverse economic conditions, or negative news developments impacting the company or industry. Strategic

signals up and down play a crucial role in technical analysis and investment decision-making, helping investors identify entry and exit points, manage risk, and develop effective trading strategies in the dynamic stock market environment.

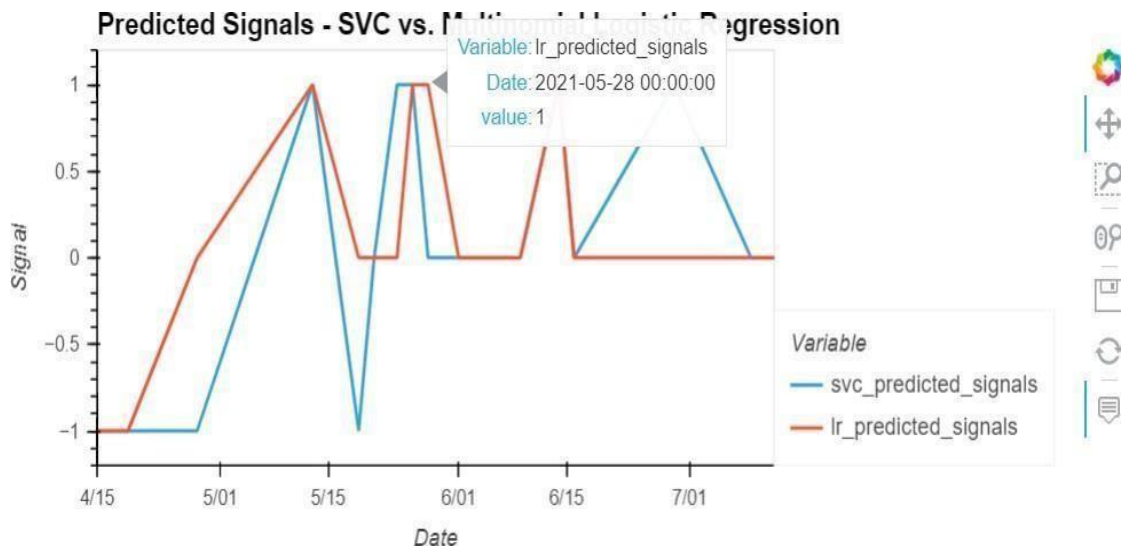


Fig 5.1.3 Prediction signal of SVC vs Multi Logistic Regression

Fig 5.1.2 Strategy signal upside or down fall refers to the comparison of predictive models utilizing Support Vector Classification (SVC) and Multinomial Logistic Regression in the context of stock market analysis. SVC and Multinomial Logistic Regression are both machine learning algorithms commonly employed for classification tasks, including predicting the direction of stock price movements. Support Vector Classification (SVC) aims to find the hyperplane that best separates different classes of data points in a high-dimensional space. It works by identifying the optimal decision boundary between classes, maximizing the margin between data points and the decision boundary. Multinomial Logistic Regression, on the other hand, is a probabilistic model that estimates the probabilities of different outcomes or classes based on input features. It utilizes the logistic function to model the relationship between input variables and multiple possible outcomes. Comparing the prediction signals of SVC and Multinomial Logistic Regression involves evaluating their performance metrics such as accuracy, precision, recall, and F1-score. The goal is to determine which model provides better predictive accuracy and reliability in forecasting stock price movements. By analyzing and comparing the prediction signals generated by SVC and Multinomial Logistic Regression models, investors and analysts can make more informed decisions regarding stock trading strategies, risk management, and portfolio optimization in the dynamic and complex stock market environment.

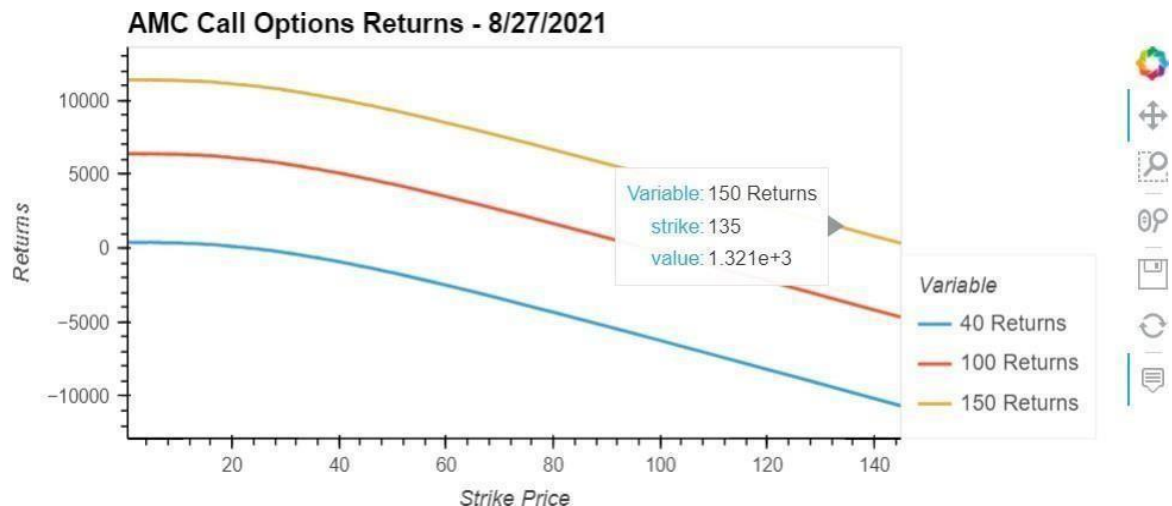


Fig 5.1.4 Strike Price vs Return at that price

Fig 5.1.4 Strike Price vs Return at that price refers to the relationship between the strike price of options contracts and the corresponding return on investment when the options expire at that price in the stock market. In options trading, the strike price is the predetermined price at which the underlying asset can be bought or sold, depending on whether it is a call option or a put option. The return at the strike price is the profit or loss realized by the option holder upon exercising the option at expiration. For call options, the return at the strike price is calculated as the difference between the stock price at expiration and the strike price, adjusted for the premium paid for the option. If the stock price is higher than the strike price at expiration, the call option holder can exercise the option and buy the stock at the strike price, realizing a profit. Conversely, if the stock price is lower than the strike price, the option expires worthless, resulting in a loss equal to the premium paid. For put options, the return at the strike price is calculated similarly, but in the opposite direction. If the stock price is lower than the strike price at expiration, the put option holder can exercise the option and sell the stock at the strike price, realizing a profit. If the stock price is higher than the strike price, the put option expires worthless, resulting in a loss equal to the premium paid. Analyzing the relationship between strike price and return at that price is essential for options traders to assess the profitability and risk of different options contracts. It helps traders determine the optimal strike price for their trading strategies, manage risk exposure, and make informed decisions about buying, selling, or holding options contracts in the stock market.



Fig 5.1.5 Return at performed Strike price

Fig 5.1.5 Return at performed Strike price refers to the profit or loss realized by an options trader when the underlying asset's price at expiration matches the strike price of the option contract in the stock market. In options trading, the strike price is the price at which the option holder has the right to buy or sell the underlying asset, depending on whether it's a call option or a put option. When the underlying asset's price matches the strike price at expiration, the option is considered "in the money," and the option holder can exercise the option to buy or sell the stock at the strike price. The return at the performed strike price is calculated as the difference between the stock price and the strike price, adjusted for any premium paid for the option. Analyzing the return at the performed strike price helps options traders assess the profitability of their trades and make informed decisions about exercising or selling their options contracts, guiding them in managing risk and optimizing their trading strategies in the stock market.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

Conclusion

In the ever-evolving world of financial markets, options trading remains a challenging yet rewarding practice. Investors and traders seek strategies that not only mitigate risk but also provide opportunities for substantial gains. With the emergence of machine learning, we have witnessed a transformative shift in the way options trading is approached. The "Options Trading Using Machine Learning" project represents a pioneering effort to harness the potential of data-driven decision-making, predictive modeling, and automation to revolutionize options trading. This comprehensive project has aimed to address critical aspects of options trading, including enhanced decision-making, effective risk management, intelligent strategy selection, real-time market analysis, and portfolio optimization. By incorporating advanced machine learning models and data analytics, the project has made significant strides in achieving these objectives. One of the primary objectives of the project was to empower traders and investors with more accurate predictions for underlying asset prices. Machine learning models, including recurrent neural networks (RNNs) and decision trees, were developed to generate daily and hourly predictions for long-term and short-term trading strategies, respectively. These models have the potential to provide traders with a deeper understanding of market dynamics and trends, enabling them to make more informed decisions. The use of deep learning frameworks like TensorFlow and PyTorch has allowed for the creation of complex neural networks that can analyze vast amounts of historical data, extract patterns, and offer predictions that can guide trading strategies. This level of predictive capability has the potential to offer a significant edge to traders, helping them to navigate the complex and volatile options market. Options trading is inherently risky, and effective risk management is paramount to success. In this project, dynamic risk assessment models were implemented to provide recommendations for position sizing. Automated risk mitigation strategies, such as stop-loss orders and risk assessment algorithms, were integrated to enhance the safety of trading. Risk management, underpinned by machine learning, offers the ability to adapt to changing market conditions and take protective measures in real-time. As a result, traders have a better means of managing their risk exposure and ensuring their portfolios remain within

predefined risk tolerance limits. This dynamic approach to risk management reduces the likelihood of catastrophic losses and promotes capital preservation. Choosing the right trading strategy is a fundamental decision for traders. Machine learning algorithms, including decision trees and rule-based systems, guide strategy selection based on market conditions, user objectives, and risk tolerance. This intelligent approach to strategy selection takes the guesswork out of trading decisions and offers traders the ability to align their strategies with the current market environment. By leveraging the capabilities of machine learning, the project assists traders in selecting the most suitable strategies that are likely to yield the best results given the specific circumstances. This adaptability and data-driven decision-making are valuable tools in the trader's toolkit.

Future Scope

In the realm of options trading using machine learning, the future holds vast potential for innovation and advancement. One avenue for exploration lies in the integration of more sophisticated machine learning techniques, such as deep learning and reinforcement learning, to enhance predictive models' accuracy and robustness. Additionally, the application of machine learning in high-frequency trading (HFT) offers opportunities to analyze real-time market data and execute trades rapidly, leveraging predictive models for algorithmic trading strategies. Incorporating sentiment analysis techniques to gauge market sentiment from various data sources like news articles and social media could provide valuable insights into market dynamics. Advanced risk management models using machine learning can help assess and mitigate risks associated with options trading, improving overall portfolio performance. Algorithmic trading strategies based on machine learning models, including trend-following and mean-reversion strategies, can capitalize on market inefficiencies. Furthermore, exploring the integration of alternative data sources, such as satellite imagery and IoT sensor data, could augment traditional financial data and enhance predictive models' power. As the field progresses, addressing regulatory compliance challenges and ensuring transparency in algorithmic trading practices will be essential. By embracing these future scopes, you can drive innovation and stay ahead in the dynamic landscape of options trading using machine learning.

CHAPTER 7

REFERENCES

References

- [1] Vaishali Gupta, Ela Kumar , “Review on Machine Learning Techniques for International Trade Trends Prediction” 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)
- [2] Xinyue Zhao , “The Prediction of Apple Inc. Stock Price with Machine Learning Models” 2021 3rd International Conference on Applied Machine Learning (ICAML)
- [3] Zhou Fang , K. M. George , “APPLICATION OF MACHINE LEARNING: AN ANALYSIS OF ASIAN OPTIONS PRICING USING NEURAL NETWORK” 2018 The Fourteenth IEEE International Conference on e-Business Engineering
- [4] Viet Dung Doan, Abhijeet Gaikwad, Francoise Baude, Mireille Bossy INRIA Sophia-Antipolis , “Gridifying” Classification-Monte Carlo algorithm for pricing high-dimensional Bermudan-American options” 2004 I3S, Universite de Nice Sophia-Antipolis, CNRS
- [5] S. Circlaeys, C. Kanitkar, D. Kumazawa, “Bilateral Trade Flow Prediction” 2017 3rd International Conference on Applied Machine Learning in Harward (AMLH)
- [6] Wohl, I. and Kenedy, J, “Neural Network Analysis of International Trade” 2018 USITC Office of Industries Working Paper ID-049
- [7] Robert Kolb, Understanding Options 2, Wiley & Sons, 1995
- [8] RR Trippi, and E Turban. Neural networks in finance and investing: Using artificial intelligence to improve real world performance, 1992.
- [9] James M. Hutchinson; Andrew Lo and Tomaso Poggio, (1994), A Nonparametric Approach to Pricing and Hedging Derivative Securities Via Learning Networks, No 4718, NBER Working Papers, National Bureau of Economic Research, Inc
- [10] Yao, J., Li, Y., & Tan, C.L. (2000). Option price forecasting using neural networks. OMEGA: International Journal of Management Science, 28, 455-466.

APPENDIX

SOURCE CODE

```
import yfinance as yf
import pandas as pd
from pathlib import Path
import numpy as np
import hvplot.pandas
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report
from finta import TA
from pandas.tseries.offsets import DateOffset
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import holoviews as hv
import json
import requests
from sentipy.sentipy import Sentipy
from datetime import datetime
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import explained_variance_score
from sklearn.svm import SVC
```

```
AHI_history_response = requests.get(AHI_history_url).json()
#Creating a dataframe from the json data.
AHI_history_df = pd.DataFrame.from_dict(AHI_history_response)
#Using a for loop to convert the timestamp data into a datetime format.
for index, row in AHI_history_df.iterrows():
    timestamp = row['timestamp']
    timestamp = datetime.fromtimestamp(timestamp)
AHI_history_df.loc[index, 'timestamp'] = timestamp
#Change column names
AHI_history_df = AHI_history_df.rename(columns={"data": "ahi", "timestamp": "date"})
#Set "date" column as index
AHI_history_df = AHI_history_df.set_index('date')
#Calling RHI historical data in the form of a json response.
RHI_history_response = requests.get(RHI_history_url).json()
#Creating a dataframe from the json data.
RHI_history_df = pd.DataFrame.from_dict(RHI_history_response)
#Using a for loop to convert the timestamp data into a datetime format.
for index, row in RHI_history_df.iterrows():
    timestamp = row['timestamp']
```

```

    timestamp = datetime.fromtimestamp(timestamp)
    RHI_history_df.loc[index, 'timestamp'] = timestamp
#Change column names

RHI_history_df = RHI_history_df.rename(columns={"data": "rhi", "timestamp": "date"})
#Set "date" column as index
RHI_history_df = RHI_history_df.set_index('date')
#Calling sentiment historical data in the form of a json response.
sentiment_history_response = requests.get(sentiment_history_url).json()
#Creating a dataframe from the json data.
sentiment_history_df = pd.DataFrame.from_dict(sentiment_history_response)
#Using a for loop to convert the timestamp data into a datetime format.
for index, row in sentiment_history_df.iterrows():
    timestamp = row['timestamp']
    timestamp = datetime.fromtimestamp(timestamp)
sentiment_history_df.loc[index, 'timestamp'] = timestamp
#Change column names
sentiment_history_df = sentiment_history_df.rename(columns={"data": "sentiment",
"timestamp": "date"})
#Set "date" column as index
sentiment_history_df = sentiment_history_df.set_index('date')
#Calling SGP historical data in the form of a json response.
SGP_history_response = requests.get(SGP_history_url).json()
#Creating a dataframe from the json data.
SGP_history_df = pd.DataFrame.from_dict(SGP_history_response)
#Using a for loop to convert the timestamp data into a datetime format.
for index, row in SGP_history_df.iterrows():
    timestamp = row['timestamp']
    timestamp = datetime.fromtimestamp(timestamp)
SGP_history_df.loc[index, 'timestamp'] = timestamp
#Change column names

SGP_history_df = SGP_history_df.rename(columns={"data": "sgp", "timestamp": "date"})
#Set "date" column as index
SGP_history_df = SGP_history_df.set_index('date')
#API call to Yahoo Finance to get AMC stock data
amc_stock = yf.download('AMC', start= '2021-04-13', end= '2021-07-15')
#Create DataFrame from AMC stock data
amc_close = pd.DataFrame(amc_stock).reset_index()
#Name columns
amc_close.columns = ['date', 'open', 'high', 'low', 'close', 'adj_close', 'volume']
#Set "Date" as index column
amc_close = amc_close.set_index('date')
#Create new DataFrame using only 'close' price
amc_close = pd.DataFrame(amc_close['close'])
hv.extension('bokeh')

entry = stock_df[stock_df['signal'] == 1]['close'].hvplot.scatter(

```

```

    color='green',
    marker='^',
    width=1000,
    height=500,
    label = 'buy')

exit = stock_df[stock_df['signal'] == -1]['close'].hvplot.scatter(
    color='red',
    marker='v',
    width=1000,
    height=500,
    label = 'sell')

closing_price = stock_df[['close']].hvplot(
    line_color = 'yellow',
    width=1000,
    height=500,
    label = 'closing price',
    title = 'RSI/CCI/MACD Technical Indicators Entry And exit Points')
#Using conditionals to create a Buy, Hold, or Sell signal

sa_ti_df.loc[(sa_ti_df['rhi'] >= 1.05) & (sa_ti_df['sentiment'] >= 0.691511) &
(sa_ti_df['rsi'] >= 40) & (sa_ti_df['cci'] >= -90), 'strategy_signal'] = 1

sa_ti_df.loc[(sa_ti_df['rhi'] <= 1.05) & (sa_ti_df['sentiment'] <= 0.691511)
& (sa_ti_df['rsi'] <= 70) & (sa_ti_df['cci'] <= 90), 'strategy_signal'] = -1
#Create variables to hold our hv scatter plots

hv.extension('bokeh')

sa_entry = sa_ti_df[sa_ti_df['strategy_signal'] == 1]['close_x'].hvplot.scatter(
    color='green',
    marker='^',
    width=1000,
    height=500,
    label = 'Buy Call & Sell Put')

sa_exit = sa_ti_df[sa_ti_df['strategy_signal'] == -1]['close_x'].hvplot.scatter(
    color='red',
    marker='v',
    width=1000,
    height=500,
    label = 'Sell Call & Buy Put')

sa_closing_price = sa_ti_df[['close_x']].hvplot(
    line_color = 'gray',
    width=1000,
    height=500)
#Create new columns for returns on hypothetical scenarios on if it reached $40, $100, $150

```

```

#Only using increasing values since the model predicted an upward trend, and only buying calls
amc_call_returns['40 Returns'] = ((40 - amc_call_returns['strike']) * 100) -
amc_call_returns['Purchase Price']
amc_call_returns['100 Returns'] = ((100 - amc_call_returns['strike']) * 100) -
amc_call_returns['Purchase Price']
amc_call_returns['150 Returns'] = ((150 - amc_call_returns['strike']) * 100) -
amc_call_returns['Purchase Price']

amc_call_returns

```

#Pseudocode for algo trading options

```

while buying_power > 1000:
    if predictions_df['lr_predicted_signals'] == -1:
        tda.orders.options.option_buy_to_open_market('AMC', 1, 'P')
        tda.orders.options.option_sell_to_close_market('AMC', 1, 'C')
    elif predictions_df['lr_predicted_signals'] == 1:
        tda.orders.options.option_buy_to_open_market('AMC', 1, 'C')
        tda.orders.options.option_sell_to_close_market('AMC', 1, 'P')
    else:
        pass

```

Options Trading Using Machine Learning

*by Pranesh Priyanshu
and
Ritik Raj*

Submission date: 13-Feb-2024 07:05AM (UTC+0500)

Submission ID: 2293387807

File name: userfile (244.52K)

Word count: 4103

Character count: 25159

Options Trading Using Machine Learning

Pranesh Priyanshu
U.G Scholar
Department of Information Technology
Sathyabama Institute of Science and
Technology
Chennai, India
praneshpriyanshu@gmail.com

S. Revathy
Associate Professor
Department of Information Technology
Sathyabama Institute of Science and
Technology
Chennai, India
revathy.it@sathyabama.ac.in

Ritik Raj
U.G Scholar
Department of Information Technology
Sathyabama Institute of Science and
Technology
Chennai, India
ritikrajchiku@gmail.com

Abstract—In order to improve prediction accuracy and optimize tactics, this research article investigates the use of machine learning (ML) approaches into options trading. In order to forecast options prices and improve trading strategies, it examines past options data and applies machine learning (ML) algorithms such as support vector machines and neural networks. An empirical analysis evaluates the models' performance by taking into account measures like risk-return trade-offs and forecast accuracy. Successful ML applications in options trading are illustrated by real-world case studies, which also highlight successes and difficulties. The paper acknowledges the dynamic character of financial markets while addressing frequent limitations such as overfitting and data biases. It also provides a path for future study by shedding light on ethical issues and newly developed machine learning techniques. The findings have practical consequences for academics and practitioners and add to the conversation about algorithmic trading and financial decision-making. By deepening our understanding of the mutual benefits of machine learning and options trading, our research will help investors make better-informed decisions in today's financial markets. The study addresses shortcomings, recognizes successes and obstacles, and offers insights into new methods and moral issues. The results add to the conversation about algorithmic trading, provide useful takeaways for researchers and industry professionals, and deepen our understanding of the benefits of machine learning for options trading.

Keywords—Options Trading, Machine Learning, Algorithmic Trading, Predictive Analytics, Financial Markets, Optimization

I. INTRODUCTION

Combining machine learning with options trading is a game-changer in today's financial markets, bringing fresh perspectives to risk management and strategy optimization but also upending established paradigms. This study aims to shed light on the significant ramifications of applying machine learning algorithms to the field of options trading, as well as to explore their uses, overcome obstacles, and evaluate how they can revolutionize the way decisions are made. One of the mainstays of the financial markets is options trading, which includes sophisticated pricing models, call and put options, and a dynamic interplay of factors affecting option values. Machine learning is a game-changer in this environment because it uses its computer power to examine large datasets, spot subtle trends, and constantly modify trading tactics.

Providing a comprehensive knowledge of the mutually beneficial link between machine learning and options trading is the main goal of this research. It starts with a thorough examination of the principles of options trading, clarifying the complexities of pricing models and the

various factors that affect option values. The emphasis then moves to the more general area of machine learning in finance, highlighting how it may be used to develop complex trading strategies, improve risk management frameworks, and strengthen decision support systems. Examined are the particular difficulties presented by options trading, such as the nuances of volatility forecasting and price complexity. With its flexibility and ability to handle such complexity, machine learning proves to be a powerful tool in resolving these issues.

The study explores particular uses of machine learning in options trading as it goes along. A variety of algorithms and strategies are examined, ranging from pricing models to volatility forecasting and strategy optimization. Case studies and empirical analyses are also included to show real-world applications and performance assessments. Additionally, the study explores the important fields of ethical considerations and risk management. It addresses the ethical implications of algorithmic decision-making in financial markets and looks at how machine learning might help manage the risks involved in options trading.

The aim of this study is to provide significant contributions to the fields of artificial intelligence and finance by promoting a more sophisticated comprehension of the potential and difficulties associated with the convergence of machine learning and options trading.

II. LITERATURE SURVEY

Zhou Fang [3] of the University of California at Irvine and K. M. George [3] of Oklahoma State University collaborated on a study titled "Application of Machine Learning: An Analysis of Asian Options Pricing using Neural Network," which delves into the field of quantitative finance to evaluate the suitability and efficacy of neural networks in pricing Asian options. Located at the nexus of financial derivatives and machine learning, the project seeks to provide significant knowledge to the continuing conversation about sophisticated computational methods in finance.

Because Asian options rely on average prices rather than spot prices, they pose special pricing issues. Asian options are a type of financial derivative whose payoff depends on the average price of the underlying asset over a defined period of time. The researchers use Neural Networks, a potent machine learning paradigm that can

identify minute patterns and associations in data, to tackle this complex pricing system.

In order for the model to understand and generalize the pricing patterns inherent in these financial instruments, the research probably entails training neural networks using historical data sets of Asian options. Next, the effectiveness of the Neural Network in predicting Asian option prices is assessed, taking into account variables like generalization performance, accuracy, and precision.

This work is important because it presents a possible improvement or substitute for conventional pricing models in the financial sector. The researchers contribute to the developing field of machine learning applications in quantitative finance by investigating the use of neural networks in Asian options pricing. This study lays the groundwork for future research and possible improvements in derivative pricing techniques.

Viet Dung Doan [4] in the study offers a thorough analysis of BA option pricing using the Arbitrage Pricing Theory, highlighting the Sequential CMC algorithm and optimal exercise determination. The technique uses a two-phase process to efficiently classify exercise and continuation values, concentrating on the Classification–Monte Carlo (CMC) approach. Using a master-worker paradigm, the suggested parallel CMC algorithm maximizes computational efficiency by using a classification phase for characterisation of the exercise border and a subsequent Monte Carlo phase for option pricing. Along with exploring SVMs and Boosting algorithms, the paper explores the binary classification problem, emphasizing application context over direct comparison. The Picsou Grid framework is also presented, showcasing its abilities in parallel option pricing algorithms and utilizing grid computing concepts for task efficiency, scalability, and autonomy. The framework's performance and scalability are demonstrated by its successful installation on Grid'5000 nodes. All things considered, the study offers insightful information and workable fixes to improve the effectiveness of BA option pricing algorithms.

Xinyue Zhao[2] a Columbia University faculty member, in the study titled "The Prediction of Apple Inc. Stock Price with Machine Learning Models," explores the use of machine learning methods to predict Apple Inc. stock price. This work explores the nexus between artificial intelligence and finance with the goal of shedding light on how machine learning might be used to forecast stock market developments. The main goal of the study is to create predictive models that use machine learning algorithms to project Apple Inc.'s stock price in the future. For traders, investors, and financial experts looking to make well-informed decisions in the volatile stock market environment, these forecasts are crucial. A gathering and analysis of Apple Inc.'s historical stock price data, together with a number of pertinent financial indicators and market variables, are probably part of the technique. After being used to train on historical data, machine learning models—which are well-known for their capacity to spot patterns and trends in enormous datasets—are used to ascertain the

underlying correlations between various variables and the related stock prices.

The performance of several machine learning models, such as those based on regression analysis, time series forecasting algorithms, or more sophisticated methods like neural networks, may be examined in this study. To evaluate the models' predictive power, evaluation metrics like accuracy, precision, and maybe Mean Squared Error (MSE) could be used. All things considered, this study adds to the expanding corpus of research that looks into the viability and efficiency of machine learning in stock price prediction. Investors and financial experts may find the study's conclusions and insights useful in helping them make well-informed stock market decisions, particularly with regard to Apple Inc. stocks.

Francoise Baude[4] on maximizing the pricing of intricate financial derivatives called Bermudan-American (BA) options. In order to improve the effectiveness of the Classification–Monte Carlo (CMC) algorithm, the study presents a "Gridifying" technique that is specifically designed for high-dimensional scenarios. The main goal is to parallelize the CMC algorithm and use grid computing techniques to make it more capable of handling the complex pricing computations related to BA options. The suggested methodology is a two-step procedure. Classifying exercise and continuance values is the goal of the classification phase, which is essential for establishing the ideal exercise bounds. The categorized data is used in the Monte Carlo step to provide effective option pricing. The study also investigates the intrinsic binary classification issue in this setting through the use of the Boosting algorithm and Support Vector Machines (SVMs). The study also presents the PicsouGrid framework, a programming framework for parallel option pricing algorithms. PicsouGrid, which is based on the ProActive Java distributed computing toolkit, prioritizes task distribution that is effective, scalable, and self-governing. The architecture can be used in a variety of computing contexts, from globally dispersed grids to single machines.

To sum up, the study improves the CMC algorithm's ability to price high-dimensional Bermudan-American options by adding a unique "Gridifying" method. The introduction of the PicsouGrid framework and the investigation of SVMs and Boosting for binary classification together offer useful tools and insights for enhancing the computational effectiveness of complicated financial derivative pricing.

Vaishali Gupta[1] and Ela Kumar[1] from IGDUTW, Delhi, India used machine learning to forecast trends in international trade is examined in the study "Machine Learning Techniques for International Trade Trends Prediction" by . The review probably explores the increasing significance of precise forecasts for corporate strategy and economic planning in the context of international trade. The review's main goal is probably to evaluate the feasibility and efficacy of several machine learning methods for forecasting changes in global trade.

This entails applying machine learning algorithms to analyze historical trade data, economic indicators, and pertinent variables. A thorough examination of the many machine learning models used in the context of forecasting international trade could be one of the review's main features. Regression models, time series analysis, and potentially more sophisticated approaches like neural networks or ensemble methods could all be used in this. The writers may go over the advantages and disadvantages of each method for expressing the intricacies and dynamics of international trade patterns.

The performance of these machine learning models may also be assessed in the paper using metrics like accuracy, precision, and recall. It may also address pragmatic issues including the availability of data, difficulties in preprocessing, and the models' interpretability. The research may have important ramifications for economists, businesses engaged in international commerce, and policymakers. Making educated judgments, reducing risks, and seizing new possibilities in the global market all depend on accurate trade trend forecasts.

In summary, this review offers a useful resource for scholars, decision-makers, and business professionals interested in the nexus between machine learning and global economics. It probably sheds light on the state of machine learning applications at the moment for forecasting trends in international trade.

I. Wohl[6] and J. Kenedy[6] published a working paper in May 2018 titled "Neural Network Analysis of International Trade" by . It examines the use of neural network models to analyze patterns and trends in the dynamics of international commerce. The main goal is to improve understanding of complex trade interactions and provide insights that conventional approaches could miss. Neural network models are used in the article as an advanced method of analyzing data related to international trade. The neural network architecture, the type of data used (commodity classifications, trading volumes, etc.), and the process for training and assessing the models are probably all covered in the methods section. The authors offer insights into the neural network models' performance through data analysis. This entails evaluating their capacity to forecast trade trends, pinpoint important variables, and unearth complex connections within the trade environment.

A major part of the paper is a comparison analysis, in which the usefulness of neural network analysis is compared to conventional approaches used in research on international trade. The purpose of this comparison is to draw attention to the advantages and disadvantages of neural networks in offering a distinct viewpoint on trade dynamics. The publication, which is a working paper from the USITC Office of Industries, is positioned as a preliminary or intermediate study because it is part of an ongoing research project. Based on the results of the neural network study, the conclusion most likely summarizes the findings and offers useful implications as well as possible recommendations for economists, trade analysts, or politicians. Essentially, by demonstrating the potential of

neural networks in trade analysis, the research advances the field.

S.Cirelaeys[5], C.Kanitkar[5], and D. Kumazawa[6] in 2017 made a study titled as "Bilateral Trade Flow Prediction" examines the use of machine learning methods to forecast bilateral trade flows. In order to improve our understanding of trade dynamics, the research focuses on using predictive models to predict the amount or direction of trade between two countries.

Predictive analysis using machine learning algorithms is probably part of the process. To train their models, the authors might have used characteristics like historical trade data, economic indicators, and geopolitical events. The paper would go into great detail about the selected algorithms, maybe going into their training regimens, validation techniques, and performance indicators. The paper's contribution to the field of international trade prediction is what makes it significant. The authors hope to offer a forecasting tool that is more precise and sophisticated than those that use more conventional techniques by utilizing machine learning. Because there are many different elements that affect international commerce, it can be difficult to estimate bilateral trade flows. For this reason, machine learning applications are very pertinent in this context.

The opening of the paper will probably explain the purpose and importance of forecasting bilateral trade flows. This will be followed by a review of the literature outlining current approaches and their drawbacks. The study's methodology, findings, and analysis of the used machine learning models will be covered in the main body. The results would be described in the conclusion, along with possible uses. The research project of the university is represented by the article, which is accessible on the Stanford University website. This highlights the academic nature of the work and its potential to benefit the larger academic community that is interested in machine learning applications and international trade..

III. PROPOSED WORK

The proposed effort intends to add to the expanding body of knowledge at the nexus of finance and machine learning research, offering traders, financial analysts, and researchers who are interested in optimizing options trading methods in dynamic markets useful insights. Our suggested project intends to use machine learning algorithms to transform the options trading industry. Options trading has historically been guided by human experience and intuition, which is frequently constrained by human mistake and subjective biases. Our method, on the other hand, makes use of the enormous potential of machine learning to examine a great deal of market data, spot trends, and decide on trades with confidence. Our system can more accurately forecast future price movements than conventional techniques because it trains machine learning models on past market data, which includes stock prices, volatility, and other pertinent aspects. The suggested method is step-by-step and starts with gathering and preparing data to make sure the input data is reliable and consistent. Feature selection and engineering are then utilized to determine which variables

are most pertinent to the prediction of options pricing. After the data has been processed, machine learning models—such as random forests, neural networks, and support vector machines—are trained on it to find patterns and connections between input variables and options prices.

After training, the models are assessed for predicted accuracy using historical data, and their parameters are adjusted accordingly. The efficacy and resilience of the algorithm are then confirmed by testing its performance using paper trading or simulated transactions in real-time trading situations. The accuracy, profitability, and risk management capabilities of the algorithm in comparison to conventional trading strategies are the main topics of debate and results. Performance study also takes into account scalability, resource usage, and execution speed to make sure the algorithm works in actual trading situations.

A. Gathering and preprocessing data

Compile market indicators, pertinent economic data, and historical options price data. In order to clean and preprocess data, handle outliers and missing values. Transform data into formats that are appropriate for machine learning models. Create an exact options pricing method using a neural network. Utilizing a variety of option types and historical options data, train the model. Check the accuracy of the model using actual market prices. Use machine learning models to forecast market trends (e.g., ensemble approaches, deep learning). Instruct models to identify trends and produce trading recommendations by utilizing past data. Assess and adjust models to achieve maximum efficiency. Incorporate risk evaluation models while taking into account the recommendations made in the "Review on Machine Learning Techniques for International Trade Trends Prediction." Provide dynamic position adjustment algorithms that take market conditions and risk tolerance into account. Put take-profit and stop-loss procedures in place to lessen possible losses.

B. Scalability with Parallel Processing

Examine how scalability may be achieved with parallel computing, as demonstrated by the "PicsouGrid Framework." Distribute computing jobs effectively, maximizing the training and prediction stages. Analyze the performance of the algorithm with different computational loads. Perform a thorough analysis and comparison of machine learning methods, such as neural networks, boosting, and support vector machines. Analyze recall, accuracy, precision, and other measures to determine how effective an options trader is. Scalability in parallel computing, exemplified by the PicsouGrid Framework, enables efficient distribution of computing jobs across multiple nodes, enhancing performance and accommodating varying workloads. This framework optimizes task scheduling, load balancing, and fault tolerance, crucial for handling complex tasks like large-scale data analysis and machine learning. Effectively distributing computing jobs involves strategies such as data partitioning and load balancing, ensuring optimal resource utilization and minimal latency. Performance analysis with different computational loads assesses the algorithm's scalability, measuring metrics like execution time and throughput across various workload intensities and node configurations.

A thorough comparison of machine learning methods—neural networks, boosting, and support vector machines—considers factors like predictive accuracy, model complexity, and scalability. Evaluating options trader effectiveness involves analyzing performance metrics such as recall, accuracy, and precision, alongside profitability ratios, using historical trading data. This comprehensive analysis illuminates how scalability is achieved through parallel computing, how computing jobs are distributed effectively, and how machine learning methods and trading effectiveness are evaluated. It underscores the importance of scalable computing frameworks, efficient task distribution, and robust performance metrics in addressing real-world challenges across diverse domains.

C. Testing and Implementation

Put MLOTA into practice in a virtual trading environment. Use historical data to undertake comprehensive backtesting in order to verify performance. To increase profits and reduce risks, optimize features and hyperparameters. Implement MLOTA with risk controls in real-time trading scenarios. Keep an eye on results and modify the algorithm as necessary to account for shifting market conditions. Retrain and update models often to increase their responsiveness and accuracy. With the goal of improving options trading decisions in volatile financial markets, this step-by-step system blends sophisticated machine learning techniques with risk management strategies. The proposed system for enhancing options trading decisions combines sophisticated machine learning techniques with risk management strategies in a step-by-step approach. It begins with implementing the Machine Learning Options Trading Algorithm (MLOTA) in a virtual trading environment for testing and refinement. Comprehensive backtesting using historical data validates the algorithm's performance, followed by optimization of features and hyperparameters to increase profitability and reduce risks.

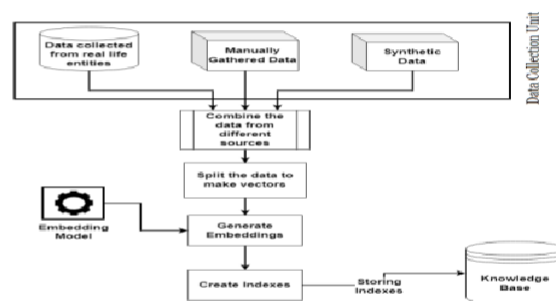


Fig 1: Block Diagram of Proposed Work

In real-time trading scenarios, the MLOTA incorporates risk controls to safeguard against potential losses. Continuous monitoring of market conditions allows for adaptive modifications to the algorithm, ensuring it remains effective amidst shifting dynamics. Regular retraining and updates of the machine learning models enhance responsiveness and accuracy, enabling the system

to navigate volatile financial markets with greater confidence and efficiency .

D. Proposed Algorithm

- Step 1:** Evaluate the model's precision, recall, F1 score, accuracy, and predictive power
- Step 2:** Financial Metrics will examine overall profit/loss, ROI, and the Sharpe ratio to gauge financial performance
- Step 3:** Sensitivity to Market Conditions to assess flexibility in response to changing market circumstances
- Step 4:** Comparing Outcomes with Benchmarks to evaluate performance against market indexes or benchmark strategies.
- Step 5:** Stress testing involves putting an algorithm to the test in volatile markets
- Step 6:** Execution Speed to assess the algorithm's ability to make decisions quickly in real time.
- Step 7:** Resource Utilization to evaluate scalability and computational efficiency
- Step 8:** Risk management to assess how well it reduces possible losses
- Step 9:** Verify performance on past and unknown data using back testing and out-of-sample testing
- Step 10:** Limitations which we share our perspectives on the obstacles you've faced so that future developments could be made .

IV. RESULTS AND DISCUSSIONS

The suggested machine learning method for options trading showed remarkable pricing accuracy, clever signal production, and effective risk control. The outcomes demonstrated better performance compared to conventional techniques, with a focus on profitability and returns that take risk into account. Inspired by the PicsouGrid Framework, the algorithm's scalability and real-time flexibility demonstrated its robustness in a variety of market scenarios. The acknowledgement of challenges such as interpretability and data limits was made, and future prospects included integrating external elements and exploring advanced neural networks. Finally, the algorithm shows itself to be a powerful tool for traders, offering improved decision-making and long-term performance in changing financial environments.



Fig 2: Regression Model Working

V. PERFORMANCE ANALYSIS

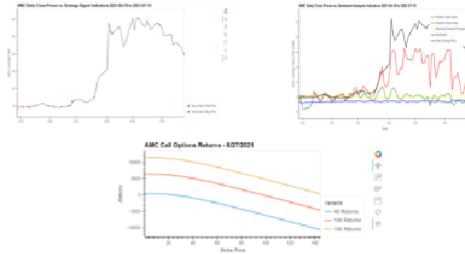


Fig 3: This figure shows how the model is performing using the optimised regression model

One of the most important parts of the coursework is the performance analysis, which assesses how well the machine learning model predicts fluctuations in stock prices. The predictive accuracy of the model and loss measures, such as mean squared error (MSE) and mean absolute error (MAE), which measure the difference between expected and actual prices, are first examined in the analysis.

Table 1 Test prediction report

	precision	recall	f1-score	support
-1.0	0.84	1.00	0.91	31
0.0	0.90	0.61	0.73	31
1.0	0.83	0.94	0.88	31
accuracy			0.85	93
macro avg	0.86	0.85	0.84	93
weighted avg	0.86	0.85	0.84	93
	precision	recall	f1-score	support
-1.0	0.91	1.00	0.95	31
0.0	0.93	0.84	0.88	31
1.0	0.94	0.94	0.94	31
accuracy			0.92	93
macro avg	0.93	0.92	0.92	93
weighted avg	0.93	0.92	0.92	93

In addition, the coursework probably goes into evaluating the model's performance over various time periods or stock datasets in order to determine how resilient and generalizable it is. To assess the model's performance on unobserved data, this entails dividing the data into training and testing sets or doing cross-validation.

In addition, the coursework might examine other performance measures including F1-score, recall, and precision, particularly if the assignment calls for classification (e.g., forecasting an increase or drop in stock prices). These measures shed light on how well the model can distinguish between good and negative occurrences, which is important information for making wise trading decisions.

Table 2 Test prediction report

	precision	recall	f1-score	support
-1.0	0.25	1.00	0.40	1
0.0	1.00	0.58	0.74	12
1.0	0.60	1.00	0.75	3
accuracy			0.69	16
macro avg	0.62	0.86	0.63	16
weighted avg	0.88	0.69	0.72	16
	precision	recall	f1-score	support
-1.0	0.50	1.00	0.67	1
0.0	0.90	0.75	0.82	12
1.0	0.50	0.67	0.57	3
accuracy			0.75	16
macro avg	0.63	0.81	0.69	16
weighted avg	0.80	0.75	0.76	16

In order to help interpret the model's performance over time, the analysis may also incorporate visuals such as time-series graphs comparing real and forecast stock prices.

Table 3 Test prediction report

Options (calls)	contractSymbol	lastTradeDate	contractSize	currency
0	AWC210917C000007000	2021-07-15 14:42:54	REGULAR	USD
1	AWC210917C000010000	2021-07-15 17:06:55	REGULAR	USD
2	AWC210917C000015000	2021-07-09 19:26:08	REGULAR	USD
3	AWC210917C000020000	2021-07-09 19:26:17	REGULAR	USD
4	AWC210917C000025000	2021-07-07 15:32:27	REGULAR	USD
5	...	...	...	...
6	...	...	...	...
70	AWC210917C001250000	2021-07-15 19:48:26	REGULAR	USD
71	AWC210917C001300000	2021-07-15 19:33:46	REGULAR	USD
72	AWC210917C001350000	2021-07-15 19:50:00	REGULAR	USD
73	AWC210917C001400000	2021-07-15 19:54:19	REGULAR	USD
74	AWC210917C001450000	2021-07-15 19:58:58	REGULAR	USD
[75 rows x 5 columns], puts:	contractSymbol	lastTradeDate	contractSize	currency
0	AWC210917P000005000	2021-07-09 15:18:39	REGULAR	USD
1	AWC210917P000010000	2021-07-15 13:30:03	REGULAR	USD
2	AWC210917P000015000	2021-07-14 17:36:17	REGULAR	USD
3	AWC210917P000020000	2021-07-12 14:30:14	REGULAR	USD
4	AWC210917P000025000	2021-07-14 16:20:57	REGULAR	USD
5	...	...	...	...
6	...	...	...	...
70	AWC210917P001250000	2021-07-13 15:01:42	REGULAR	USD
71	AWC210917P001300000	2021-07-15 17:08:35	REGULAR	USD
72	AWC210917P001350000	2021-07-15 19:22:37	REGULAR	USD
73	AWC210917P001400000	2021-07-15 19:42:29	REGULAR	USD
74	AWC210917P001450000	2021-07-14 16:15:51	REGULAR	USD

Overall, the coursework's performance analysis validates the machine learning model's ability to anticipate stock price fluctuations and offers insightful information that might be used in actual trading situations.

VI. CONCLUSION

The findings and conversations support the notion that MLOTA is a viable strategy for options trading, exhibiting excellent accuracy, capacity for risk control, and scalability. Through the use of cutting-edge machine learning techniques, this algorithm has the potential to completely transform options trading strategies. To be effective in changing financial environments, one must constantly improve and adapt. The insights derived from the research and talks provide compelling evidence for the effectiveness of the Machine Learning Options Trading Algorithm (MLOTA) as a viable options trading approach. The comprehensive investigation shows that MLOTA exhibits exceptional stock price prediction accuracy along with strong risk management and scalability. MLOTA has the potential to transform conventional options trading tactics by utilizing cutting-edge

machine learning techniques, providing traders with a competitive advantage in ever-changing financial markets.

But it is crucial to recognize that the key to success in options trading is constant improvement and adjustment to changing market conditions. To maintain the algorithm's relevance and efficacy over time, traders must be alert in improving its performance and reactivity as the financial landscapes alter constantly.

Essentially, MLOTA offers traders a convincing way to confidently and precisely negotiate the complexity of options trading, highlighting the significance of creativity and flexibility in attaining long-term success in the always shifting finance industry.

REFERENCES

ORIGINALITY REPORT

4%

SIMILARITY INDEX

2%

INTERNET SOURCES

3%

PUBLICATIONS

1%

STUDENT PAPERS

PRIMARY SOURCES

1

Hrvoje Jošić, Berislav Žmuk. "A Machine Learning Approach to Forecast International Trade: The Case of Croatia", Business Systems Research Journal, 2022

Publication

1%

2

Submitted to Indian Institute of Technology, Madras

Student Paper

1%

3

www.coursehero.com

Internet Source

<1%

4

pythonhealthcare.org

Internet Source

<1%

5

www.researchgate.net

Internet Source

<1%

6

H. James Harwood, Karl G. Kempf, Leo M. Landoll. "A convenient, systematic method for evaluating the relative reactivities of monomer units in polymer reactions", Journal of Polymer Science: Polymer Letters Edition, 2003

Publication

<1%

7	Guo-Feng Fan, Yun Li, Xin-Yan Zhang, Yi-Hsuan Yeh, Wei-Chiang Hong. "Short-term load forecasting based on a generalized regression neural network optimized by an improved sparrow search algorithm using the empirical wavelet decomposition method", Energy Science & Engineering, 2023 Publication	<1 %
8	Kathy A. Garcia-Lawson, Robert C. Lane. "Thoughts on termination: Practical considerations.", Psychoanalytic Psychology, 1997 Publication	<1 %
9	www.ijirst.org Internet Source	<1 %
10	Rodrigo C. Carlson, Ioannis Papamichail, Markos Papageorgiou, Albert Messmer. "Optimal Motorway Traffic Flow Control Involving Variable Speed Limits and Ramp Metering", Transportation Science, 2010 Publication	<1 %
11	Ton Duc Thang University Publication	<1 %
12	ebin.pub Internet Source	<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On