TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

PROJECT NO.: THA079MSISE011

DEEP REINFORCEMENT LEARNING TO DESIGN TRADING STRATEGIES FOR ORGANIZATIONS LISTED IN NEPSE

BY

RAVI PRAJAPATI

A PROJECT

SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATICS AND INTELLIGENT SYSTEMS ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

KATHMANDU, NEPAL

JULY, 2024

Deep Reinforcement Learning to Design Trading Strategies for Organizations listed in NEPSE

by

Ravi Prajapati

THA079MSISE011

Project Supervisor

Devendra Kathayat

A project submitted in partial fulfillment of the requirements for the degree of

Master of Science in Informatics and Intelligent Systems Engineering

Department of Electronics and Computer Engineering
Institute of Engineering, Thapathali Campus
Tribhuvan University
Kathmandu, Nepal

July, 2024

ACKNOWLEDGMENT

This project work would not have been possible without the guidance and the help of several individuals who in one way or another contributed and assisted in the preparation and completion of this study.

First of all, I would like to extend my heartfelt gratitude to the M.Sc. coordinator, **Er. Dinesh Baniya Kshatri**, whose pivotal role in overseeing the project works, offering invaluable insights, and demonstrating boundless patience has been immensely appreciated. Secondly, I would also like to express my gratitude to my supervisor, **Devendra Kathayat,** for guiding me on this project.

I would also like to thank my classmates and friends for offering me advice and moral support. A special thanks to my parents, who have been my biggest supporters, loving me unconditionally and wanting the best for me.

Ravi Prajapati
THA079MSISE011
July, 2024

ABSTRACT

In recent years, deep reinforcement learning (DRL) has gained significant traction in financial markets, offering innovative methods for designing robust trading strategies. This project delves into the use of the Proximal Policy Optimization (PPO) algorithm, a cutting-edge DRL technique, to develop trading strategies specifically tailored for organizations listed on the Nepal Stock Exchange (NEPSE), with a particular focus on the hydropower sector.

By leveraging the PPO algorithm, this research aims to optimize trading decisions to maximize returns while effectively managing risks in the volatile and dynamic stock market environment. The study involves a thorough analysis of historical trading data, simulation of trading scenarios, and rigorous backtesting to validate the effectiveness of the proposed strategies.

Keywords: Backtesting, Deep Reinforcement Learning (DRL), Nepal Stock Exchange (NEPSE), Proximal Policy Optimization (PPO), Trading Strategies

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADX	Average Directional Index
AHL	Asian Hydropower Limited
AHPC	Arun Valley Hydropower Development Company Limited
AKJCL	Ankhukhola Hydropower Company Limited
AKPL	Arun Kabeli Power Limited
API	Api Power Company Limited
ARI-MA	Auto-Regressive Moving Average
BARUN	Barun Hydropower Company Limited
BEDC	Bhugol Energy Development Company Limited
BGWT	Bhagawati Hydropower Development Company Limited
BHDC	Bindhyabasini Hydropower Development Company Limited
BHL	Balephi Hydropower Limited
BHPL	Barahi Hydropower Public Limited
BNHC	Buddha Bhumi Nepal Hydropower Company Limited
BPCL	Butwal Power Company Limited
CHCL	Chilime Hydro power Company Limited
CHL	Chhyangdi Hydropower Company Limited
CKHL	Chirkhwa Hydro Power Limited
CNN	Convolutional Neural Network
DDQL	Double Deep Q Learning
DHPL	Dibyashwari Hydropower Company Limited
DOLTI	Dolti Power Company Ltd
DORDI	Dordi Khola Hydropower Company Limited
DRL	Deep Reinforcement Learning
EHPL	Eastern Hydropower Limited
GDPG	Gated Deterministic Policy Gradient trading strategy
GDQN	Gated Deep Q-learning trading strategy
GHL	Ghalemdi Hydro Limited
GLH	Greenlife Hydropower Limited
GPU	Graphics Processing Unit

GRU	Gated Recurrent Unit
GVL	Green Ventures Limited
HDHPC	Himal Dolakha Hydropower Company Limited
HHL	Himalayan Hydropower Limited
HPPL	Himalayan Power Partner Limited
HURJA	Himalaya Urja Bikas Company Limited
IHL	Ingwa Hydropower Ltd
JOSHI	Joshi Hydropower Development Company Limited
KBSH	Kutheli Bukhari Small Hydropower Limited
KKHC	Khani Khola Hydropower Company Limited
KPCL	Kalika Power Company Limited
LEC	Liberty Energy Company Limited
LS-SVM	Least Squares Support Vector Machine
LSTM	Long Short Term Memory
MACD	Moving Average Convergence Divergence
MAE	Mean Absolute Error
MAKAR	Makar Jitumaya Suri Hydropower Company Limited
MANDU	Mandu Hydropower Limited
MAPE	Mean Absolute Percentage Error
MBJC	Madhya Bhotekoshi Jalavidyut Company Limited
MCHL	Menchhiyam Hydropower Limited
MEHL	Manakamana Engineering Hydropower Limited
MEL	Modi Energy Limited
MEN	Mountain Energy Nepal Limited
MHCL	Molung Hydropower Company Limited
MHL	Mandakini Hydropower Limited
MHNL	Mountain Hydro Nepal Limited
MKHC	Maya Khola Hydropower Company Limited
MKHL	Mai Khola Hydropower Limited
MKJC	Mailung Khola Jal Vidhyut Company Limited
ML	Machine Learning
MMKJL	Mathillo Mailun Khola Jalvidhyut Limited

MSHL	Mid Solu Hydropower Company Limited
NASDAQ	National Association of Securities Dealers Automated Quotations
NEPSE	Nepal Stock Exchange
NGPL	Ngadi Group Power Limited
NHDL	Nepal Hydro Developer Limited
NHPC	National Hydro Power Company Limited
NYADI	Nyadi Hydropower Limited
OBV	On Balance Volume
PHCL	Peoples Hydropower Company Limited
PMHPL	Panchakanya Mai Hydropower Limited
PPCL	Panchthar Power Company Limited
PPL	People's Power Limited
PPO	Proximal Policy Optimization
RADHI	Radhi Bidyut Company Limited
RAWA	Rawa Energy Development Ltd.
RFPL	River Falls Power Limited
RHGCL	Rapti Hydro & General Construction Limited
RHPL	Rasuwagadhi Hydropower Company Limited
RIDI	Ridi Power Company Limited
RL	Reinforcement Learning
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RURU	Ru Ru Jalbidhyut Pariyojana Limited
S&P500	Standard and Poor's 500
SAHAS	Sahas Urja Limited
SGHC	Swet-Ganga Hydropower & Construction Limited
SHEL	Singati Hydro Energy Limited
SHPC	Sanima Mai Hydropower Limited
SIKLES	Sikles Hydropower Limited
SJCL	Sanjen Jalavidhyut Company Limited
SMH	Supermai Hydropower Limited
SMHL	Super Madi Hydropower Limited

SMJC	Sagarmatha Jalbidhyut Company Limited
SO	Stochastic Oscillator
SPC	Samling Power Company Limited
SPDL	Synergy Power Development Limited
SPHL	Sayapatri Hydropower Limited
SPL	Shuvam Power Limited
SSHL	Shiva Shree Hydropower Limited
TAMOR	Sanima Middle Tamor Hydropower Limited
TPC	Terhathum Power Company Limited
TSHL	Three Star Hydropower Limited
TVCL	Trishuli Jal Vidhyut Company Limited
UHEWA	Upper Hewakhola Hydropower Company Limited
ULHC	Upper Lohore Khola Hydropower Company Limited
UMHL	United Modi Hydropower Limited
UMRH	United Idi-Mardi and R.B. Hydropower Limited
UNHPL	Union Hydropower Limited
UPCL	Universal Power Company Limited
UPPER	Upper Tamakoshi Hydropower Limited
USHEC	Upper Solu Hydro Electric Company Limited
USHL	Upper Syange Hydropower Limited
VLUCL	Vision Lumbini Urja Company Limited

1 INTRODUCTION

1.1 Background

Financial markets have become a key area for applying advanced computational techniques, and the use of artificial intelligence (AI) and machine learning (ML) has significantly transformed trading strategies. Deep reinforcement learning (DRL), a branch of ML, stands out as a particularly effective tool for optimizing complex decision-making processes in this domain. DRL combines the feature-extraction capabilities of deep learning with the decision-making framework of reinforcement learning, allowing for powerful end-to-end control and learning.[1] Unlike traditional methods, DRL adapts by learning from its interactions with the market, making it especially suitable for the unpredictable nature of financial markets.

The Nepal Stock Exchange (NEPSE), Nepal's primary stock exchange, is an emerging market with distinct challenges and opportunities due to its market structure, regulatory framework, and economic environment. Within NEPSE, the hydropower sector plays a vital role, reflecting Nepal's abundant water resources and the importance of hydroelectric power in the country's energy and economic strategies.

Proximal Policy Optimization (PPO), a leading DRL method developed by OpenAI, has proven highly effective across various complex tasks,[2] including financial trading. PPO is favored for its balance between ease of use and high performance. It improves upon earlier policy gradient methods by ensuring more stable and reliable updates, which is essential in the volatile and high-risk world of financial markets.

Despite DRL's potential, its use in NEPSE, particularly within the hydropower sector, has not been widely explored. Traditional trading strategies often depend on historical data and fixed rules, which may not respond well to rapidly changing market conditions. DRL, with its ability to continuously learn and adapt to market dynamics, offers a promising alternative for traders in this sector.

1.2 Motivation

The motivation for this project comes from the need to improve trading strategies in emerging markets, with a focus on the Nepal Stock Exchange (NEPSE). The hydropower sector, which plays a crucial role in Nepal's economy, has unique market dynamics, making it a perfect candidate for advanced techniques like deep reinforcement learning (DRL).

Traditional trading strategies often fall short when dealing with the volatility and complexity of financial markets, especially in emerging economies. These strategies typically rely on static models and historical data, which struggle to adapt to rapid market changes and new information. As a result, traders and investors face difficulties in making informed decisions that maximize returns while effectively managing risks.

DRL offers a promising alternative by allowing trading algorithms to continuously learn and adapt based on real-time market interactions. The Proximal Policy Optimization (PPO) algorithm, in particular, has been effective in handling complex decision-making tasks. Applying PPO to develop trading strategies for NEPSE's hydropower sector could lead to more robust and adaptive approaches that better handle market fluctuations.

This project aims to create a sophisticated trading strategy using DRL and the PPO algorithm, specifically tailored to the hydropower sector within NEPSE. The expected benefits include better decision-making for traders and investors, improved market efficiency, and more stable and sustainable financial growth in the sector. Additionally, this research hopes to contribute to a deeper understanding of how DRL can be applied in financial markets, especially in the context of emerging economies.

1.3 Problem Statement

Financial markets, especially in emerging economies like Nepal, offer unique challenges and opportunities for traders and investors. Traditional trading strategies, which often depend on static models and historical data, are becoming less effective in dealing with the complexities and volatility of these markets. This is particularly true for the hydropower sector in the Nepal Stock Exchange (NEPSE), where market dynamics are shaped by a variety of economic, environmental, and regulatory factors.

While deep reinforcement learning (DRL) has the potential to address these challenges, its use in NEPSE is still largely unexplored. The Proximal Policy Optimization (PPO) algorithm, a leading DRL method, holds promise due to its balance of performance and stability. However, there is a lack of research and practical implementations of

PPO-based trading strategies specifically for NEPSE's hydropower sector.

This project seeks to develop a robust and adaptive trading strategy for the hydropower sector in NEPSE using the PPO algorithm. The research will focus on the following key questions:

1. How can the PPO algorithm be customized to optimize trading decisions within NEPSE's hydropower sector?
2. How does the performance of a PPO-based trading strategy compare to traditional methods in terms of profitability and risk management?
3. How effective is the PPO-based strategy in real-time trading environments when tested against historical data and through simulations?

By addressing these questions, this research aims to connect advanced DRL techniques with practical applications in emerging markets, offering valuable insights and tools for traders, investors, and financial analysts.

1.4 Objective of Project

The objectives of this project are:

- To develop a trading strategy based on the Proximal Policy Optimization (PPO) algorithm for organizations in the hydropower sector listed on the Nepal Stock Exchange (NEPSE).
- To validate the effectiveness of this trading strategy through thorough backtesting and simulation.

1.5 Scope of Project

This project will investigate the potential of the PPO algorithm in crafting effective trading strategies for the hydropower sector in the Nepal Stock Exchange (NEPSE). The study will involve gathering historical trading data, training the PPO model, and simulating various trading scenarios to assess its performance. Additionally, the project aims to provide insights into the adaptability of deep reinforcement learning (DRL) algorithms in emerging markets, highlighting their potential for broader financial applications.

While the goal is to develop a robust trading strategy, the project is limited by the availability and quality of historical trading data from NEPSE. Market anomalies and unpredictable economic factors could also affect the model's performance. Furthermore, the focus will be specifically on the hydropower sector, meaning the findings may not be directly applicable to other sectors within NEPSE or to other stock exchanges.

1.6 Potential Applications

The potential applications of this project extend across various areas of the Nepalese financial market, including investment strategy development and risk management.

1. **Algorithmic Trading Systems:** The PPO-based trading strategy could be implemented in automated trading systems used by brokerage firms and individual traders. This could streamline trading operations and improve performance through data-driven decision-making.

2. **Investment Advisory Services:** The research findings could be used to provide tailored advice and recommendations to clients interested in investing in the hydropower sector, enhancing the quality and effectiveness of advisory services offered by financial professionals.

3. **Automated Trading Bots:** The PPO algorithm could be incorporated into trading bots operating on NEPSE, allowing these bots to make autonomous trading decisions based on predefined strategies that adapt to changing market conditions, optimizing trading performance.

4. **Risk Management Tools:** Software tools could be developed using the PPO-based strategy to monitor and manage risks associated with investments in the hydropower sector. These tools could help identify potential market fluctuations and support proactive risk mitigation efforts.

5. **Portfolio Optimization Platforms:** The PPO-based strategy could be integrated into investment platforms, enabling users to create and manage diversified portfo-

lios that align with their risk tolerance and investment goals, thereby improving portfolio performance and risk management.

6. **Academic Research and Education:** The outcomes of this research could contribute to a better academic understanding of how DRL can be applied in financial markets. This knowledge could support the development of educational programs and research initiatives focused on the intersection of AI and finance.

1.7 Originality of Project

The originality of this project lies in the development and application of a novel Proximal Policy Optimization (PPO) based trading strategy specifically tailored for the hydro power sector within the Nepal Stock Exchange (NEPSE), a context that has been largely unexplored in existing research.

1.8 Organization of Project Report

This project report is structured into six sections, each with a specific focus. In the Introduction, we define the central problem and objectives, setting the stage for our research. Next, the Literature Review explores existing research to pinpoint gaps and opportunities for innovation. The Methodology section provides a comprehensive overview of the methods and steps used in the project. The Results section presents the findings and outcomes of our work. In the Discussion and Analysis, we interpret the results and their implications. Finally, the Remaining Tasks section outlines any additional work needed to be done.

2 LITERATURE REVIEW

The literature review is an important step in creating trading strategies using deep reinforcement learning (DRL) for the hydropower sector in the Nepal Stock Exchange (NEPSE). It helps build a clear understanding of how predictive modeling works in financial markets. By looking at previous research, we can learn about different methods, models, and factors that affect stock price predictions.

This review specifically examines DRL methods to understand their effectiveness, challenges, and current trends in predicting hydropower stock prices in NEPSE. It aims to lay the groundwork for developing a strong and effective trading model, advancing financial forecasting methods, and improving decision-making in the Nepalese stock market.

2.1 Design of Stock Trading Agent Using Deep Reinforcement Learning

The study by Janak Kumar Lal explores using Double Deep Q-Learning (DDQL) to develop a trading strategy for stocks on the Nepal Stock Exchange (NEPSE).[3] This research merges reinforcement learning with technical indicators to enhance trading strategies and boost profitability on NEPSE.

Lal's study demonstrates that a DDQL agent trained on NEPSE stock data significantly outperforms baseline models in various scenarios. The DDQL model excels in identifying profitable trading strategies by recognizing patterns in stock price data.

It incorporates multiple technical indicators, such as the Stochastic Oscillator, On Balance Volume (OBV), Moving Average Convergence Divergence (MACD), and the Average Directional Index (ADX), to detect trading signals and trends in the market.

Additionally, the model uses two neural networks (policy and target networks) to reduce overestimation bias and improve decision-making accuracy. Empirical results show that the DDQL agent achieves higher net worth and profitability compared to baseline models across different NEPSE stock data sets.

This research effectively combines various technical indicators, utilizes an advanced DDQL framework, and includes thorough empirical validation, highlighting the agent's efficiency. Lal's study offers valuable insights into applying deep reinforcement learning for stock trading on NEPSE, highlighting both the potential benefits and challenges of such advanced trading strategies.

2.2 Deep reinforcement learning approach for trading automation in the stock market

The paper by Kabbani and Duman investigates how the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm can enhance automated stock trading. [4] They frame the trading problem as a Partially Observed Markov Decision Process (POMDP), combining prediction and decision-making for fully automated trading.

Kabbani finds that the TD3 algorithm significantly improves trading automation by effectively integrating price prediction with portfolio allocation while considering market constraints. This approach helps optimize returns and manage risks better.

The TD3 model addresses both prediction and decision-making challenges, offering a more integrated and efficient method for automated trading. The results demonstrate that the TD3 algorithm achieves a high Sharpe Ratio of 2.68 on the test data, reflecting strong performance in risk-adjusted returns.

The study also takes into account market constraints like liquidity and transaction costs, which are often ignored in traditional models. The TD3 algorithm excels by balancing prediction accuracy with decision-making for portfolio management, showing its robustness in achieving overall trading objectives.

Overall, this research provides important insights into using deep reinforcement learning for stock market trading, emphasizing the benefits and challenges of advanced trading strategies.

2.3 Adaptive Stock Trading Strategies with Deep Reinforcement Learning Methods

The paper by Xing Wu, Haolei Chen, Jianjia Wang, Luigi Troiano, Vincenzo Loia, and Hamido Fujita explores advanced stock trading strategies using deep reinforcement learning (DRL). The study focuses on creating adaptive trading strategies by employing Gated Recurrent Units (GRUs) to analyze raw financial data and technical indicators.[5]

By combining GRUs with DRL techniques like Gated Deep Q-Network (GDQN) and

Gated Deep Policy Gradient (GDPG), the researchers show that these methods outperform traditional trading approaches and offer more stable returns in volatile markets.

The results reveal that GDQN and GDPG strategies exceed the performance of the Turtle trading strategy and other advanced methods in terms of stability and returns, especially in fluctuating markets.

These strategies performed well across different markets, including the U.S., U.K., and China, with GDPG showing better stability and fewer losses compared to GDQN.

The self-learning capabilities of these strategies allow them to adapt effectively to changing market conditions, making them highly effective for high-frequency trading. This research highlights the potential of DRL for developing robust and adaptable stock trading strategies.

2.4 Stock price forecast based on combined model of ARI-MA-LS-SVM

The research conducted by Xiao et al. in 2020 investigates how combining different forecasting models can enhancethe prediction of stock market trends compared to using individual models alone. [6] The study introduces an innovative approach that merges the Cumulative Auto-Regressive Moving Average (ARI-MA) with the Least Squares Support Vector Machine (LS-SVM) into a unified forecasting model known as ARI-MA-LS-SVM.

To develop this new model, the researchers first analyze the shortcomings of existing forecasting methods and standard SVMs. They identify specific limitations in these approaches, which lead to the creation of the ARI-MA-LS-SVM model.

This model improves forecasting by preprocessing data using cumulative auto-regressive moving averages and then applying LS-SVM to predict stock price movements based on fundamental indicators.

The results from the study show that the ARI-MA-LS-SVM model significantly outperforms individual forecasting models. It achieves higher accuracy and better applicability across various market conditions. The model's effectiveness is further validated through simulation tests, which confirm its stability and relevance in different market environments.

This research provides important insights for both investors and market regulators. By advancing stock market forecasting techniques, the ARI-MA-LS-SVM model contributes to more accurate predictions and improved decision-making processes.

It highlights the benefits of integrating multiple forecasting methods to address the complexities of financial markets and offers practical guidance for enhancing investment strategies.

2.5 Performance of Deep Learning in Prediction of Stock Market Volatality

In their 2019 study, Moon and Kim use the Long Short-Term Memory (LSTM) deep learning algorithm to develop a method for predicting stock market indices and their volatility. They tested their approach on data from five major stock market indices—the S&P 500, NASDAQ, German DAX, Korean KOSPI200, and Mexico IPC—covering a period of seven years (2010-2016). Their research demonstrates that the LSTM algorithm is effective in making these predictions.[7]

The study finds that the best results for forecasting both market index values and volatility come from using a hybrid momentum approach. This technique calculates the difference between the current price and the moving average of past prices.

One key insight from their research is that predicting stock index values relies on a variety of financial factors, including opening prices, low and high prices, and trading volume. In contrast, predicting volatility depends primarily on past volatility itself, rather than other financial variables.

This difference emphasizes the distinct nature of volatility forecasting and suggests that specialized models like LSTM can offer valuable insights. Moon and Kim's research contributes significantly to the field of financial forecasting by showing how deep learning techniques can improve predictions of market volatility and enhance investment decision-making.

Their work helps to understand the complexities of financial markets and demonstrates the potential of advanced algorithms for making more accurate forecasts.

2.6 Research Gap

The literature on stock market prediction using Deep Reinforcement Learning reveals several notable research gaps that provide opportunities for further investigation. The research gap lies in the scarcity of studies focusing on assessing the performance and adaptability of the PPO algorithm in the context of NEPSE's volatility and relatively lower liquidity, particularly in the hydro power sector which might be prone to sudden market shifts.

3 METHODOLOGY

3.1 Theoretical Formulations

3.1.1 Basic Concept of Proximal Policy Optimization(PPO Algorithm)

The PPO algorithm, a state-of-the-art reinforcement learning technique, serves as the cornerstone of our methodology. PPO operates within the framework of DRL, which combines deep neural networks with reinforcement learning principles to enable agents to learn optimal policies through interaction with their environment. Within our context, the environment encompasses the dynamics of the NEPSE market, including historical market data.

PPO is an actor-critic method where the ”actor” is responsible for selecting actions, and the ”critic” evaluates those actions based on a value function. The algorithm optimizes the policy by performing multiple epochs of gradient descent on a clipped objective function, balancing exploration and exploitation efficiently.

For Nepse stock trading agent, the PPO algorithm seeks to maximize the expected cumulative reward, which the net profit. The agent interacts with the stock market environment by taking actions (buy, sell and hold) based on the observed state aiming to learn a policy that yields the best possible trading strategy.

PPO aims to efficiently optimize the policy of an agent in an environment by balancing exploration and exploitation, while ensuring stability during training.

Policy Exploration and Exploitation: PPO begins by allowing the agent to explore various actions in the environment to gather experience. As training progresses, the agent learns to exploit the knowledge gained from exploration to select actions that maximize its expected return or rewards.

Objective of Maximizing Total Rewards: The primary objective of the agent is to maximize its total rewards across a series of states or episodes in the environment which is maximizing the net profit. This is achieved by learning a policy function that maps states to actions, allowing the agent to make decisions based on its observations.

3.1.2 Benefits of Proximal Policy Optimization Method

The selection of PPO within the DRL paradigm offers several key advantages:

1. **Adaptability**: PPO excels in environments with complex and uncertain dynamics, making it well-suited for the inherently volatile nature of financial markets.
2. **Policy Improvement**: By iteratively optimizing policy parameters, PPO facilitates the discovery of robust and adaptive trading strategies, capable of capitalizing on evolving market conditions.
3. **Sample Efficiency:** PPO's efficient use of data allows for faster convergence and reduced computational resources compared to alternative reinforcement learning algorithms.
4. **Generalization:** The learned policies have the potential to generalize across diverse market scenarios, enhancing their applicability to different market conditions and sectors.
5. **Stability and Reliability:** PPO is known for its stable training process. Unlike other policy gradient methods that can exhibit erratic behavior due to high variance in the updates, PPO introduces a clipped objective function that prevents drastic changes to the policy, ensuring smoother and more reliable learning.
6. **Versatility:** The algorithm is flexible and can be applied to various types of financial markets and assets. Its adaptability allows it to perform well in both trending and mean-reverting market conditions.
7. **Robustness to Hyperparameters:** PPO is less sensitive to hyperparameter tuning compared to other reinforcement learning algorithms, reducing the need for extensive experimentation to find optimal settings.

3.1.3 Assumptions Taken into Account

In developing and deploying the PPO algorithm for trading hydropower stocks listed in NEPSE , several key assumptions are made to ensure the model's effectiveness and adaptability in real-world scenarios. These assumptions help define the scope and limitations of the model, providing a framework within which the algorithm operates.

1. Market Efficiency Assumption

We assume that the market is efficient enough such that historical data can provide meaningful insights for predicting future stock prices. It relies on the assumption that any predictable patterns or trends in the data will be captured by the algorithm.

2. Stationarity of Data

It is assumed that the statistical properties of the data (such as mean and variance) are relatively stable over time. This means that historical patterns are assumed to continue into the future, which might not always hold true in real-world scenarios.

3. Complete and Accurate Data

We assume that the historical data used for training the model is complete, accurate, and free from significant errors. Any gaps or inaccuracies in the data could affect the model's performance.

4. Action Constraints

It is also assumed that the trading actions (e.g., buying or selling) can be executed without slippage or significant market impact. This assumption may not hold in less liquid markets where large trades can influence stock prices.

By using the PPO algorithm in a stock trading strategy, the model aims to develop an optimal trading policy that maximizes returns while managing risk, leveraging the algorithm's inherent stability and efficiency to adapt to various market conditions.

3.2 Mathematical Modelling

3.2.1 PPO Algorithm

The Proximal Policy Optimization (PPO) algorithm is a reinforcement learning technique used to train policies in environments with continuous action spaces. Here's a simplified explanation of how PPO works.

1. **Initialize Parameters**:
   - Initialize the parameters of the policy network θ and the value network ϕ.
   - Set the hyperparameters such as the learning rate, clipping parameter ε,

discount factor γ, and the number of epochs.

2. **Collect Trajectories**:

   - Run the current policy π_θ in the environment to collect a set of trajectories τ. Each trajectory is a sequence of states, actions, rewards, and possibly next states: $\tau = \{(s_t, a_t, r_t, s_{t+1})\}_{t=0}^{T}$.

3. **Compute Rewards and Advantages**:

   - Calculate the cumulative discounted rewards for each trajectory:

   $$R_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k} \tag{3.1}$$

   - Compute the advantage estimates using the value function $V_\phi(s_t)$:

   $$A_t = R_t - V_\phi(s_t) \tag{3.2}$$

4. **Calculate the Probability Ratios**:

   - For each action taken during the trajectories, compute the probability ratio between the new and old policy:

   $$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \tag{3.3}$$

5. **Compute the Clipped Objective**:

   - Define the clipped surrogate objective function:

   $$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)A_t\right)\right] \tag{3.4}$$

   - The function $\text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)$ ensures that the ratio $r_t(\theta)$ does not deviate significantly from 1, thereby preventing large updates.

6. **Update the Policy Network**:

- Use gradient descent to maximize the clipped objective with respect to the policy parameters θ:

$$\theta \leftarrow \theta + \alpha \nabla_\theta L^{\text{CLIP}}(\theta) \tag{3.5}$$

- Here, α is the learning rate.

7. **Update the Value Network**:

    - Minimize the mean squared error between the estimated value and the cumulative rewards:

    $$L^{\text{VF}}(\phi) = \mathbb{E}_t \left[(V_\phi(s_t) - R_t)^2 \right] \tag{3.6}$$

    - Update the value network parameters ϕ using gradient descent:

    $$\phi \leftarrow \phi - \beta \nabla_\phi L^{\text{VF}}(\phi) \tag{3.7}$$

    - Here, β is the learning rate for the value network.

8. **Repeat**:

    - Repeat the process from step 2 to step 7 for a specified number of iterations or until convergence.

The clipping mechanism in PPO ensures that the policy updates are conservative and stable, preventing large and destabilizing changes to the policy. This approach allows PPO to achieve a balance between exploration and exploitation, making it effective for a wide range of reinforcement learning tasks.

3.2.2 Technical Indicators

In the pre-processing step, we use technical indicators to feed to the model. Traders use technical indicators to study past price movements and trading activity (volume and open interest) of an asset. These indicators help them estimate future price changes by finding patterns and trends in the historical data. By feeding the values of technical indicators to the model the model can predict the future price and decide whether to buy, hold or sell the stock.

Simple Moving Average (SMA)

A Simple Moving Average smooths out price data by creating a constantly updated average price. The SMA is useful for identifying the direction of the trend and smoothing out price fluctuations.

Simple Moving Averages (SMAs) help to identify support and resistance levels, making them useful for spotting potential reversal points. Commonly used time periods for SMAs are the 50-day and 200-day SMAs. A rising SMA indicates an uptrend, suggesting that the asset's price is generally increasing, while a falling SMA indicates a downtrend, suggesting that the asset's price is generally decreasing.

The Simple Moving Average is calculated by averaging the closing prices of a security over a specific number of periods n.

Formula:

$$SMA = \frac{P_1 + P_2 + \ldots + P_n}{n} \tag{3.8}$$

where:

- SMA is the Simple Moving Average.
- $P_1, P_2, \ldots, P_n$ are the closing prices of the security for n periods.

For example, if you want to calculate a 5-day SMA Add up the closing prices of the last 5 days and then divide the total by 5.

This calculation provides a smoothed average price that helps traders identify trends and potential support/resistance levels in the market.

Exponential Moving Average (EMA)

The Exponential Moving Average is a type of moving average that gives more weight to recent prices, making it more responsive to new information. The EMA reacts more significantly to recent price changes than the SMA.

Exponential Moving Averages (EMAs) are commonly used with time periods of 12 days and 26 days. The EMA is calculated as:

Formula:

$$\text{EMA}_t = \left(P_t \times \frac{2}{n+1}\right) + \left(\text{EMA}_{t-1} \times \left(1 - \frac{2}{n+1}\right)\right) \tag{3.9}$$

where:

- EMA_t is the current EMA value.
- P_t is the current price.
- n is the number of periods.
- EMA_{t-1} is the previous EMA value.

Moving Average Convergence Divergence (MACD)

The MACD is a momentum indicator that follows trends and shows the relationship between two moving averages of a stock's price. It consists of the MACD line (the difference between two EMAs), the signal line (an EMA of MACD line), and a histogram that represents the difference between the MACD line and the signal line.

The Moving Average Convergence Divergence (MACD) produces buy signals when the MACD line moves above the signal line, and it generates sell signals when the MACD line falls below the signal line. Additionally, the histogram helps to identify the strength of the trend.

Formula:

Formula:

$$\text{MACD Line} = \text{EMA}_{12} - \text{EMA}_{26} \tag{3.10}$$

$$\text{Signal Line} = \text{EMA}_9 \text{ of MACD Line} \tag{3.11}$$

where:

- EMA_{12} is the 12-period Exponential Moving Average.
- EMA_{26} is the 26-period Exponential Moving Average.
- EMA_{9} is the 9-period Exponential Moving Average.

The MACD Line is calculated as the difference between the 12-period EMA and the 26-period EMA. The Signal Line is a 9-period EMA of the MACD Line.

Bollinger Bands

Bollinger Bands consist of a middle band (SMA) and two outer bands that are set a certain number of standard deviations above and below the middle band. They provide a relative definition of high and low prices.

Bollinger Bands indicate that the market is overbought when prices move towards the upper band and oversold when prices move towards the lower band. They help to identify volatility and potential price breakouts.

Formula:

$$\text{Middle Band (SMA)} = \frac{1}{n}\sum_{i=1}^{n} P_i \tag{3.12}$$

$$\text{Upper Band} = \text{Middle Band} + 2 \times \text{Standard Deviation} \tag{3.13}$$

$$\text{Lower Band} = \text{Middle Band} - 2 \times \text{Standard Deviation} \tag{3.14}$$

where:

- P_i represents the closing price for the i-th period.
- n is the number of periods used for the Simple Moving Average (SMA).
- Standard Deviation is calculated based on the closing prices over n periods.

Stochastic Oscillator

The Stochastic Oscillator is a momentum indicator that compares a security's closing price to its price range over a given period of time. It consists of two lines: %K and %D.

The Stochastic Oscillator ranges in value from 0 to 100. A reading above 80 indicates that the asset is overbought, while a reading below 20 indicates that it is oversold. Buy signals occur when the %K line crosses above the %D line in the oversold region, and sell signals occur when the %K line crosses below the %D line in the overbought region.

Formula:

$$\%K = \left(\frac{C - L_n}{H_n - L_n} \right) \times 100 \tag{3.15}$$

$$\%D = \text{3-period moving average of \%K} \tag{3.16}$$

where:

- C is the current closing price.
- H_n is the highest high over the last n periods.
- L_n is the lowest low over the last n periods.
- $\%K$ is the Stochastic Oscillator.
- $\%D$ is the Signal Line, typically a 3-period Simple Moving Average of $\%K$.

On-Balance Volume (OBV)

On-Balance Volume (OBV) is a momentum indicator that reflects market sentiment based on volume flow. It measures cumulative buying and selling pressure.

A rising OBV suggests that buyers are actively pushing the price higher, indicating strong bullish momentum. Conversely, a falling OBV suggests that sellers are dominating, potentially pushing the price lower, indicating bearish sentiment.

Traders often look for divergences between OBV and price movements as they can signal potential trend reversals, highlighting shifts in buying or selling pressure that may precede changes in the direction of the asset's price.

Formula:

$$\mathrm{OBV}_t = \begin{cases} \mathrm{OBV}_{t-1} + \mathrm{Volume}_t & \text{if } \mathrm{Close}_t > \mathrm{Close}_{t-1} \\ \mathrm{OBV}_{t-1} - \mathrm{Volume}_t & \text{if } \mathrm{Close}_t < \mathrm{Close}_{t-1} \\ \mathrm{OBV}_{t-1} & \text{if } \mathrm{Close}_t = \mathrm{Close}_{t-1} \end{cases} \tag{3.17}$$

where:

- OBV_t is the On-Balance Volume at time t.
- Volume_t is the trading volume at time t.
- Close_t is the closing price at time t.
- Close_{t-1} is the closing price at the previous time $t-1$.
- OBV_{t-1} is the On-Balance Volume at the previous time $t-1$.

Relative Strength Index (RSI)

The Relative Strength Index is a momentum oscillator that measures the speed and change of price movements. It ranges from 0 to 100 and is typically used to identify overbought or oversold conditions.

An RSI above 70 suggests that the asset may be overbought, while an RSI below 30 suggests that the asset may be oversold. RSI can be used to identify potential reversal points and confirm trend strength.

Formula:

$$\text{RSI} = 100 - \frac{100}{1+\text{RS}} \tag{3.18}$$

$$\text{RS} = \frac{\text{Average Gain}}{\text{Average Loss}} \tag{3.19}$$

where:

- RSI is the Relative Strength Index.
- RS is the Relative Strength, calculated as the ratio of Average Gain to Average Loss over a specified period.
- Average Gain is the average of all upward price changes over the specified period.
- Average Loss is the average of all downward price changes over the specified period.

Average Directional Index (ADX)

The Average Directional Index (ADX) is a technical indicator used to measure the strength of a trend in the market, regardless of its direction. The ADX is derived from the Directional Movement Index (DMI), which consists of two indicators: the Positive Directional Indicator (+DI) and the Negative Directional Indicator (-DI).

1. **Calculate the True Range (TR)**

   The True Range is defined as the greatest of the following three values:

$$\begin{aligned}\text{TR} = \max(&\text{Current High} - \text{Current Low},\\ &|\text{Current High} - \text{Previous Close}|,\\ &|\text{Current Low} - \text{Previous Close}|)\end{aligned} \tag{3.20}$$

2. **Determine the Directional Movements (+DM and -DM)**

Calculate the Positive Directional Movement (+DM) and the Negative Directional Movement (-DM) using the following formulas:

$$+DM = \begin{cases} \text{Current High} - \text{Previous High}, & \text{if (Current High} - \text{Previous High)} \\ & > \text{(Previous Low} - \text{Current Low)} \\ & \text{and (Current High} - \text{Previous High)} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.21}$$

$$-DM = \begin{cases} \text{Previous Low} - \text{Current Low}, & \text{if (Previous Low} - \text{Current Low)} \\ & > \text{(Current High} - \text{Previous High)} \\ & \text{and (Previous Low} - \text{Current Low)} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.22}$$

3. **Calculate the Smoothed True Range (ATR)**

   Compute the Average True Range (ATR), which is a smoothed moving average of the True Range (TR) over a specified period, typically 14 periods.

4. **Calculate the Directional Indicators (+DI and -DI)**

   The Positive Directional Indicator (+DI) and the Negative Directional Indicator (-DI) are calculated as follows:

$$+DI = \left(\frac{\text{Smoothed +DM}}{\text{ATR}}\right) \times 100 \tag{3.23}$$

$$-DI = \left(\frac{\text{Smoothed -DM}}{\text{ATR}}\right) \times 100 \tag{3.24}$$

5. **Compute the Directional Index (DX)**

   The Directional Index (DX) measures the absolute difference between +DI and

-DI, normalized by their sum:

$$\text{DX} = \left(\frac{|+DI - -DI|}{+DI + -DI} \right) \times 100 \tag{3.25}$$

6. **Calculate the ADX**

   Finally, the Average Directional Index (ADX) is obtained by taking the exponential moving average (EMA) of the DX values over a specified period, typically 14 periods:

$$\text{ADX} = \text{EMA of DX over 14 periods} \tag{3.26}$$

The ADX provides a quantifiable measure of trend strength, where values above 25 generally indicate a strong trend, while values below 20 suggest a weak trend or a market without a clear direction.

3.3 System Block Diagram

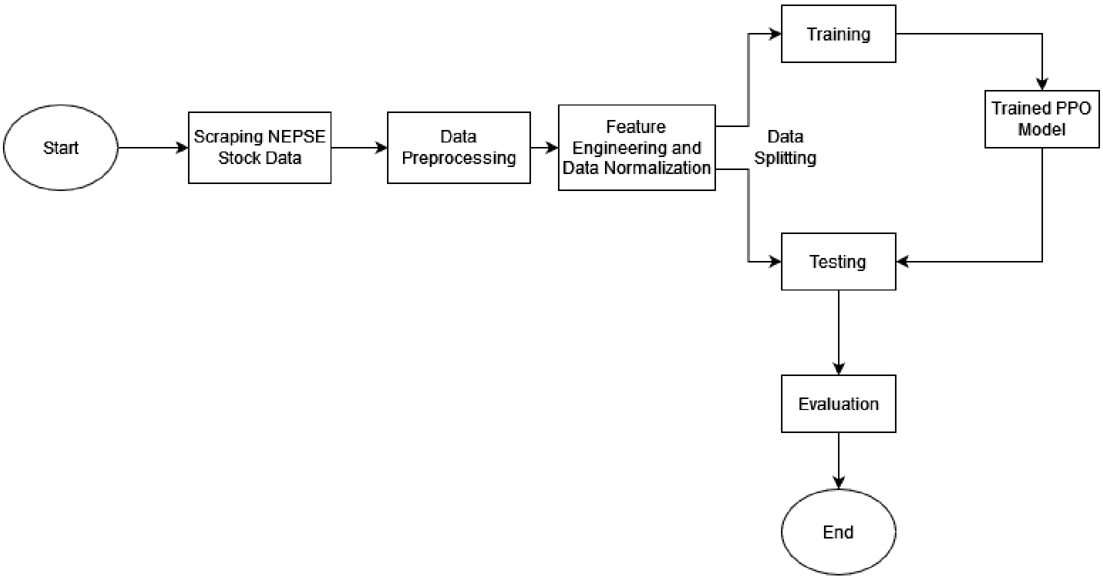

Figure 3.1: System Block Diagram

Figure 3.1 shows a block diagram of how the model is proposed starting from the data collection to model evaluation. All the steps involved are described below.

3.3.1 Scraping Stock Data

Data collection is a crucial initial step in constructing predictive models for stock market index prediction. We gather historical price data of stocks in the Hydropower Sector of the Nepal Stock Exchange (NEPSE) from Sharesansar.com. This data provides insights into past stock price movements and trends, serving as the foundation for our predictive modeling efforts.

3.3.2 Data Pre-Processing

Once the data is collected, it undergoes preprocessing to ensure its suitability for analysis. This involves cleaning the data to rectify inconsistencies and errors, and handling missing values through imputation or deletion.

1. **Data Cleaning** Data cleaning is the process of fixing or removing incorrect, incomplete, or messy data to make sure it's accurate and usable for analysis or building models. This involves tasks like filling in missing information, correcting mistakes, getting rid of duplicates, and making sure everything is in the right format. The goal is to have clean, consistent data that leads to more reliable results when you analyze it or use it to train a model.

2. **Feature Engineering**

   Feature engineering is employed to create new variables or transform existing ones, enhancing the model's predictive power. This step enriches the data and improves the model's ability to make accurate predictions.

   We create multiple technical indicators as features for the model such as SMA, EMA, RSI, MACD etc.

3. **Data Normalization**

   Data normalization is performed to bring variables to a common scale, preventing features with larger magnitudes from dominating the model. For our model we use Standard Scalar to Normalize the data. These steps are critical for preparing the data for subsequent modeling stages and ensuring the reliability of the predictive models.

3.3.3 Training

For this study, we created two cases:

Case I:

First was with the dataset partitioned such that the training set encompasses historical data for 91 Hydropower stock up to May 8, 2024. This period is selected to provide a comprehensive view of past stock behavior, capturing a variety of market conditions and trends.

Case II:

For as second case we selected the hydropower stocks with more than 5 years of data to test in the last 3 years and keep the remaining in the training set. So, there were total of 21 hydropower stock in this case where we trained on dataset till 29th December, 2021. This was done to check the performance of the model in various conditions of the market.

During the training phase, the model is exposed to this subset of historical data, enabling it to learn from and adapt to the complexities of the stock market. This approach allows the model to build a nuanced understanding of the underlying patterns and dynamics that influence stock prices. By leveraging a rich dataset that includes various market scenarios, the model aims to develop a policy that can make informed and effective trading decisions.

The core of the training process involves employing advanced reinforcement learning techniques. Specifically, the Proximal Policy Optimization (PPO) Algorithm is utilized. PPO is renowned for its efficiency and stability in optimizing policy parameters, making it well-suited for complex decision-making tasks such as stock trading.

By the end of the training phase, the model is expected to have a well-optimized policy that can be tested against a separate testing set to validate its performance. The goal is to ensure that the model not only performs well on historical data but also demonstrates robust decision-making capabilities in unseen market conditions. This comprehensive training and evaluation process is crucial for developing a reliable and effective trading model.

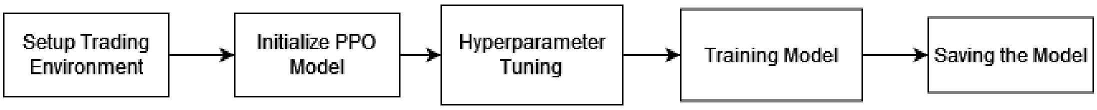

Figure 3.2: System Block Diagram Training

3.3.4 Testing

The testing phase is a critical component in evaluating the effectiveness and robustness of the trading model. For this purpose, data from May 9, 2024, to July 29, 2024, is utilized to assess the model's performance on a separate dataset that was not used during the training phase. This period is chosen to provide a fresh set of market conditions, allowing for a comprehensive evaluation of the model's ability to generalize beyond the training data.

Similarly, for the second case of the testing, data from 2nd January, 2022 till 29th July, 2024 was taken to allow the model to test in various market condition.

During the testing phase, the model is subjected to real-world scenarios through a process of simulated trading on the unseen data. This enables us to observe how well the model adapts to and performs in novel market environments, which is essential for validating its practical applicability. The testing dataset includes various market dynamics and fluctuations that the model has not previously encountered, providing insights into its robustness and adaptability.

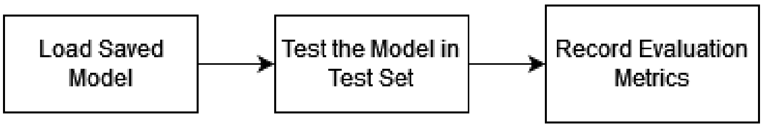

Figure 3.3: System Block Diagram Testing

3.3.5 Evaluation

Evaluation involves assessing the model's predictive accuracy, profitability, and robustness using various metrics and techniques. Comprehensive evaluation helps determine the model's viability for real-world stock trading applications and identifies areas for improvement.

The steps in evaluating the agent include;

1. **Calculating Performance Metrics:**

   After developing a trading agent, evaluating its performance is crucial to understand its effectiveness and reliability. Performance metrics help measure how well the strategy has worked and identify areas for improvement. Some of the metrics that we use are Sharpe Ratio, Annualized Return, Cumulative Return, and Portfolio Value.

2. **Backtesting:**

   Backtesting involves applying the trading strategy to historical data to assess how it would have performed in the past. This helps validate the strategy's effectiveness and uncover potential issues before using it in real trading.

   Historical data is used to simulate trading decisions, and the resulting performance is compared against expected outcomes. It helps identify any weaknesses in the strategy and makes adjustments as needed.

These steps are fundamental in developing and refining trading strategies, ensuring they are robust, effective, and ready for real-world application.

3.4 Instrumentation Requirement

For this project, a combination of hardware and software tools is required to facilitate data collection, model development, and evaluation.

3.4.1 Hardware Tools:

Python 3 Google Compute Engine Backend: Google Colab GPU with an Intel Xeon CPU @2.20 GHz, 13 GB RAM, a Tesla K80 accelerator, and 12 GB GDDR5 VRAM.

Lenovo Legion Legion5 15ARH05: The device used will have AMD Ryzen 5 4600H processor, 8GB DDR4 RAM, and dual storage with a 1TB HDD and 256GB SSD.It has a NVIDIA GeForce GTX 1650 GPU with 4GB GDDR6 memory,

3.4.2 Software Tools:

Python Programming Language: Python serves as the primary programming language for implementing machine learning algorithms, data preprocessing, and model evaluation. Libraries such as NumPy, Pandas, and Scikit-learn are utilized for data manipulation and model development.

Octoparse Web Scraping Tool: Octoparse is web scraping tool that is used for collecting share market price data from sharesansar.com.

Deep Learning Frameworks: PyTorch is used for building and training deep learning models, particularly PPO Algorithm for stock price prediction. We specifically use stable baselines 3 which is a set of reliable implementations of reinforcement learning algorithms in PyTorch to create and train the model.

In this project, both hardware and software tools will be accessed through the free version of Google Colab, providing access to a comprehensive suite of resources essential for building and evaluating the predictive model.

3.4.3 Access and Utilization:

Access to both hardware and software tools for model training will be obtained through Google Colab's free-tier plan, providing researchers with a cost-effective solution for conducting machine learning experiments. Users can connect to Google Colab's virtual environment via a web browser, where they can access the allocated CPU resources and utilize the built-in software tools for model development.

By leveraging Google Colab's free version, researchers can benefit from scalable computing resources and a rich ecosystem of machine learning tools without the need of dedicated hardware or software installations. This approach enables efficient experimentation and prototyping of machine learning models, facilitating the exploration of various algorithms and techniques for predicting stock market indices accurately.

3.5 Dataset Explanation

For this project, the dataset will be sourced from Sharesansar, a leading financial portal in Nepal, known for providing comprehensive information on the Nepalese stock market.

Table 3.1 shows the sample dataset that can be used for model training. The dataset contains essential stock market metrics for the stock, including Open, High, Low, Last Trading Price, Change, Turnover and Date of Arun Valley Hydropower Development Company Limited (AHPC). These metrics capture crucial aspects of daily market activity, such as the opening and closing prices of the index, the highest and lowest prices reached during the trading day, the change in index value, and the corresponding turnover volume. Additionally, the dataset includes the date associated with each trading day, allowing for chronological analysis of market trends over time.

Table 3.1: Sample Stock Price Dataset of AHPC

Date	Open	High	Low	Ltp	% Change	Qty	Turnover
2024-06-02	160.00	160.00	156.40	157.00	-1.20	177,319.00	27,960,202.10
2024-05-30	161.30	163.00	158.00	158.90	-1.49	172,008.00	27,470,999.70
2024-05-29	165.00	166.00	161.10	161.30	-0.80	200,870.00	32,775,237.30
2024-05-27	165.30	167.00	162.10	162.60	-1.33	191,749.00	31,276,705.00
2024-05-26	166.00	166.00	161.10	164.80	0.61	272,417.00	44,369,040.00
2024-05-22	163.00	166.00	161.20	163.80	1.74	323,763.00	53,071,154.50
2024-05-21	161.80	163.00	150.00	161.00	-1.83	571,540.00	91,786,718.70
2024-05-20	163.00	166.40	162.00	164.00	1.17	74,008.00	12,207,544.50
2024-05-19	162.00	164.00	154.00	162.10	0.06	50,807.00	8,195,061.10
2024-05-16	168.30	168.30	161.70	162.00	-1.82	16,093.00	2,626,050.80
2024-05-15	165.00	171.00	164.00	165.00	-0.54	36,594.00	6,103,579.00
2024-05-14	160.10	167.00	158.00	165.90	5.00	40,183.00	6,494,293.10
2024-05-13	158.20	160.00	156.10	158.00	1.35	39,301.00	6,226,427.30
2024-05-12	153.90	160.00	153.90	155.90	-0.70	40,482.00	6,305,438.90
2024-05-09	162.80	162.80	156.00	157.00	-1.69	49,152.00	7,746,254.10
2024-05-08	162.50	162.50	157.00	159.70	-0.37	23,064.00	3,659,894.75

3.6 Description of Algorithms

3.6.1 Algorithm Description of Data Collection

Algorithm 1 Web Scraping Algorithm for sharesansar.com

Require: *base_url* ← “https://www.sharesansar.com/company-list”

Ensure: Excel files with company, date, open, high, low, LTP, % change, qty, turnover

```
Navigate to base_url
dropdown ← find_dropdown()
select_hydropower ← select_from_dropdown(dropdown, ”Hydropower”)
company_list ← get_company_list()
while not end of company_list do
    for all company $\in$ company_list do
        company_page ← click_company(company)
        price_history ← click_price_history(company_page)
        data ← initialize_empty_table()
        while not end of price_history do
            table ← extract_price_history_table(price_history)
            data ← append_to_table(data, table)
            price_history ← click_next_page(price_history)
        end while
        save_to_excel(data, company_name)
        company_list ← navigate_back()
    end for
    company_list ← click_next_page(company_list)
end while
```

This pseudocode outlines the process of web scraping data from `sharesansar.com`, specifically targeting hydropower companies. It starts by navigating to the company list page and selecting ”Hydropower” from a dropdown menu to filter the companies displayed. The algorithm then iterates through each company, navigating to its detail page and accessing the ”Price History” section. It extracts key data, including Date, Open, High, Low, LTP, % Change, Qty, and Turnover.

For each company, the extracted data is saved into an Excel file named after the company.

The algorithm handles pagination in both the company list and the price history tables, ensuring that all available data is captured and stored efficiently. This structured approach ensures comprehensive data collection from the specified category on the website.

3.6.2 Algorithm Description of PPO Algorithm

Algorithm 2 Proximal Policy Optimization (PPO)

```
Input: Initial policy $\theta_0$, value function parameters $\phi_0$
Hyperparameters: Clip parameter $\varepsilon$, learning rates $\alpha_\theta$, $\alpha_\phi$, number of epochs $K$, mini-batch size $M$
for each iteration do
    Collect Trajectories: Run policy $\pi_\theta$ for $T$ timesteps and collect trajectories $\{\tau_i\}$, where $\tau_i = \{(s_t, a_t, r_t, s_{t+1})\}_{t=0}^{T}$.
    Step 2: Compute Advantages
    Estimate returns $R_t = \sum_{l=0}^{T-t} \gamma^l r_{t+l}$
    Compute advantages $\hat{A}_t = R_t - V_\phi(s_t)$
    Update Policy:
    for each epoch $k$ do
        Sample mini-batch of trajectory, compute ratio $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ and clipped objective:
            $L^{\text{CLIP}}(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t\right)\right]$
        Update policy: $\theta \leftarrow \theta + \alpha_\theta \nabla_\theta L^{\text{CLIP}}(\theta)$.
    end for
    Step 4: Update Value Function
    Update value function parameters by minimizing the loss:
        $L^{\text{VF}}(\phi) = \mathbb{E}\left[(V_\phi(s_t) - R_t)^2\right]$
    Update value function parameters:
        $\phi \leftarrow \phi + \alpha_\phi \nabla_\phi L^{\text{VF}}(\phi)$
end for
```

The pseudo code provided is for the Proximal Policy Optimization (PPO) algorithm, a reinforcement learning method that optimizes policies to maximize cumulative rewards. Below is an explanation of each step in the algorithm:

1. **Initialization**

   The Proximal Policy Optimization (PPO) algorithm begins with the initialization of two key components: the policy parameters and the value function parameters. The policy parameters, denoted as θ_0, define the agent's behavior in the environment. Similarly, the value function parameters ϕ_0 are initialized to estimate the expected cumulative rewards from any given state. This initialization sets the starting point for the learning process, allowing the agent to begin interacting with the environment and gathering data.

2. **Collect Trajectories (Exploration)**

   In the exploration phase, the agent uses its current policy π_θ to interact with the environment over a predefined number of timesteps, denoted as T. During this interaction, the agent collects trajectories, which are sequences of state-action-reward-next state tuples. These trajectories represent the agent's experiences and provide the raw data needed for learning.

3. **Compute Advantages**

   After collecting trajectories, the next step is to compute the advantages, which measure how much better or worse an action was compared to the expected outcome. First, the returns R_t are calculated as the total accumulated reward from a given time t until the end of the episode, discounted by a factor γ. This helps quantify the actual rewards received by the agent. The advantages $\hat{A}_t$ are then computed as the difference between these returns and the value function's estimate of the state value $V_\phi(s_t)$. This advantage estimation is essential for understanding the relative benefit of actions taken.

4. **Update Policy (Policy Optimization)**

   The policy update step is where PPO's core innovation comes into play. The algorithm computes the probability ratio $r_t(\theta)$, which is the ratio of the probability of taking an action under the current policy to the probability under the old policy. This ratio is used to calculate the clipped objective function, $L^{\text{CLIP}}(\theta)$, which includes a clipping mechanism to limit how much the policy can change. By using this clipped objective, PPO ensures that updates to the policy are constrained, preventing large deviations that could destabilize learning. The policy parameters θ are then updated using stochastic gradient descent based on this objective, refining the policy in a stable and controlled manner.

5. **Update Value Function (Critic Optimization)**

   In parallel with updating the policy, the value function also needs to be refined. The value function's loss, $L^{\text{VF}}(\phi)$, is computed as the mean squared error between the predicted value of the state $V_\phi(s_t)$ and the actual returns R_t. This loss function helps in assessing how well the value function predicts future rewards. The value function parameters ϕ are then updated to minimize this loss, improving the accuracy of the value function's predictions and thereby enhancing the overall learning process.

6. **Repeat**

   The PPO algorithm iterates through these steps repeatedly, gradually refining both the policy and the value function. Each iteration involves collecting new trajectories, computing advantages, updating the policy and value function, and then starting the process anew. This iterative approach allows the agent to continuously improve its performance and adapt to the environment, ultimately learning to make better decisions and achieve higher rewards.

By following these steps, PPO effectively balances exploration and exploitation, ensures stable learning, and drives the agent towards optimal performance in complex environments.

3.7 Working Principle

This project uses the Proximal Policy Optimization (PPO) algorithm to develop a trading strategy for hydropower stocks listed on NEPSE. Starting with raw data, the process involves data cleaning, feature engineering, and normalization to prepare it for the machine learning model. The PPO algorithm then learns to make buy, sell, or hold decisions by interacting with a simulated trading environment, where it receives rewards for profitable trades and penalties for losses. Finally, the model's output is analyzed and visualized to assess the effectiveness of the trading strategy.

3.7.1 Data Preprocessing

Pre-processing step in PPO based Stock trading Agent involves preparing and cleaning data before it's used for modeling. The main goal is to enhance the quality and relevance of the data.

1. **Data Cleaning**

   Data cleaning is a critical step in the data preprocessing process. It involves detecting and correcting or removing inaccuracies, inconsistencies, and incomplete data to improve the quality of the dataset before analysis or model training. The data that is scraped from sharesansar.com looks like the sample in Table 3.1 with 8 fields, Date, Open, High, Low, Ltp, % Change, Qty and Turnover.

   The process begins with handling missing data, where rows or columns with significant amounts of missing information is removed. Similarly, handling duplicates is done to prevent bias, and this involves identifying and removing duplicate rows from the dataset if present and confirming that there are no duplicates.

   Correcting data types is also important, as it ensures that each column has the appropriate data type, such as converting strings to dates or integers to floats, and includes parsing date strings into a proper datetime format to facilitate time-based analysis.

   These steps collectively help to prepare the data for accurate and effective analysis or model training.

2. **Feature Engineering:**

   Feature engineering involves creating new variables from existing data to help the model capture more complex patterns and relationships. This step enriches the data and improves the model's ability to make accurate predictions.

From the 8 Fields of data, we create multiple technical indicators as features for the model such as:

- **Simple Moving Averages (MA):** Average prices over specific periods to identify trends by constantly averaging the price. Fields 9, 10, 11 and 12 are MA5, MA20, MA50 and MA200 which is calculated as in Equation 3.8 The price value of LTP field is used to calculate these fields.
- **Exponential Moving Average (EMA):** A type of moving average that places greater weight on more recent prices, making it more responsive to recent price changes compared to the SMA. Field 13, 14, 15 and 16 are EMA5, EMA20, EMA50 and EMA200 which is calculated as in Equation 3.9.The price value of LTP field is used to calculate these fields.
- **Volatility:** Volatility refers to the degree of variation(i.e standard derivation) of a stock's price over time. Field 17, 18 and 21 are Volatality5, Volatality10, Volatility 20, which is the 5, 10 and 20 day volatility of the price.The value of Qty field is used to calculate these fields.
- **Stochastic Oscillator (SO):** Compares a particular closing price to a range of its prices over a specific period, helping to identify overbought and oversold conditions. Field 19 is Stochastic Oscillator Value calculated as in Equation 3.15.
- **Volume Moving Average:** Average of Volume of Stock Traded. Field 20 is Volume MA 20, which is the simple moving average of Volume of Stock Traded calculated as in Equation 3.8 but for Volume.
- **On-Balance Volume (OBV):** Uses volume flow to predict changes in stock price, with the idea that volume precedes price movement and can thus indicate future price changes. Field 22 is OBV which is calculated as Equation 3.17. The value of Qty field is used to calculate these fields.
- **Relative Strength Index (RSI):** Measures the speed and change of price movements to assess overbought or oversold conditions, helping to identify potential reversal points. Field 23 is RSI which is calculated as Equation 3.18. To calculate RSI we use the Values of LTP column.

- **Moving Average Convergence Divergence (MACD):** Helps identify changes in the strength, direction, momentum, and duration of a trend by comparing the relationship between two moving averages of a security's price. We calculate Field 24 and 25 as MACD and Signal Line which is calculated as Equation 3.10 and Equation 3.11. Field 26 is Histogram which is difference of value of MACD and Signal Line.
- **Bollinger Bands (BB):** Consists of a middle band (SMA) and two outer bands that are standard deviations away from the middle band. They help assess volatility and identify potential overbought or oversold conditions and is calculated in Field 27, 28 and 29 as Bollinger Mid, Bollinger Upper and Bollinger Lower. It is calculated as in Equation 3.12, Equation 3.13 and Equation 3.14.
- **Average Directional Index (ADX):** Measures the strength of a trend, regardless of its direction, to assess whether a market is trending or in a range-bound condition. Here we calculate two fields TR and ADX, as Fields 30 and 31 which are calculated as in Equation 3.20 and Equation 3.26.
- **Past Value of LTP:** It is simply the data of past values of LTP. Here, we have Fields 32, 33, 34, 35, and 36 as lag1, lag2, lag3, lag4 and lag5. These are the past 5 days data of LTP. so, lag1 is the LTP of yesterday, lag2 is of day before yesterday and so on.
- **Prediction:** The prediction in Field 37 is generated by fitting a linear regression model to lagged features (Fields 32-36) using least squares. It calculates coefficients for these lags, and predictions are made by applying these coefficients, forecasting the current LTP values.

3. **Data Normalization:**

   Stock prices and technical indicators vary in scale. Normalizing these values ensures that each feature contributes equally to the model's performance. For example, if one feature is in the range of 0-1 and another is in the range of 1000-10000, normalizing brings them to a similar scale. This helps prevent features with larger scales from dominating the model's learning process.

   For our PPO agent we use Standard Scalar to scale all the features.

Standard Scaler:

In machine learning, features with different scales can impact the performance of models, especially when using algorithms that rely on distance measures or gradient-based optimization, such as the PPO algorithm. To address this we use standard scaler for data normalization.

The Standard Scaler transforms data such that each feature has a mean of 0 and a standard deviation of 1. This process involves two main steps:

(a) **Mean Subtraction:** The mean value of each feature is subtracted from the data points, centering the data around zero.

(b) **Scaling by Standard Deviation:** Each data point is then divided by the standard deviation of the feature, normalizing the spread of the data.

Mathematically, for a feature x, the transformed value x' is calculated as:

$$x' = \frac{x - \mu}{\sigma} \tag{3.27}$$

Where:

- μ is the mean of the feature x.
- σ is the standard deviation of the feature x.

This transformation ensures that the features are on a similar scale, making the training process more efficient and preventing features with larger numerical ranges from disproportionately influencing the model's learning. For our PPO agent, using the Standard Scaler allows all features—whether stock prices or technical indicators—to contribute equally to the model's performance.

4. **State Representation:**

In reinforcement learning, the state space defines the information available to the model at each point in time. A well-defined state representation provides a comprehensive view of the market environment for decision-making. Each state space includes normalized Historical prices, Turnover, Volume, High, Low, Open, Close(LTP) and technical indicators calculated in feature engineering step.

3.7.2 Working of PPO Model

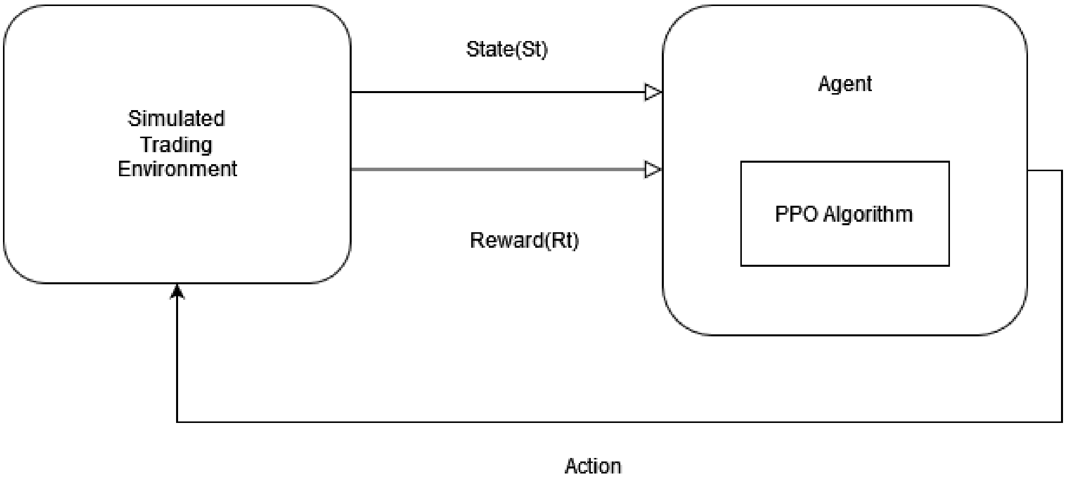

Figure 3.4: Working of The Model

The model works in a simulated trading environment, where the agent uses PPO algorithm to determine the best action in the current state. The trading environment will contain the following.

1. **State Space:** The state space would consist of various features and indicators relevant to the hydro power sector and stock market trading. These could include:
   - Historical stock prices of hydro power companies.
   - Volume of shares traded.
   - Technical indicators like moving averages, Exponential Moving Average(EMA), Relative Strength Index (RSI), Stochastic Oscillator(SO), Moving Average Convergence Divergence(MACD),Average Directional Moving Index(ADX), On Balance Volume(OBV) etc.
2. **Action Space:** The action space would represent the trading decisions the agent can take. Actions might include:
   - Buying shares of a specific hydro power company.
   - Selling shares of a specific hydro power company.
   - Holding onto current positions.

3. **Reward System:** The reward system would provide feedback to the agent based on the performance of its trading decisions. Rewards is calculated based on:

   - Profit and loss from executed trades.
   - Final Portfolio Value

4. **Agent:** The agent would observe the current state of the environment, which includes the historical data, indicators, and other relevant information mentioned in the state space and take actions. The environment allow the agent to interact with it and learn from the trading experience.

   We use the simulated trading environment and preprocessed data to train the PPO agent. The agent learns to optimize its trading strategy over time by maximizing cumulative rewards. After that, we evaluate the trained agent's performance on a validation set and then fine-tune the agent's parameters as needed to improve its performance.

In our PPO Algorithm we have two neural networks, the **policy network** and the **value network**.

Policy Network: The policy network's main role is to determine the action probabilities based on the current state. It produces a probability distribution over possible actions in discrete action spaces or directly outputs action values for continuous spaces. This network processes the observation to predict action probabilities and is optimized to maximize the expected cumulative reward by improving these predictions.

Structure:

- **Input Layer**: Linear(in_features=37, out_features=64, bias=True)

  Takes the observation vector with 37 features which are the technical indicator values and raw data like high, low, LTP ect and projects it into a 64-dimensional space.

- **Activation Function**: Tanh()

  Applies the hyperbolic tangent function, introducing non-linearity.

- **Hidden Layer**: Linear(in_features=64, out_features=64, bias=True)

  Projects the 64-dimensional space into another 64-dimensional space.

- **Activation Function**: Tanh()

  Again applies the `Tanh` function for non-linearity.

The policy network processes the state input and outputs probabilities for each action in the action space. Given the action space is 3, this network would eventually use a final layer to convert the 64-dimensional output into a 3-dimensional probability distribution over the actions. The Policy Network of the Stock trading agent is shown in figure 3.5.

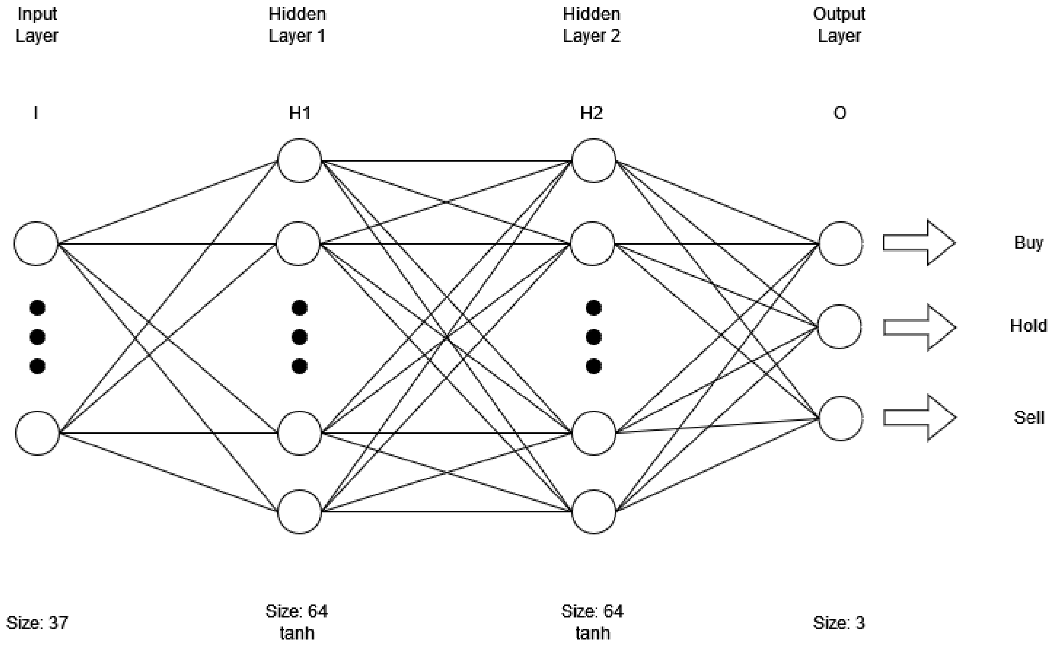

Figure 3.5: Policy Network of PPO based Stock Trading Agent

Value Network: On the other hand, the value network estimates the value of being in a given state. It provides a scalar value that indicates how beneficial it is to be in that state or to take that action in that state. The value network helps compute the advantage function, which is crucial for guiding updates to the policy network.

Structure:

- **Input Layer**: Linear(in_features=37, out_features=64, bias=True)

Similar to the policy network, this layer projects the 37-dimensional input into a 64-dimensional space.

- **Activation Function**: Tanh()

  Applies the `Tanh` function for non-linearity.

- **Hidden Layer**: Linear(in_features=64, out_features=64, bias=True)

  Projects the 64-dimensional space into another 64-dimensional space.

- **Activation Function**: Tanh()

  Applies the `Tanh` function again.

The value network processes the same state input to output a single scalar value representing the estimated value of the state. This value is used to compute the advantage function, which helps in updating the policy. The Value Network of the Stock trading agent is shown in figure 3.6.

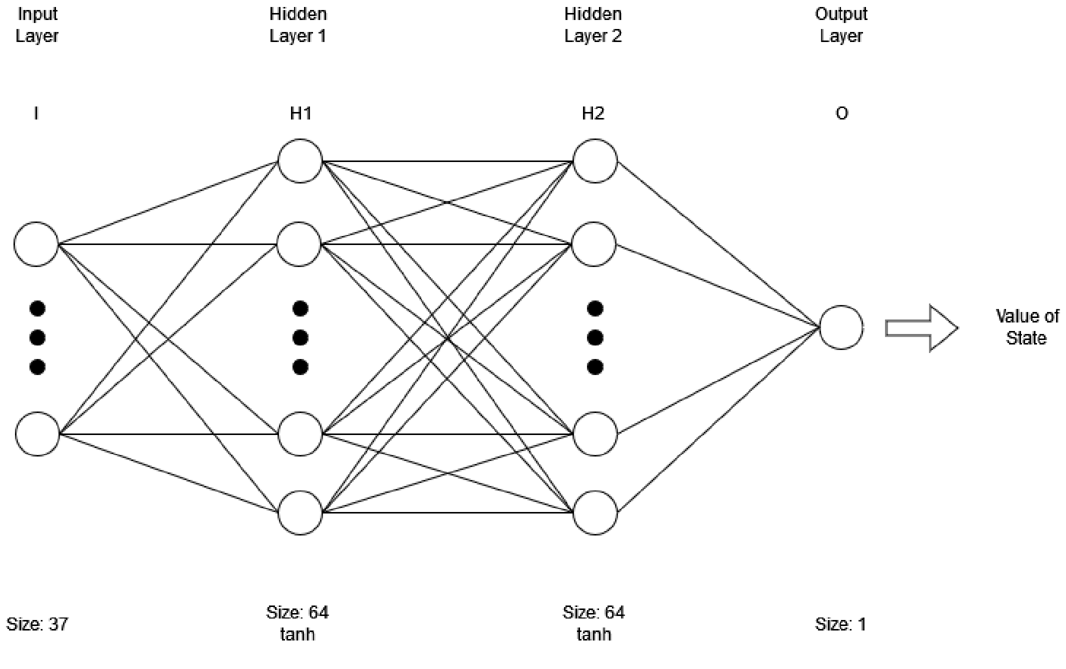

Figure 3.6: Value Network of PPO based Stock Trading Agent

While both networks generally share the initial layers for feature extraction from the state input, they diverge at this point. The policy network focuses on predicting action probabilities or actions directly, whereas the value network focuses on estimating the state value. This separation allows each network to specialize in its respective task, with the policy network handling action selection and the value network aiding in evaluating and improving the policy.

3.7.3 Data Post-Processing

Post-processing the output of the Proximal Policy Optimization (PPO) algorithm is a crucial step to ensure that the decisions made by the model are effectively implemented and evaluated.

Output Generation

The policy network of the PPO model outputs a probability distribution over a set of possible actions. For our model, these actions include:

- Buy
- Sell
- Hold

The action with the highest probability is selected as the decision. For example, if the output probabilities are $\{0.2$ for Buy, 0.5 for Sell, 0.3 for Hold$\}$, the selected action is 'Sell'.

Then the selected action is converted into a market order. If the action is 'Buy', an order to purchase a specific quantity of the stock is placed in the simulated trading environment. If the action is 'Sell', an order to sell a specific quantity is placed. The 'Hold' action implies no trade is made for that period.

Transaction costs, such as broker fees, is considered as these fees also impact the trades profitability.

Every action taken by the model is logged for future reference. This includes the action type, the quantity traded, the price at which the trade was executed, and the timestamp.

Maintaining a detailed log helps in performance analysis and debugging.

Performance Analysis

Once that is done we evaluate the model regarding its profitability, risk assessment and comparing it with other strategy.

1. **Profitability**

   In this method we measure the financial success of the model when applied to real trading scenarios. Following Metrics are used to calculate the Profitability of the agent.

   - **Annualized Return**: This metric calculates the geometric average annual return over the backtesting period, providing a consistent basis for comparison with other investments or benchmarks.
   - **Sharpe Ratio**: The Sharpe ratio is calculated to evaluate the risk-adjusted return of the trading strategy. A higher Sharpe ratio indicates that the model is generating higher returns per unit of risk, which is crucial for evaluating the profitability of a trading strategy.
   - **Cumulative Return**: The total return generated by the model over the entire backtesting period, showing the overall profitability of the strategy.

2. **Risk Assessment**

   It is used to understand the level of risk the PPO model takes on in making trading decisions. Following Metric can be used for risk assessment.

   - **Volatility**: The standard deviation of returns is measured to assess the overall risk and variability in the model's performance. High volatility indicates higher risk.
   - **Maximum Drawdown**: This metric measures the largest peak-to-trough decline in the portfolio's value during the backtesting period. It helps assess the potential for significant losses, which is particularly important in trading volatile assets like hydropower stocks.

3. **Comparative Analysis**

   In this evaluation method we compare the performance of the PPO-based model with other trading strategies like buy and hold, SMA crossover, EMA crossover, RSI, SO, Bollinger Band Strategy and MACD Strategy.

Trading Signal Visualization

This step demonstrates when the Proximal Policy Optimization (PPO) algorithm makes decisions to buy, sell, or hold stocks. This visualization is crucial for understanding the decision-making process of the algorithm and for evaluating its effectiveness in identifying profitable trading opportunities.

- **Entry Points**:

  These are the moments when the algorithm decides to buy a stock. Entry points typically occur when the algorithm detects a potential upward trend, indicating that the stock price is likely to increase.

  Ideal Scenario: An entry point should be identified just before a significant upward movement in the stock price. This allows the trader to maximize profits by buying low and selling high.

- **Exit Points**:

  Exit points are the moments when the algorithm decides to sell a stock. These points are ideally chosen near the peak of an upward trend, just before the stock price starts to decline.

  Ideal Scenario: An exit point should be chosen at or near the highest price in the trend, allowing the trader to sell high and secure the maximum possible profit before the price drops.

Visualization:

- **Markers for Trading Signals**:

- **Buy Signals**: Green markers, often in the form of upward-pointing arrows, are placed on the chart to indicate the exact points where the algorithm decided to buy the stock.
- **Sell Signals**: Red markers, often in the form of downward-pointing arrows, are used to show where the algorithm decided to sell the stock.

By analyzing these markers over time, we assess whether the algorithm consistently buys before significant price increases and sells before declines, thus evaluating its effectiveness in real-world trading scenarios.

3.8 Verification and Validation Procedures

The verification and validation procedures play a critical role in assessing the performance and reliability of the predictive model developed for stock price prediction. In this section, we discuss the dataset splitting strategy and the chosen metric for model verification, along with the relevance of the selected metric in evaluating the model's output.

3.8.1 Dataset Splitting

The dataset acquired from Sharesansar is split into two subsets: training set, and test set. The training set, contains historical data for the Hydropower stock up to May 8, 2024, which is used to train the model. The remaining portion of the data, from May 9, 2024, to July 29, 2024 forms the test set, which is reserved for evaluating the final model's performance on unseen data.

3.8.2 Chosen Metric for Verification

These six metrics can be used for the verification of our model.

1. **Sharpe Ratio**: This is a widely used metric in finance to measure the risk-adjusted return of an investment strategy. It considers both the strategy's return and its volatility, providing insight into how well the strategy performs relative to the risk taken.

   It is calculated as the difference between the average return of the strategy (R_p) and the risk-free rate of return (R_f), divided by the standard deviation of the strategy's returns (σ_p). The formula for the Sharpe Ratio is:

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p} \tag{3.28}$$

2. **Annualized Return**: This metric measures the average annual return generated by the trading strategy. It provides a straightforward measure of performance. It is calculated using the formula:

$$\text{Annualized Return} = \left(\frac{P_t}{P_0}\right)^{\frac{1}{n}} - 1 \tag{3.29}$$

where P_t is the final value of the portfolio or investment at time t, P_0 is the initial value of the portfolio or investment, and n is the number of years. These metrics are essential for evaluating the performance of trading strategies and making informed investment decisions.

3. **Cumulative Return**: This measures the total return of the trading strategy over a given period. It is calculated as:

$$\text{Cumulative Return} = \frac{P_t - P_0}{P_0} \tag{3.30}$$

where P_t is the final value and P_0 is the initial value of the portfolio or investment.

4. **Maximum Drawdown**: This metric quantifies the largest peak-to-trough decline in the portfolio value, representing the worst loss experienced during a specific period. It is calculated as:

$$\text{Maximum Drawdown} = \frac{P_{peak} - P_{trough}}{P_{peak}} \tag{3.31}$$

where P_{peak} is the highest value and P_{trough} is the lowest value during the drawdown period.

5. **Volatility**: This measures the variability of returns of the trading strategy. It is often used as an indicator of risk and is calculated as the standard deviation of the returns. The formula is:

$$\text{Volatility} = \sigma_p \tag{3.32}$$

where σ_p is the standard deviation of the strategy's returns.

3.8.3 Relevance of Chosen Metric

The Sharpe Ratio evaluates the risk-adjusted return of investments, offering insights into how well the trading agent performs relative to the risk taken. By comparing the agent's returns to its volatility, aids in objective comparisons of different strategies and helps in optimizing portfolio allocation. Annualized Return provides a clear measure of the agent's historical performance on an annual basis, making it easier to benchmark against other strategies and assess long-term profitability.

Cumulative Return captures the total return over a given period, reflecting the overall effectiveness of the trading strategy in adding value to the portfolio. Maximum Drawdown measures the largest peak-to-trough decline, helping to assess the worst-case scenarios and the agent's risk management capabilities. Lastly, Volatility indicates the variability in returns, providing insight into the stability of the agent's performance.

Together, these metrics offer a comprehensive assessment of both performance and risk, guiding effective portfolio management and strategy optimization.

4 RESULTS

The results section presents the output of the model, showcasing the generated buy and sell signals as part of the trading strategy along with the outcomes of the data preprocessing and feature engineering stages.

4.1 Data Preprocessing and Feature Engineering Output Visualization

For all the hydropower sector stocks in NEPSE, we analyzed and calculated the values of various technical indicators. The following example showcases the calculated technical indicators for Asian Hydropower Limited (AHL). These indicators were similarly calculated for all hydropower sector stocks listed on NEPSE to provide a comprehensive analysis of the sector.

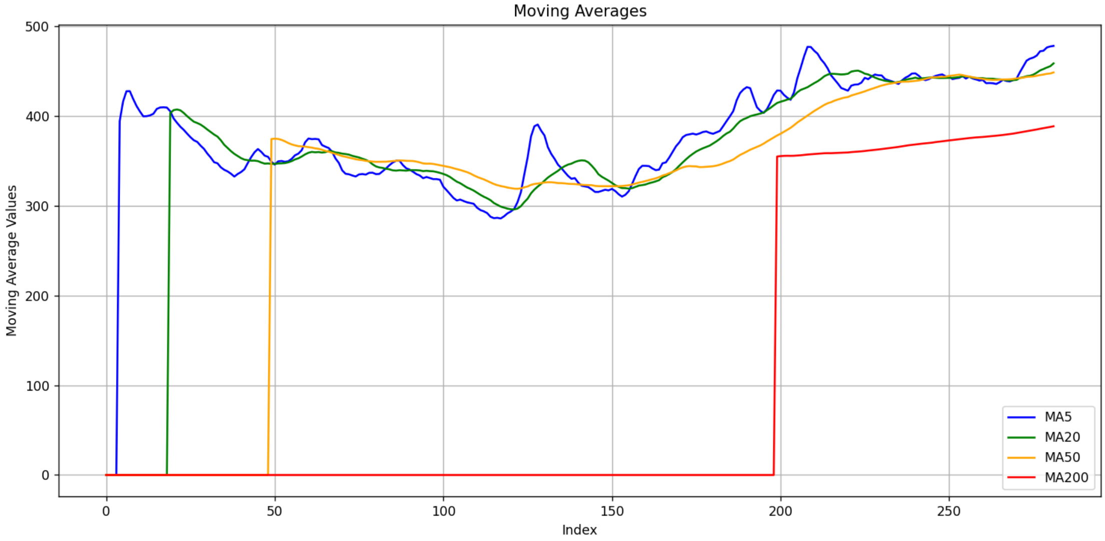

Figure 4.1: SMA Indicator for AHL

The graph in Figure 4.1 shows plot of four different moving averages (MA5, MA20, MA50, and MA200) over a period of Asian Hydropower Limited (AHL), where each moving average represents the average price of the asset over the respective number of periods (5, 20, 50, and 200).

These moving averages are used to smooth out price data and highlight trends over different time frames. The shorter moving averages (MA5 and MA20) react more quickly to recent price changes, showing more volatility, while the longer moving averages (MA50 and MA200) provide a broader view of the overall trend by reacting more slowly to price changes.

This visualization helps in identifying trend directions, crossovers, and divergences. For instance, when a shorter moving average crosses above a longer one, it may signal a bullish trend, and when it crosses below, it may signal a bearish trend. The plotted moving averages allow for a clear comparison of short-term and long-term trends within the same chart.

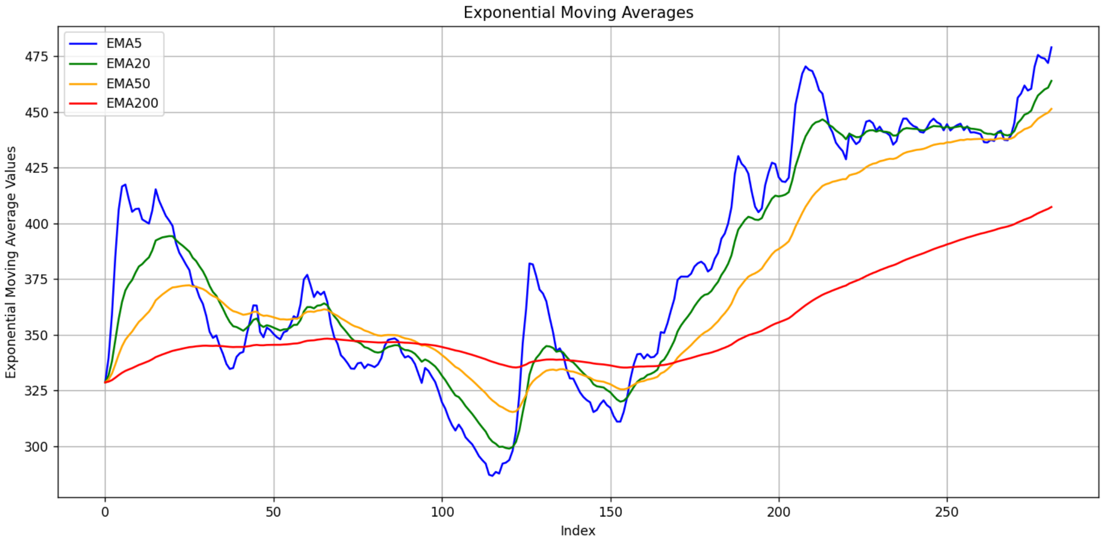

Figure 4.2: EMA Indicator for AHL

The graph in Figure. 4.2 shows four different exponential moving averages (EMA5, EMA20, EMA50, and EMA200) over a period of Asian Hydropower Limited (AHL), with more weight given to recent prices.

These exponential moving averages are used to smooth out price data and highlight trends over different time frames, similar to simple moving averages but with greater responsiveness to recent price changes. The shorter exponential moving averages (EMA5 and EMA20) react more quickly to recent price changes, showing more volatility, while the longer exponential moving averages (EMA50 and EMA200) provide a broader view of the overall trend by reacting more slowly to price changes.

This visualization helps in identifying trend directions, crossovers, and divergences. For instance, when a shorter exponential moving average crosses above a longer one, it may signal a bullish trend, and when it crosses below, it may signal a bearish trend. The plotted exponential moving averages allow for a clear comparison of short-term and long-term trends within the same chart, providing insight into the strength and direction of the market's momentum.

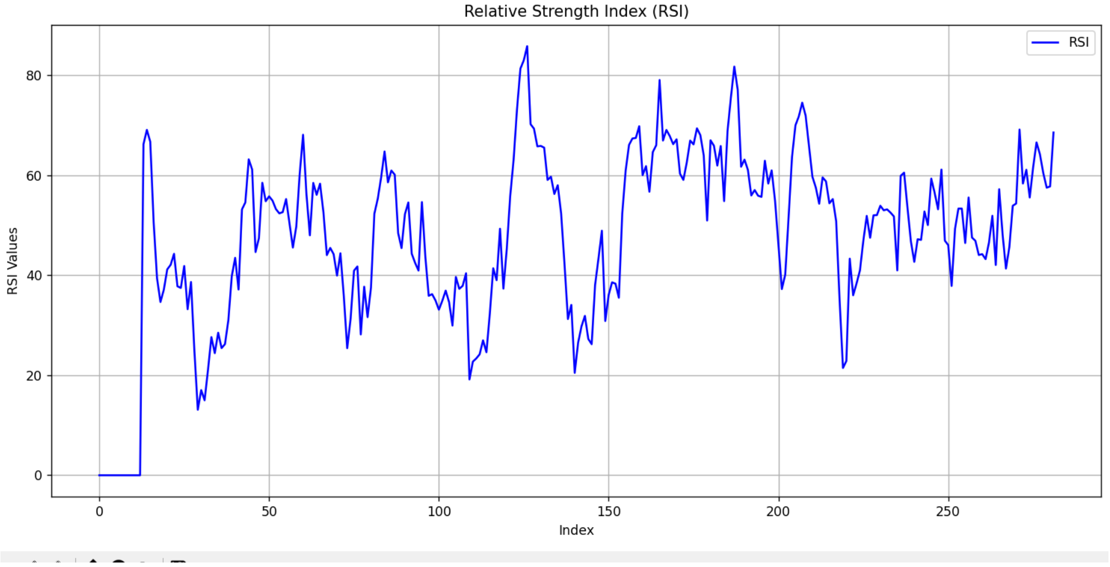

Figure 4.3: RSI Indicator for AHL

The graph in Figure 4.3 plots the Relative Strength Index (RSI) of Asian Hydropower Limited (AHL) over a specified period, where the RSI is a momentum oscillator used to measure the speed and change of price movements.

The RSI is represented by a blue line, and its values typically range from 0 to 100. This indicator helps identify overbought or oversold conditions in a market; traditionally, an RSI above 70 indicates overbought conditions, while an RSI below 30 indicates oversold conditions.

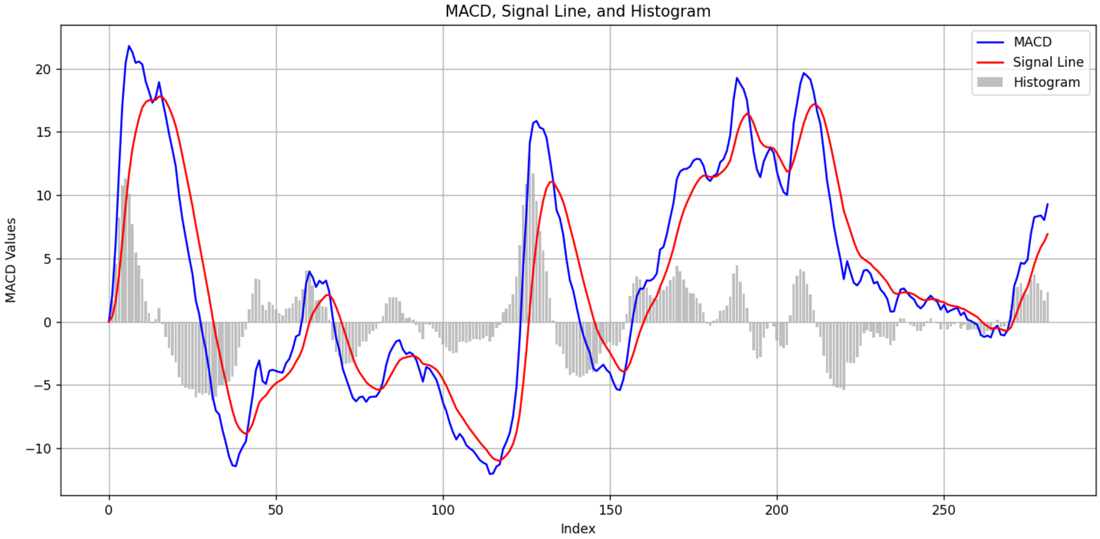

Figure 4.4: MACD Indicator for AHL

The graph in Figure 4.4 plots the MACD (Moving Average Convergence Divergence) line, Signal line, and the MACD histogram of Asian Hydropower Limited (AHL), which are key indicators used in technical analysis to identify changes in the strength, direction, momentum, and duration of a trend.

The MACD line, represented in blue, is calculated by subtracting the longer-term exponential moving average (EMA) from the shorter-term EMA. The Signal line, in red, is typically a 9-day EMA of the MACD line. The histogram, displayed as grey bars, represents the difference between the MACD line and the Signal line, visually indicating the convergence or divergence between the two.

When the MACD line crosses above the Signal line, it may indicate a bullish signal, suggesting a potential buying opportunity. Conversely, when the MACD line crosses below the Signal line, it may indicate a bearish signal, suggesting a potential selling opportunity.

The height of the histogram bars further aids in visualizing the momentum; larger bars indicate stronger momentum, while smaller bars indicate weaker momentum.

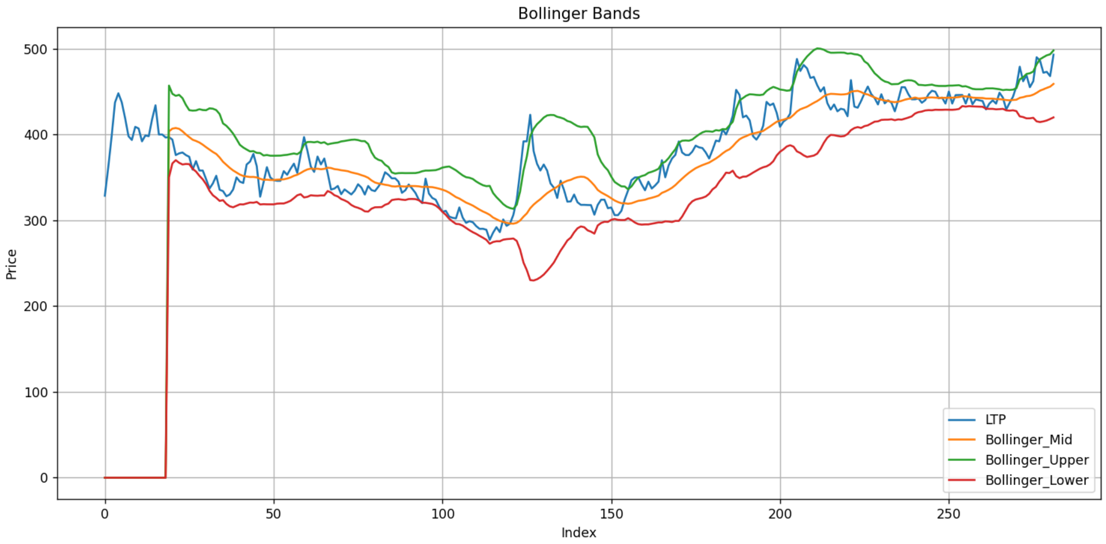

Figure 4.5: Bollinger Band Indicator for AHL

The graph in Figure 4.5 displays the Last Traded Price (LTP) along with the Bollinger Bands, which are used in technical analysis to assess price volatility and potential price reversals. The LTP, represented by a single line, shows the actual market price of the asset over time.

The Bollinger Bands consist of three lines: the Bollinger Mid line, which represents the moving average of the asset's price, and the Bollinger Upper and Lower lines, which are typically set two standard deviations away from the mid line, indicating the upper and lower bounds of price movement.

When the LTP moves towards the upper band, it may suggest that the asset is overbought, potentially indicating a price decrease or consolidation. Conversely, when the LTP approaches the lower band, it may indicate that the asset is oversold, potentially signaling a price increase or bounce back.

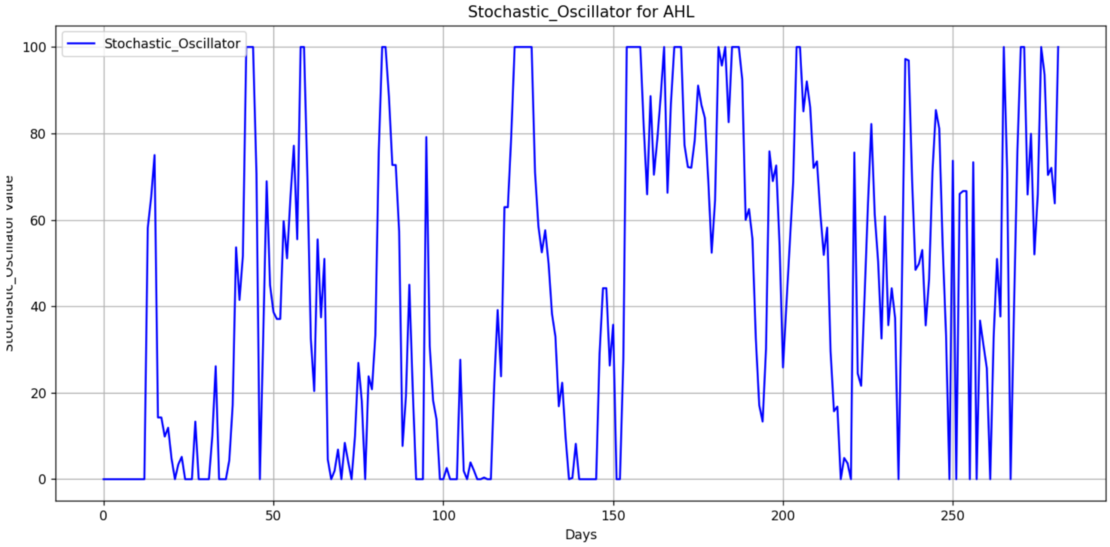

Figure 4.6: SO Indicator for AHL

The graph in Figure 4.6 displays the daily Stochastic Oscillator values for Asian Hydropower Limited (AHL). The Stochastic Oscillator is a momentum indicator used in technical analysis to determine overbought or oversold conditions of a security. It compares the most recent closing price to a range of prices over 14 days.

The Stochastic Oscillator values range from 0 to 100, with readings above 80 indicating overbought conditions and readings below 20 indicating oversold conditions.

4.2 Stockwise Return Using PPO Method for Case I

For Case I evaluation period spanned from May 9, 2024, to July 29, 2024. Table 4.1 presents a comprehensive overview of the performance metrics for each stock during the testing phase of our trading model. The testing was conducted with an initial balance of Rs 100,000 for each stock, allowing the agent to execute trades within a custom trading environment.

The table provides key financial metrics for each stock, including the Final Portfolio Value, Cumulative Returns, Annual Return, Annual Volatility, and the Sharpe Ratio. These metrics are essential for assessing the profitability and risk-adjusted returns of the portfolio.

- **Final Portfolio Value** represents the total value of the portfolio at the end of the testing period, including the value of held stocks and any remaining cash balance.

- **Cumulative Returns** (%) indicate the percentage change in the portfolio value relative to the initial investment. This metric reflects the overall performance of the portfolio over the testing period.

- **Annual Return** (%) is an annualized version of the cumulative return, providing a standardized measure of the portfolio's return that facilitates comparison with other investments or benchmarks.

- **Annual Volatility** (%) measures the standard deviation of the portfolio's returns, indicating the degree of variation or risk associated with the portfolio's returns. Higher volatility signifies greater risk.

- **Sharpe Ratio** is a risk-adjusted measure of return, calculated by dividing the portfolio's excess return (over a risk-free rate) by its volatility. A higher Sharpe Ratio indicates better risk-adjusted performance, making it a crucial metric for investors seeking to balance risk and return.

- **Maximum Drawdown** is the largest percentage decline from a portfolio's peak value to its lowest point before a new peak is reached. It is a measure of downside risk, highlighting the potential extent of losses during a market downturn.

This detailed analysis helps in understanding how the model performed across different stocks, providing valuable insights into its strengths and potential areas for improvement. By evaluating these metrics, investors can make informed decisions about portfolio allocation and risk management strategies.

Table 4.1: Return Using PPO Method For Hydropower Stocks For Case I

Company Code	**Final Portfolio Value**	**Cumulative Returns (%)**	**Annual Return (%)**	**Annual Volatility (%)**	**Sharpe Ratio**	**Maximum Drawdown (%)**
AHL	150078.37	50.08	1537.82	41.30	4.89	-4.98
AHPC	114216.45	14.22	149.78	25.58	2.59	-6.40

Continued on next page

Company Code	Final Portfolio Value	Cumulative Returns (%)	Annual Return (%)	Annual Volatility (%)	Sharpe Ratio	Maximum Drawdown (%)
AKJCL	119890.31	19.89	248.79	27.07	3.32	-3.71
AKPL	110466.42	10.47	98.48	21.50	2.31	-6.36
API	111081.68	11.08	106.22	21.23	2.46	-5.74
BARUN	119070.44	19.07	232.69	26.02	3.32	-2.49
BEDC	110318.57	10.32	96.66	23.31	2.12	-6.53
BGWT	135194.84	35.19	697.80	41.64	3.65	-8.16
BHDC	109349.50	9.35	85.06	18.16	2.43	-5.02
BHL	110735.94	10.74	101.84	30.52	1.74	-7.04
BHPL	128471.72	28.47	461.48	40.30	3.16	-12.74
BNHC	118717.12	18.72	225.96	26.30	3.23	-0.37
BPCL	110098.53	10.10	93.97	13.01	3.58	-2.96
CHCL	113800.74	13.80	143.59	20.10	3.16	-5.04
CHL	110139.73	10.14	94.47	18.41	2.58	-1.99
CKHL	109724.89	9.72	89.48	19.47	2.36	-3.59
DHPL	115033.24	15.03	162.35	21.51	3.20	-2.38
DOLTI	120466.45	20.47	260.50	25.42	3.61	-4.78
DORDI	112017.02	12.02	118.48	24.25	2.34	-5.23
EHPL	115128.53	15.13	163.85	24.26	2.88	-0.37
GHL	115574.48	15.57	170.97	23.61	3.03	-5.40
GLH	115492.80	15.49	169.65	35.94	2.08	-11.66
GVL	127961.90	27.96	446.31	33.41	3.67	-4.70
HDHPC	112418.50	12.42	123.93	26.33	2.24	-5.81
HHL	111758.02	11.76	115.02	25.73	2.18	-8.81
HPPL	113147.67	13.15	134.12	19.00	3.19	-3.42
HURJA	125167.78	25.17	369.26	34.60	3.25	-5.39
IHL	123230.82	23.23	321.47	34.19	3.07	-7.11

Continued on next page

Company Code	**Final Portfolio Value**	**Cumulative Returns (%)**	**Annual Return (%)**	**Annual Volatility (%)**	**Sharpe Ratio**	**Maximum Drawdown (%)**
JOSHI	120490.70	20.49	261.00	23.44	3.90	-2.10
KBSH	127253.29	27.25	425.82	39.51	3.09	-5.83
KKHC	120210.51	20.21	255.26	21.29	4.22	-3.48
KPCL	107650.06	7.65	66.14	24.14	1.57	-5.70
LEC	118993.31	18.99	231.21	19.07	4.43	-1.20
MAKAR	113927.72	13.93	145.47	28.33	2.32	-5.52
MANDU	123737.29	23.74	333.54	29.82	3.54	-2.92
MBJC	105277.56	5.28	42.50	14.11	1.80	-3.28
MCHL	118865.90	18.87	228.78	30.72	2.83	-5.85
MEHL	109795.23	9.80	90.32	23.45	2.01	-3.71
MEL	112926.61	12.93	130.99	21.99	2.74	-5.92
MEN	100000.00	0.00	0.00	0.00	0.00	0.00
MHCL	112273.83	12.27	121.95	32.00	1.87	-8.50
MHL	119373.78	19.37	238.58	30.43	2.92	-5.05
MHNL	108889.85	8.89	79.77	37.48	1.26	-11.09
MKHC	117179.12	17.18	197.96	25.82	3.05	-5.20
MKHL	132362.62	32.36	589.56	29.22	4.71	-4.19
MKJC	118797.90	18.80	227.49	33.75	2.59	-5.54
MMKJL	127150.43	27.15	422.90	34.21	3.51	-9.69
MSHL	108610.67	8.61	76.62	25.34	1.67	-4.50
NGPL	109853.38	9.85	91.02	17.74	2.61	-4.11
NHDL	106006.11	6.01	49.43	21.95	1.37	-5.09
NHPC	112116.58	12.12	119.82	24.60	2.33	-7.38
NYADI	118134.91	18.13	215.10	25.68	3.21	-4.46
PHCL	107710.69	7.71	66.79	16.69	2.20	-3.19
PMHPL	111704.69	11.70	114.32	26.20	2.14	-6.22

Continued on next page

Company Code	**Final Portfolio Value**	**Cumulative Returns (%)**	**Annual Return (%)**	**Annual Volatility (%)**	**Sharpe Ratio**	**Maximum Drawdown (%)**
PPCL	110585.61	10.59	99.96	21.45	2.34	-6.72
PPL	127052.52	27.05	420.13	31.99	3.72	-4.40
RADHI	113866.47	13.87	144.56	18.02	3.52	-1.29
RAWA	115813.09	15.81	174.84	24.09	3.02	-4.45
RFPL	120489.75	20.49	260.98	29.98	3.10	-7.58
RHGCL	115107.81	15.11	163.52	15.43	4.42	-2.06
RHPL	108455.73	8.46	74.89	12.75	3.09	-2.40
RIDI	113953.85	13.95	145.86	22.52	2.87	-2.38
RURU	100000.00	0.00	0.00	0.00	0.00	0.00
SAHAS	102731.29	2.73	20.39	20.85	0.72	-9.87
SGHC	107420.39	7.42	63.71	24.64	1.50	-6.89
SHEL	120374.23	20.37	258.61	28.59	3.22	-6.98
SHPC	110145.58	10.15	94.54	13.72	3.42	-1.80
SIKLES	135402.30	35.40	706.27	29.61	5.02	-8.07
SJCL	114262.87	14.26	150.48	19.11	3.41	-5.44
SMH	118962.35	18.96	230.62	25.56	3.36	-3.76
SMHL	121382.60	21.38	279.81	31.43	3.09	-7.47
SMJC	107155.93	7.16	60.96	11.44	2.93	-1.16
SPC	114176.95	14.18	149.19	36.89	1.89	-11.58
SPDL	115781.61	15.78	174.33	28.86	2.56	-9.60
SPHL	109168.38	9.17	82.96	22.94	1.93	-4.96
SPL	113757.25	13.76	142.95	24.33	2.64	-2.81
SSHL	113788.19	13.79	143.41	28.76	2.28	-6.01
TAMOR	109291.91	9.29	84.39	16.66	2.62	-4.88
TPC	105797.61	5.80	47.42	21.87	1.33	-5.24
TSHL	112868.05	12.87	130.17	28.04	2.19	-7.72

Continued on next page

Company Code	**Final Portfolio Value**	**Cumulative Returns (%)**	**Annual Return (%)**	**Annual Volatility (%)**	**Sharpe Ratio**	**Maximum Drawdown (%)**
TVCL	106920.95	6.92	58.54	30.14	1.20	-7.86
UHEWA	120435.22	20.44	259.86	24.95	3.67	-0.66
ULHC	124146.71	24.15	343.52	34.66	3.13	-4.11
UMHL	113578.20	13.58	140.33	27.30	2.35	-8.22
UMRH	112666.84	12.67	127.36	19.96	2.94	-2.22
UNHPL	121740.54	21.74	287.59	33.18	2.98	-9.12
UPCL	119075.68	19.08	232.80	26.84	3.23	-6.49
UPPER	111709.90	11.71	114.39	17.65	3.07	-2.84
USHEC	120108.32	20.11	253.19	33.33	2.78	-7.45
USHL	108783.87	8.78	78.57	11.87	3.43	-0.77
VLUCL	108006.24	8.01	69.96	14.27	2.64	-1.88
Total	**10503009.97**	**15.76**	**203.52**	**25.43**	**2.82**	**-5.22**

Stocks like **AHL** achieved exceptional returns during the testing period. AHL's impressive 1537.82% annual return and a high Sharpe Ratio of 4.89 suggest outstanding performance with highly favorable risk-adjusted returns.

Similarly, **SIKLES** also demonstrated exceptional performance, with a cumulative return of 706.27% and a Sharpe Ratio of 5.02. Such high returns, combined with a strong Sharpe Ratio, indicate that SIKLES not only achieved significant profitability but did so with effective risk management.

In contrast, stocks like **MEN** and **RURU** underperformed significantly, no return as the agent did not perform any trade for these stocks. The issue might be related to stock-specific conditions, such as the stocks not meeting the defined trading criteria or having low volatility and liquidity.

The final portfolio value across all companies is approximately 10,503,010. This represents an average cumulative return of 15.76% over the evaluated period. The annualized return for the portfolio stands at 203.52%, reflecting substantial growth in portfolio value year-over-year.

The Sharpe ratio, a measure of risk-adjusted return, is 2.82, suggesting that the returns are relatively high compared to the risk taken. Lastly, the maximum drawdown, which represents the largest peak-to-trough decline in average, is -5.22%, indicating that the portfolio experienced a peak-to-trough decline of 5.22% at its worst.

4.3 Output For Various Scenario Case I

4.3.1 Best Case Scenario

In the best-case scenario, the PPO-based stock trading agent demonstrated remarkable performance with stocks like Asian Hydropower Limited(AHL).

Over a period of 53 trading days, the agent achieved a notable increase in portfolio value from Rs100,000 to Rs150,078.37. This represents a substantial total return of 50.08%, reflecting the agent's ability to capitalize on favorable market conditions. When annualized, this return translates to an impressive rate of approximately 722%, highlighting the PPO agent's potential for exceptional gains in a bull market.

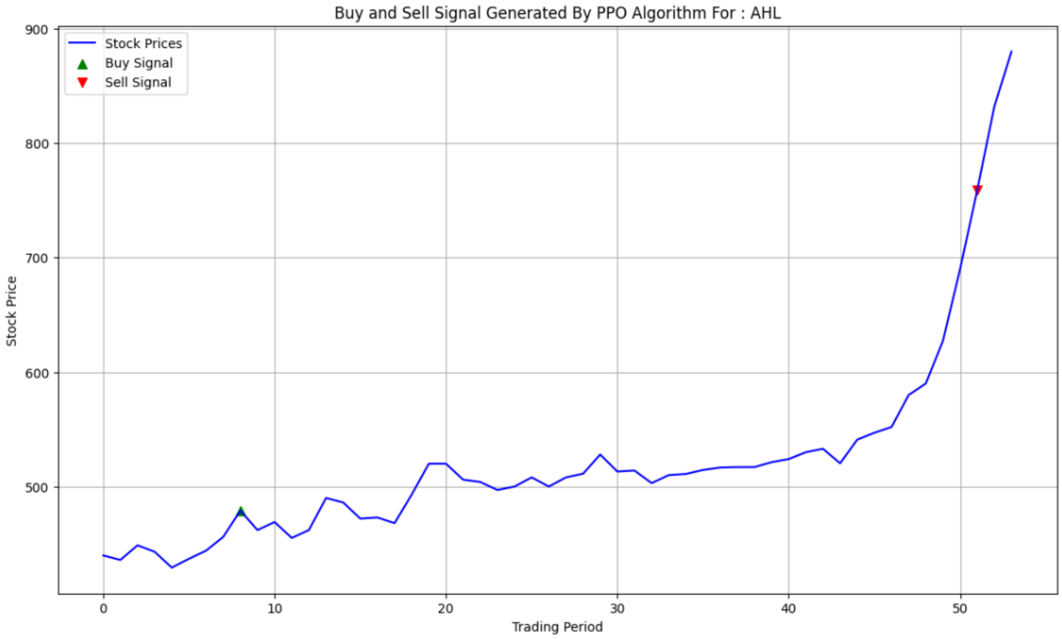

Figure 4.7: Best Case Scenario AHL Buy and Sell Signal

The agent's strategy involved purchasing 218 shares of AHL at Rs479 per share and subsequently selling them at Rs758.8 per share. This trade alone generated significant profits, underscoring the effectiveness of the PPO agent's decision-making capabilities. The substantial increase in portfolio value and the successful trade execution are clear indicators of the PPO agent's adeptness at identifying and exploiting profitable trading opportunities.

Additionally, the Sharpe Ratio during this period was notably high, reflecting the agent's ability to achieve high returns with relatively controlled risk. The enhanced performance metrics, including the Sharpe Ratio, cumulative returns, and the notable profit from the trade, collectively demonstrate the PPO agent's robust trading strategy and effectiveness in leveraging market trends to generate substantial gains. This case highlights the potential of the PPO-based approach in delivering impressive returns and managing risk effectively in favorable market conditions.

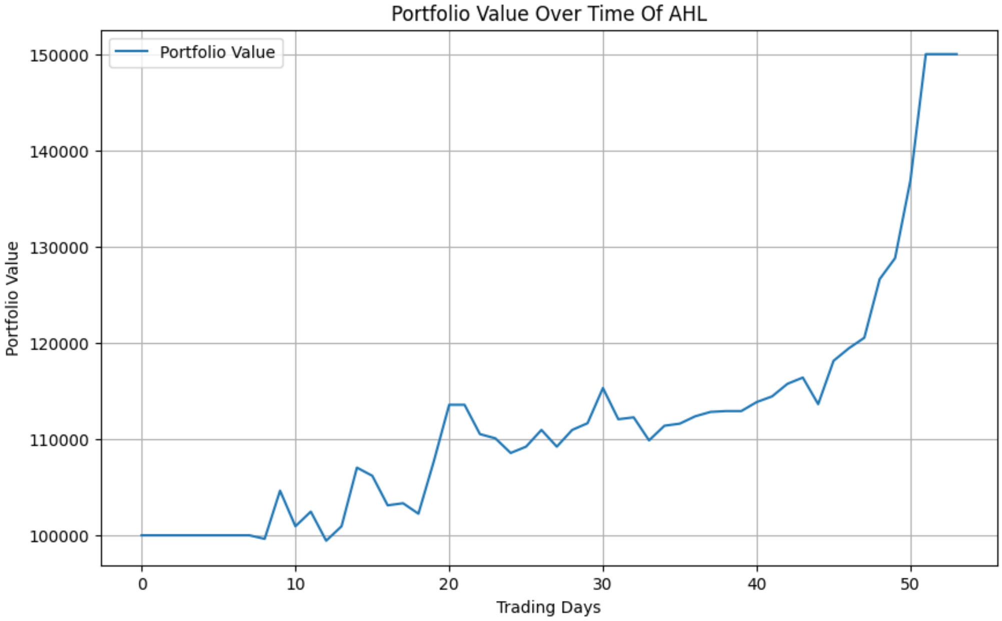

Figure 4.8: Portfolio Value over Time AHL

4.3.2 Worst Case Scenario

The PPO-based stock trading agent significantly underperformed with stocks like Mountain Energy Nepal Limited (MEN) and Ru Ru Jalbidhyut Pariyojana Limited (RURU). Despite being in a bull market, the agent recorded no returns for these stocks, as it did not execute any trades. This lack of trading highlights a critical shortcoming of the strategy in specific market conditions.

The absence of trades could be attributed to these stocks not aligning with the agent's defined trading criteria. Issues such as low volatility and liquidity may have made MEN and RURU less attractive for trading, causing the agent to bypass these opportunities even when the overall market conditions were favorable.

As a result, the agent missed potential gains in a rising market due to its rigid adherence to its trading criteria. This scenario underscores the need for the agent to adapt more flexibly to various stock characteristics and market conditions to capitalize on opportunities more effectively.

Similarly, the stock Madhya Bhotekoshi Jalavidyut Company Limited (MBJC) also underperformed during the bull market, achieving a modest cumulative return of only 5.28%. Although the market overall was rising, the trading agent struggled to generate substantial profits from MBJC. The Sharpe Ratio of 1.8 suggests that while the returns were positive, they were not particularly high relative to the risk taken.

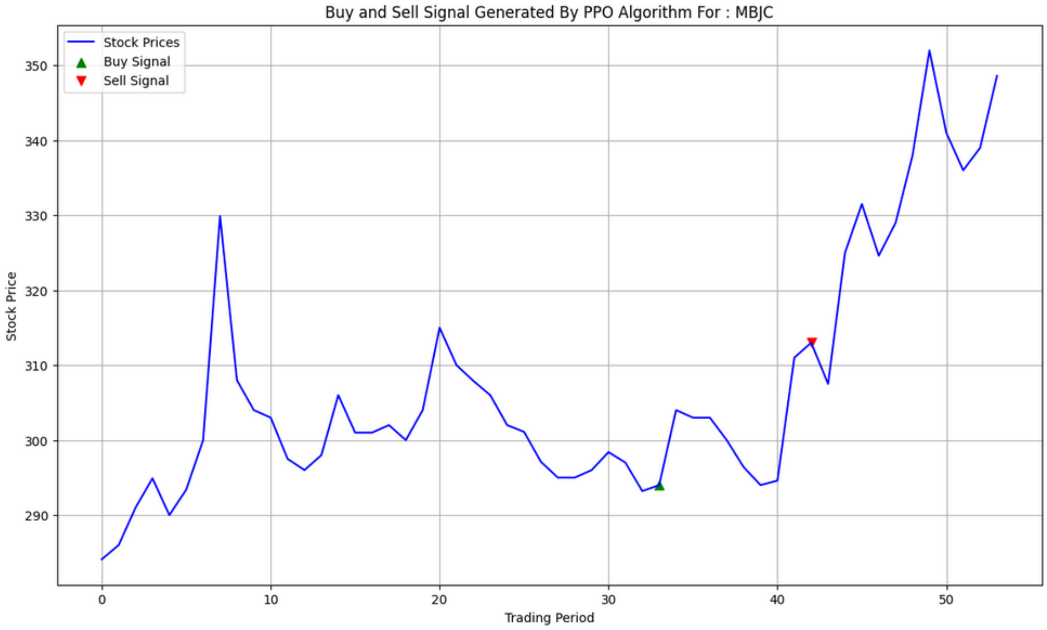

Figure 4.9: Worst Case Scenario MBJC Buy Sell Signal

The agent's performance with MBJC over a span of 53 trading days saw an increase in portfolio value from Rs100,000 to Rs105,277.55, resulting in a total return of 5.28%. The agent executed a trade by purchasing 339 shares of MBJC at Rs294 each and selling them at Rs313 each, achieving a modest gain. Despite these trades, the returns were relatively small.

This underperformance highlights a critical issue: the agent was unable to leverage the bullish market conditions effectively. The modest gains, coupled with the high volatility and risk indicated by the Sharpe Ratio, demonstrate that the strategy struggled to capitalize on the favorable market environment fully.

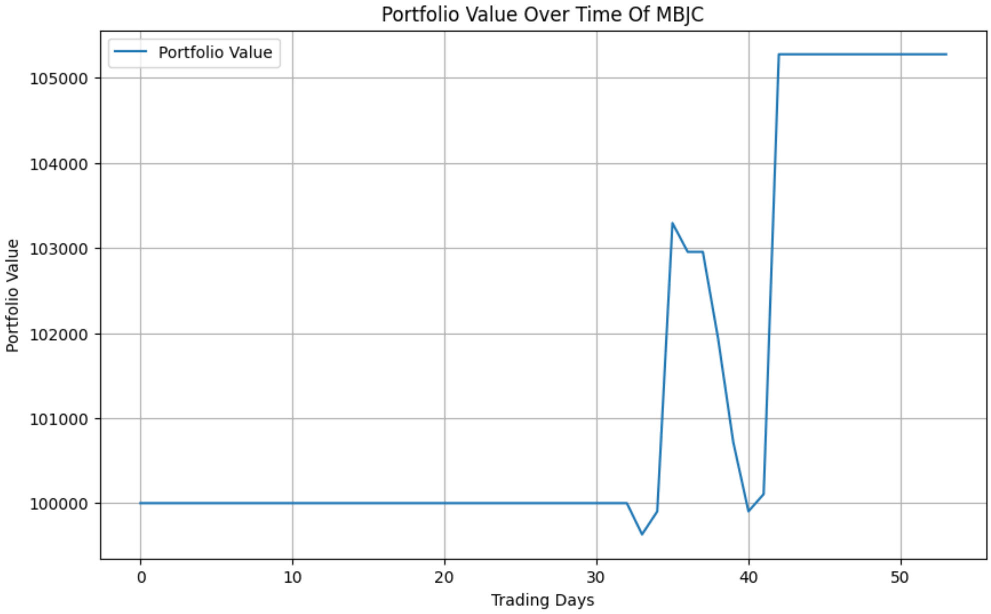

Figure 4.10: Portfolio value of MBJC

4.4 Stockwise Return Using PPO Method for Case II

For Case II evaluation period spanned from January 2, 2022, to July 29, 2024 for 21 hydropower stocks. Table 4.2 presents a comprehensive overview of the performance metrics for 21 hydropower stock during the testing phase of our trading model. The testing was conducted with an initial balance of Rs 100,000, allowing the agent to execute trades within a custom trading environment.

Table 4.2: Return Using PPO Method For Hydropower Stocks For Case II

Company Code	**Final Portfolio Value**	**Cumulative Returns (%)**	**Annual Return (%)**	**Annual Volatility (%)**	**Sharpe Ratio**	**Maximum Drawdown (%)**
AHPC	76667.72	-23.33	-14.90	36.01	-0.12	-52.32
AKJCL	108730.48	8.73	5.21	37.20	0.29	-34.64

Continued on next page

Company Code	**Final Portfolio Value**	**Cumulative Returns (%)**	**Annual Return (%)**	**Annual Volatility (%)**	**Sharpe Ratio**	**Maximum Drawdown (%)**
AKPL	67174.57	-32.83	-21.47	32.97	-0.33	-56.61
API	119386.84	19.39	11.36	31.43	0.39	-40.27
BARUN	103785.47	3.79	2.28	36.05	0.22	-45.91
BPCL	98321.49	-1.68	-1.02	21.86	0.07	-24.41
CHCL	149129.95	49.13	27.47	16.66	1.09	-21.83
CHL	105649.78	5.65	3.39	35.20	0.24	-45.51
DHPL	112532.87	12.53	7.43	34.05	0.31	-40.81
HPPL	97932.15	-2.07	-1.26	38.47	0.17	-43.96
KKHC	129044.45	29.04	16.75	33.40	0.48	-34.51
KPCL	111454.32	11.45	6.83	33.13	0.30	-33.28
NGPL	110920.12	10.92	6.50	33.65	0.29	-48.28
NHDL	104899.94	4.90	2.95	27.75	0.21	-34.09
NHPC	137960.29	37.96	21.58	31.44	0.58	-36.76
PMHPL	97938.50	-2.06	-1.26	34.00	0.14	-46.54
RADHI	111782.85	11.78	7.00	31.49	0.30	-32.31
SHPC	100982.17	0.98	0.60	23.69	0.13	-37.69
SPDL	95605.27	-4.39	-2.69	34.98	0.12	-51.03
UMHL	107395.45	7.40	4.43	36.99	0.26	-38.60
UPPER	61180.40	-38.82	-25.80	33.11	-0.44	-63.85
Total	**2208475.09**	**5.17**	**2.64**	**32.07**	**0.22**	**-41.11**

4.5 Output For Various Scenario Case II

4.5.1 Best Case Scenario

In Case II, the best case scenario was seen in Chilime Hydro power Company Limited (CHCL), over a period from 2 January, 2021, to 29 July, 2024, the agent performed 11 trades in different scenarios with 7 winning trades and increased the portfolio value from Rs 1,00,000 to Rs 179478.50. This represents a total return of 79.48% of annual return.

The agent's strategy for CHCL was much more successful, making more consistent and profitable trades. The positive Sharpe ratio of 1.10 reflects a well-optimized balance between risk and return. The smaller maximum drawdown of 21.57% indicates that the agent effectively managed risks, likely employing better exit strategies or being more cautious in its trading approach.

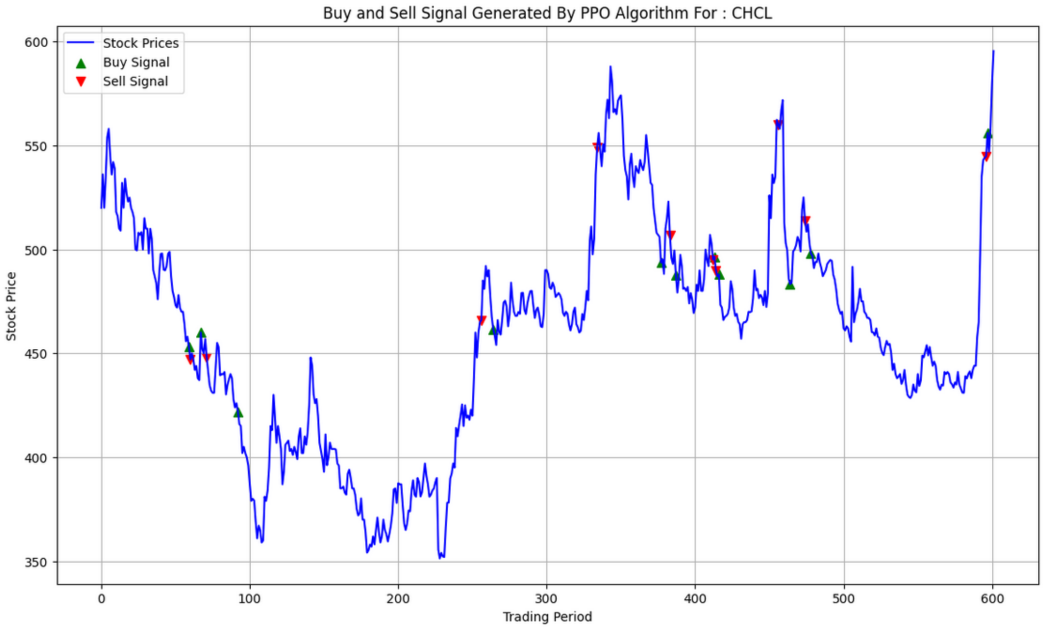

Figure 4.11: Best Case Scenario CHCL Case II

4.5.2 Worst Case Scenario

The worst performance is seen in Upper Tamakoshi Hydropower Limited (UPPER). For UPPER, the agent's strategy resulted in a significant portfolio loss of 52.66%, with a final value of Rs47339.08 from an initial Rs100,000. This underperformance, reflected in a -36.50% annual return and high volatility of 35.67%, suggests that the agent likely misjudged the market conditions failing to adapt to price movements effectively. The negative Sharpe ratio of -0.68 indicates that the agent's strategy did not compensate for the risk taken, leading to substantial losses. A maximum drawdown of 69.15% further emphasizes the failure to mitigate significant losses, possibly due to a lack of proper stop-loss mechanisms or an over-reliance on high-risk trades.

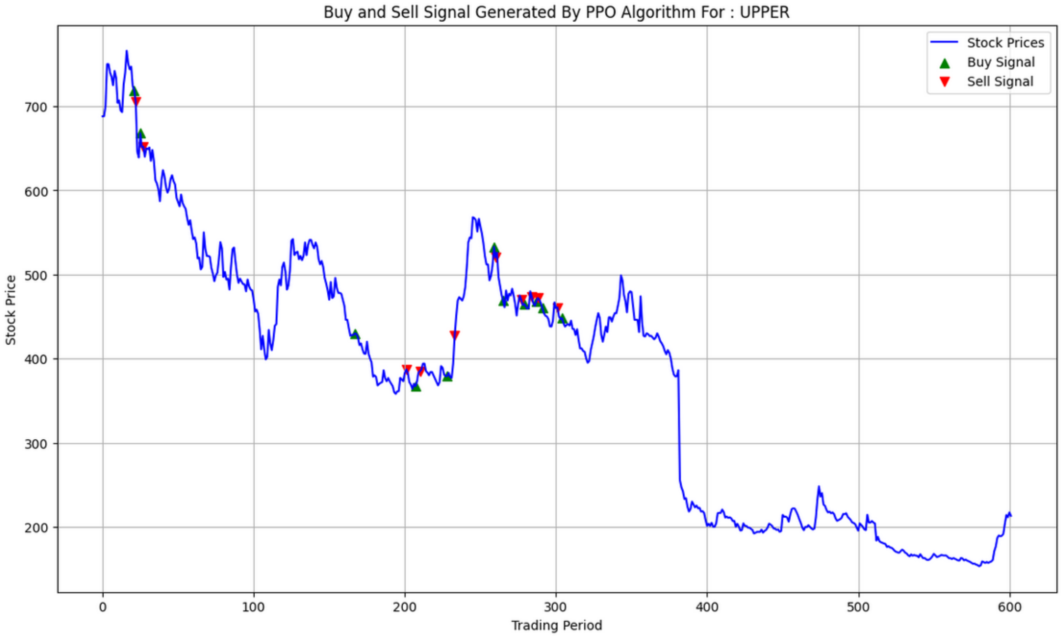

Figure 4.12: Worst Case Scenario UPPER Case II

4.6 Performance of the Model

4.6.1 Train Loss of Agent

Figure 4.13 shows that the training loss fluctuating at the beginning of the training and then decreasing continuously steadily. This pattern suggests that the loss of agent is decreasing at steady rate learning to maximize the reward.

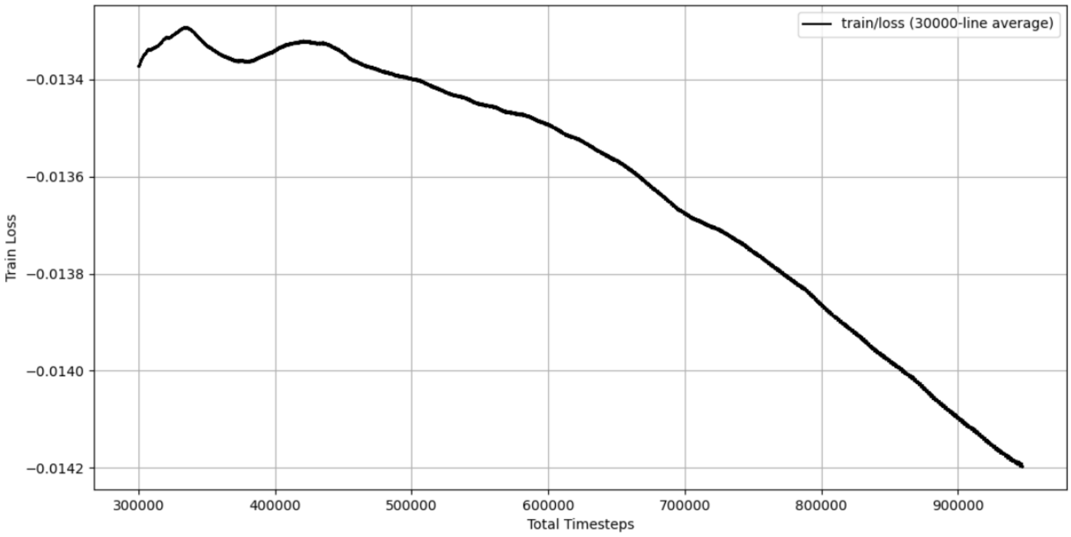

Figure 4.13: Train Loss of PPO Algorithm

As training progresses, however, the fluctuations in the loss diminish, and a clear downward trend emerges. This steady decrease in the loss indicates that the agent is effectively learning and adapting to the environment. The reduction in loss suggests that the agent is gradually optimizing its strategy and improving its ability to maximize the reward. The continuous decrease in loss reflects a more stable and efficient learning process, where the agent's actions are increasingly aligned with the goal of maximizing overall performance.

Overall, this pattern of initial fluctuation followed by a steady decline in training loss is a positive sign. It demonstrates that the agent is making significant progress in its learning journey, moving towards more effective and consistent decision-making as it gains experience and refines its approach.

4.6.2 Train Value Loss of Agent

The train value loss in figure 4.14 illustrates a pattern where the training value loss starts high, then decreases significantly, with some fluctuations toward the end. Initially, the high value loss suggests that the model's predictions for the value function were quite inaccurate, meaning the model struggled to estimate the true value of its actions early in the training process.

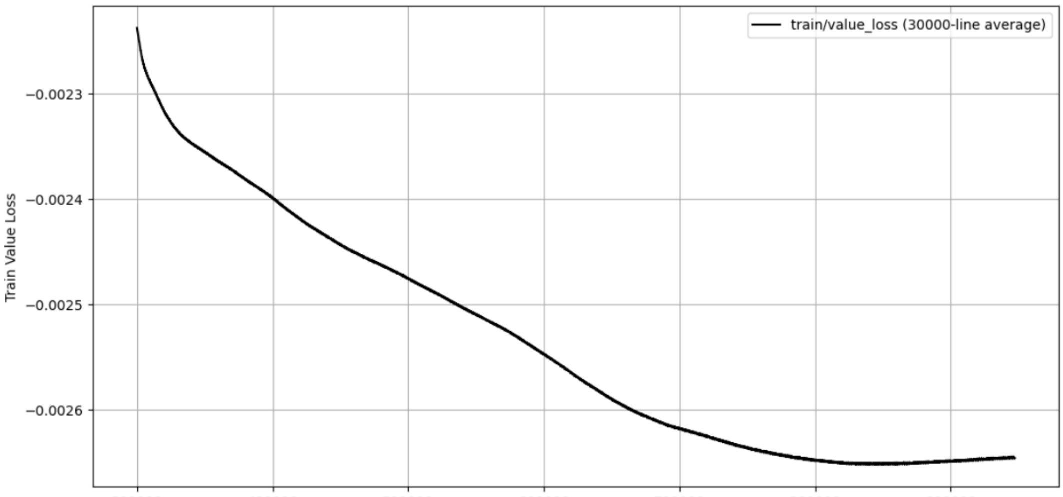

Figure 4.14: Train Value Loss of PPO Algorithm

As the training progresses, there is a sharp decrease in value loss, indicating that the model is learning and its predictions are becoming more accurate. This drop reflects the model's ability to adjust and improve its understanding of the value function, leading to better predictions.

Towards the end of the training, the loss values continue to stabilize.

This trend suggests that the model is becoming more proficient in predicting values, leading to more accurate and reliable performance over time.

4.6.3 Policy Gradient Loss of Agent

Figure 4.15 shows a decreasing trend in policy gradient loss values. This decline indicates that the reinforcement learning model's policy is improving as training progresses.

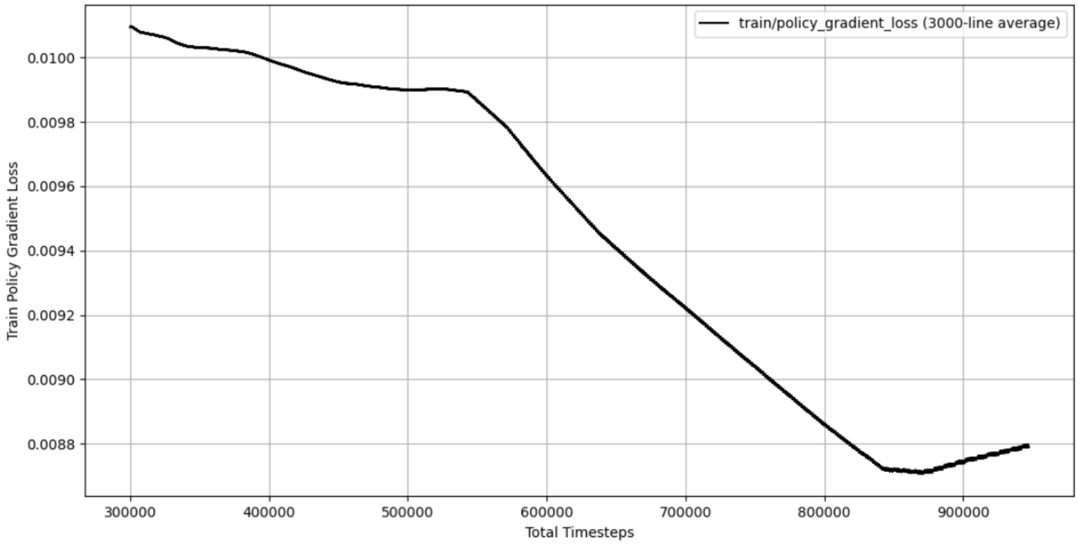

Figure 4.15: Policy Gradient Loss of PPO Algorithm

Initially, the higher values suggest that the policy was less effective, meaning the model's decisions were not well-aligned with maximizing rewards. As training continues, the loss decreases, which reflects improvements in the model's decision-making process and a gradual convergence toward an optimal policy.

The overall trend of decreasing policy gradient loss demonstrates that the model is successfully optimizing its strategy over time, leading to a more refined and effective policy.

4.6.4 Entropy Loss of the Agent

Figure 4.16 illustrates an increasing trend in entropy loss, moving from -0.54 to -0.49, indicating that the entropy of the policy's action distribution is becoming less negative over time. In reinforcement learning, entropy loss plays a crucial role in encouraging exploration by penalizing overly deterministic behavior and rewarding randomness in action selection.

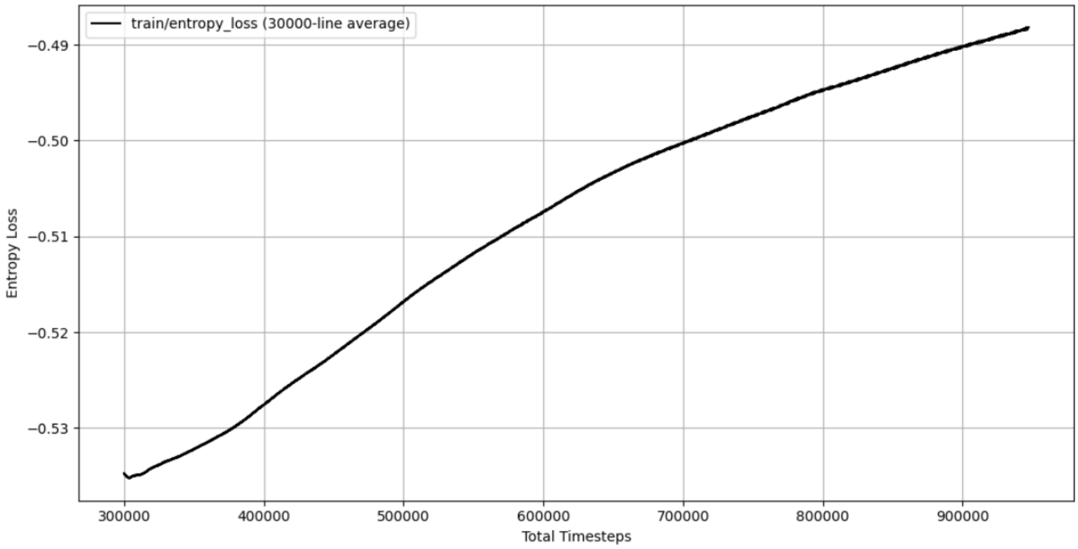

Figure 4.16: Entropy Loss of PPO Algorithm

Initially, the more negative value of -0.54 suggests that the policy had a higher level of randomness, meaning the model was exploring a wide range of actions with less certainty in its choices.

The increasing entropy loss trend highlights the model's transition from exploration to exploitation, where it is refining its strategy and making more consistent, targeted decisions based on its accumulated knowledge.

4.6.5 Average Reward of Agent

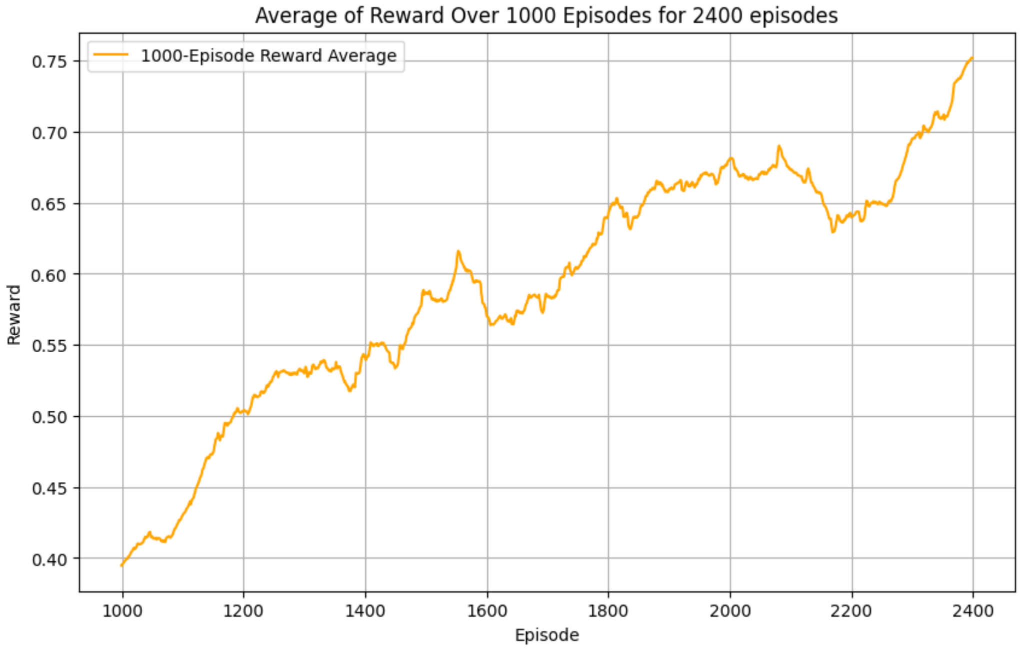

Figure 4.17: Average Reward of Agent

Figure 4.17 shows that as the model trains over 2400 episodes the average reward increases.The increase in average reward from 0.4 to 0.75 reflects its improving ability to make better trading decisions. Initially, the model explores various strategies, often leading to suboptimal outcomes and lower rewards. However, as it gains more experience, it starts to recognize patterns and adjust its actions to maximize profits, leading to a steady increase in average rewards. This progression shows that the model is effectively learning to optimize its trading strategy over time.

4.6.6 Distribution of Action of the Agent

Figure 4.18 and Figure 4.19 shows the distribution of action of the agent at the beginning of the training and after training.

At the beginning of training, the distribution of actions for the stock trading agent is more random and exploratory with about evenly distribution of all the action. The agent performed buy, sell, and hold action with similar frequency, as it is still learning the best strategies to maximize returns.

This randomness is a key part of the exploration phase, where the agent tests different actions to understand their outcomes. The agent might make some poor decisions, such as buying at the wrong time or holding onto losing stocks, leading to a wide variety of actions with no clear pattern or preference.

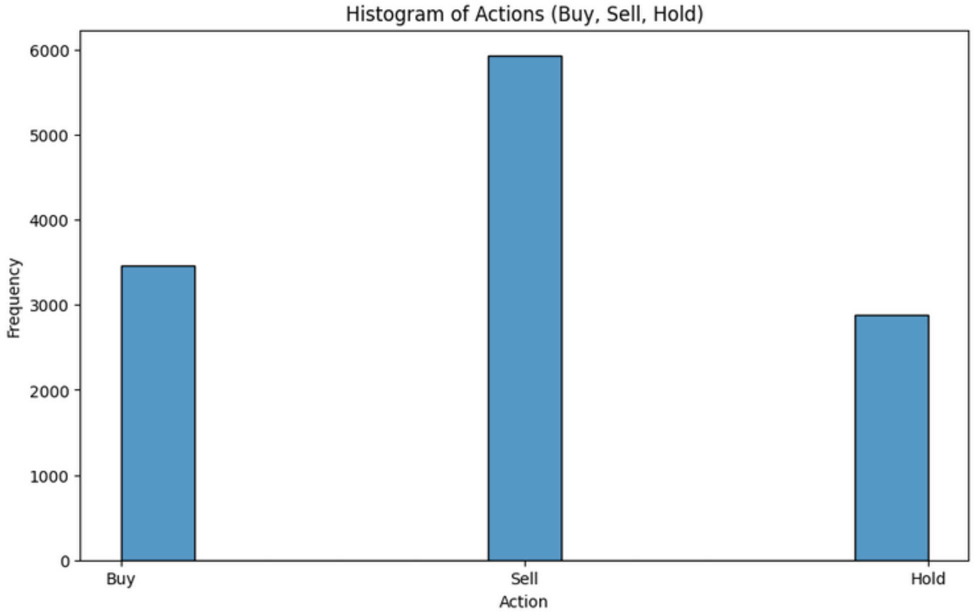

Figure 4.18: Distribution of Action at Beginning of Training

As training progresses and the agent becomes more knowledgeable, the distribution of actions shifts significantly. By the end of training, the agent's actions are more calculated and focused on maximizing profitability as it is shown in figure 4.19.

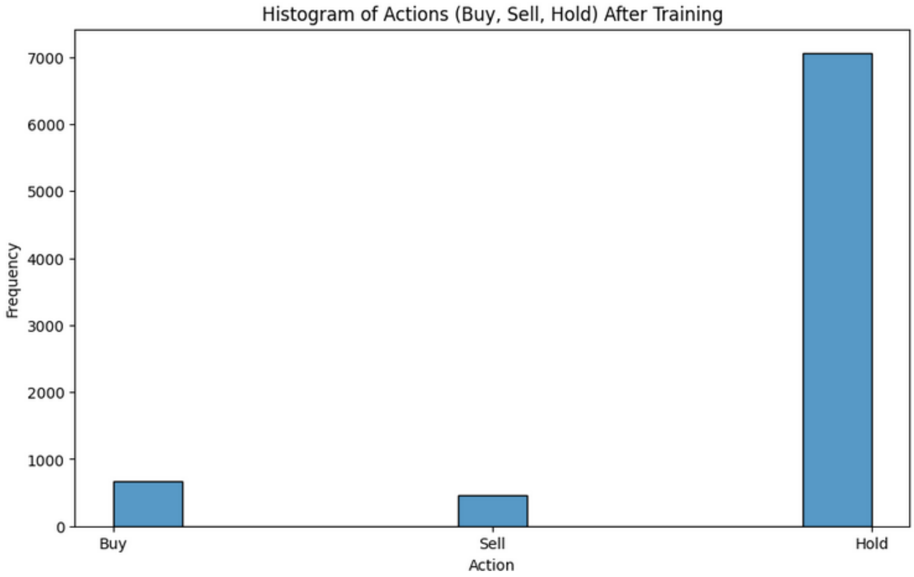

Figure 4.19: Distribution of Action After Training

The agent show a stronger tendency to hold stocks during favorable market conditions. The frequency of buy and sell actions becomes more strategic and holding action becoming more dominant, reflecting the agent's improved understanding of market dynamics. Overall, the action distribution at the end of training is less random and more aligned with the agent's learned strategy, showing a preference for actions that have historically led to higher rewards.

5 DISCUSSION AND ANALYSIS

5.1 Comparison Between Theoretical and Simulated Outputs

In this section, we compare the theoretical expectations of our stock trading agent with the results obtained from simulated trading using the NEPSE dataset. This comparison aims to identify and understand any discrepancies between theoretical predictions and actual performance, providing insights into the effectiveness and limitations of our trading strategy.

5.1.1 1. Theoretical Expectations:

- **Trading Strategy:** The agent employs a comprehensive trading strategy based on both trend-following and momentum principles, utilizing a range of popular technical indicators. Trend-following elements include Moving Averages (MA) and Exponential Moving Averages (EMA), which help identify the market's overall direction by smoothing price data over different periods.

  These indicators signal potential buying or selling opportunities based on crossovers, with short-term EMAs reacting more swiftly to recent price changes compared to longer-term SMAs. Additionally, the strategy incorporates the Moving Average Convergence Divergence (MACD) to capture shifts in market momentum and the Relative Strength Index (RSI) to detect overbought or oversold conditions, providing further guidance on entry and exit points.

  To complement these trend-following tools, the strategy includes volatility and band indicators such as Bollinger Bands, which adjust to market volatility and highlight potential breakout or reversal points. The agent also utilizes additional indicators like the Stochastic Oscillator and On-Balance Volume (OBV) to gain a more nuanced understanding of market conditions. By combining these diverse technical indicators, the strategy aims to optimize trading decisions, capitalize on prevailing trends, and manage risks effectively.

- **Expected Performance:**

  - **Profitability:**

    The theoretical framework for the trading strategy anticipates substantial profitability through its trend-following approach. By leveraging indicators such as Moving Averages (MA) and Exponential Moving Averages (EMA), the strategy is designed to identify and capitalize on prevailing market trends. During bullish trends, the strategy aims to generate profits by executing buy orders as the market moves upward. Conversely, during bearish trends, it seeks to lock in gains or avoid losses by initiating sell orders. The expectation is that by aligning trades with these trends, the strategy will consistently realize gains from sustained market movements, thus enhancing overall profitability.

- **Risk-Adjusted Returns:** The application of technical indicators such as the Moving Average Convergence Divergence (MACD) and Relative Strength Index (RSI) is anticipated to yield favorable risk-adjusted returns. MACD helps identify changes in trend direction and momentum, while RSI provides insights into overbought or oversold market conditions. By incorporating these indicators, the strategy is expected to optimize performance by capturing significant price movements while managing risk. Theoretical predictions suggest that this approach should result in a high Sharpe Ratio, reflecting strong returns relative to the volatility of those returns. Additionally, the strategy aims for low volatility, indicating stable and consistent performance.
- **Drawdown Management:** Effective management of drawdowns is a crucial component of the strategy's expected performance. By utilizing indicators that signal potential market reversals or corrections, such as Bollinger Bands and other volatility measures, the strategy is designed to mitigate the impact of significant downturns. Theoretical expectations are that these indicators will provide early warnings of adverse market conditions, allowing the agent to adjust its positions accordingly. This proactive approach aims to protect the portfolio from severe losses, thereby minimizing drawdowns and enhancing the overall risk management of the trading strategy.

5.1.2 Simulated Outputs

The simulation results closely match the expected performance metrics. Starting with an initial investment in each of the hydropower stocks, the agent's portfolio grew a lot in Case I because of a strong market. In another case, the portfolio still grew, but not as much, even though the market was bearish. This growth surpasses the expected profitability in Case I, demonstrating that the agent effectively captured market trends. The annualized return is notably high in Case I, indicating that the strategy capitalized well on prevailing market conditions.

However in Case II, because of the declining market, the agent was not able to grow the portfolio much in 3 years. Which shows that the simulation does not match the theoretical Output. In theory, the buying should not be done in declining market however the agent traded in declining market and could not grow the portfolio well.

The simulation results for Case I exceeded theoretical expectations in terms of profitability and risk-adjusted returns, and they aligned well with drawdown management. The portfolio experienced significant growth due to a strong bullish market, demonstrating the strategy's effectiveness in favorable conditions. In contrast, Case II showed limited growth over a longer period due to a prolonged bearish market, revealing the strategy's challenges in adverse conditions.

Case I, saw a significant portfolio increase in a short time due to a strong bullish market, while Case II showed limited growth over approximately three years due to a prolonged bearish market trend.

5.1.3 Key Areas of Discrepancies:

The discrepancy is seen in the bearish market trend where:

- In Case II, in the bearish market the agent's portfolio did not grow much over 3 years and traded during a declining market. This is inconsistent with the theoretical expectation, where the agent should ideally avoid trading in a declining market and capitalize short bull phases if possible.

5.1.4 Reasons for Discrepancies:

The discrepancies between the theoretical expectations and the simulated outputs of the trading strategy could arise from several factors:

- **Suboptimal Risk Management:** The agent have not incorporated sufficient risk management techniques, such as stop-loss orders or position sizing rules, leading to significant losses in a declining market.

- **Ineffective Selling Strategy in Bearish Markets:** The agent continued to buy even in a declining market, which led to underperformance and a failure to grow the portfolio.
- **Hyperparameters :** Hyperparameters like learning rate, clip range, and batch size can significantly impact the agent's performance and contribute to discrepancies between theoretical expectations and simulated outputs.

Understanding and addressing discrepancies between theoretical and simulated outputs are essential for refining and improving trading models. By critically analyzing these differences, insights can be gained into the model's robustness, potential areas of improvement, and adjustments needed to enhance its real-world applicability.

5.2 Error Analysis

In this section, we conduct a thorough error analysis to identify and pinpoint potential sources of error in the model's performance. Understanding these errors is crucial for refining the model, improving its accuracy, and enhancing its predictive capabilities.

5.2.1 Possible Source of Error

The possible source of error for this model could be as follows.

Model Assumptions: Errors can also arise from model assumptions. Simplifying assumptions about market behavior, such as constant volatility or the efficient market hypothesis, may not hold true in reality. Furthermore, complex models may overfit historical data, resulting in poor generalization to new market conditions.

Implementation Factors: Practical implementation factors contribute to errors as well. Execution delays, such as latency in trade execution, can cause trades to be executed at different prices than anticipated. Ignoring transaction costs, like fees or bid-ask spreads, can also underestimate the true cost of trading.

Market Dynamics: Market dynamics introduce another layer of potential errors. Illiquid markets can lead to price slippage and difficulty in executing large trades at desired prices.

Additionally, sudden changes in market volatility can invalidate model assumptions and predictions.

External Factors: External factors play a significant role in error introduction. Macroeconomic events, such as economic releases, geopolitical developments, or regulatory changes, can impact market behavior unpredictably. Behavioral finance aspects, including investor sentiment and psychological factors, can influence market movements beyond the predictions of quantitative models.

5.2.2 Error Mitigation and Model Improvement

The error of the model can be mitigated to improve the performance of the model by applying following strategies.

Data Enhancement: Improving data quality is crucial for model accuracy. This involves rigorous preprocessing techniques, data validation, and cleansing methods. Additionally, incorporating additional relevant data sources can help capture a more comprehensive view of market dynamics.

Model Refinement: Refining the model involves optimizing its parameters and architecture through techniques like cross-validation, hyperparameter tuning, and regularization. Considering ensemble methods or deep learning approaches can further enhance the model's robustness and adaptability.

Risk Management Strategies: Integrating risk management strategies is vital for effective trading models. This includes accounting for transaction costs, slippage, and market impact. Implementing stop-loss mechanisms or portfolio diversification strategies can also help mitigate downside risks.

Adaptive Learning: Adaptive learning techniques allow the model to continuously update and adapt to evolving market conditions. Incorporating feedback loops from trading outcomes can iteratively improve model predictions and decision-making.

Scenario Analysis: Conducting scenario analyses helps assess model sensitivity to different market scenarios and stress-tests its performance under adverse conditions. Identifying potential vulnerabilities and developing contingency plans can proactively mitigate risks.

5.3 Performance Comparison of PPO Algorithm with Traditional Benchmark Strategy

In evaluating the performance of our methodology using the Proximal Policy Optimization (PPO) algorithm for simulated stock trading, it is essential to compare and contrast its outcomes with existing approaches. This section examines the factors contributing to our methodology's comparative advantage or disadvantage over traditional and contemporary works in the field.

5.3.1 Benchmark Strategies Used to compare with PPO Agent

To analyse the PPO agents performance, we compare its returns with other benchmark trading strategy like, Buy and Hold, MA Crossover, EMA Crossover, Bollinger Band, RSI Strategy, SO Strategy and MACD and Signal Line Crossover Strategy.

All the Trading Strategy are described below.

The **Buy and Hold strategy** involves purchasing an asset and holding it for an extended period, regardless of market fluctuations. This approach aims to benefit from long-term price appreciation and avoid the complexities of timing the market.

Bollinger Bands are calculated by taking a 20-day rolling mean of the price to form the middle band, then adding and subtracting a multiple of the 20-day rolling standard deviation to create the upper and lower bands, respectively. When price reach the upper band we sell and when price reach the lower band we buy.

MA Crossover strategy we used MA 5 and MA 20 generates buy signals when the MA 5 crosses above the MA 20, indicating a bullish trend, and sell signals when the MA 5 crosses below the MA 20, suggesting a bearish trend.

EMA Crossover strategy we used EMA 5 and EMA 20 generates buy signals when the EMA 5 crosses above the EMA 20, indicating a bullish trend, and sell signals when the EMA 5 crosses below the EMA 20, suggesting a bearish trend.

MACD trading strategy generates buy signals when the MACD Line crosses above the Signal Line, indicating a bullish trend, and sell signals when the MACD Line crosses below the Signal Line, suggesting a bearish trend. MACD Line is calculated by subtracting the long-term EMA (28-day) from the short-term EMA (14-day). This is the 9-day EMA of the MACD Line, used to generate buy and sell signals.

RSI (Relative Strength Index) trading strategy generates buy signals when the RSI crosses above 30, indicating that the asset is coming out of oversold conditions and may be poised for a rebound. Conversely, sell signals are generated when the RSI crosses below 70, suggesting that the asset may be overbought and could be due for a pullback.

Similarly, the **Stochastic Oscillator trading strategy** generates buy signals when the oscillator's value crosses above 20, indicating the asset is emerging from oversold conditions. Conversely, sell signals occur when the oscillator's value crosses below 80, suggesting the asset is entering overbought conditions and may be due for a decline.

5.3.2 Comparison of PPO Agent with Benchmark Strategy For Case I

Table 5.1 compares the performance of different trading methods, including Bollinger Bands, Buy and Hold, EMA, MA, MACD, RSI, Stochastic Oscillator (SO), and PPO for Case I. It shows metrics such as Final Portfolio Value, Cumulative Return, Annual Return, Annual Volatility, Sharpe Ratio, and Max Drawdown for each method. These metrics provide insight into each strategy's overall profitability, risk-adjusted returns, and volatility.

Table 5.1: Comparison of Return of PPO algorithm with Buy and Hold and other Strategy For Case I

Method	**Final Portfolio Value**	**Cumulative Return (%)**	**Annual Return (%)**	**Annual Volatility(%)**	**Sharpe Ratio**	**Max Drawdown (%)**
Bollinger Band	9268110.72	1.85	11.27	3.12	0.47	-0.52
Buy and Hold	12165539.40	33.69	328.68	43.22	7.48	-10.15
EMA	10972605.09	20.58	192.83	35.65	4.52	-9.68
MA	11019553.12	21.09	190.91	37.16	4.55	-9.93
MACD	10620704.16	16.71	137.16	36.28	3.32	-10.24
RSI	9338826.51	2.62	15.23	5.17	0.65	-1.07
SO	9556533.57	5.02	29.59	13.37	1.49	-3.29
PPO	10265134.00	12.80	84.76	29.77	1.89	-8.24

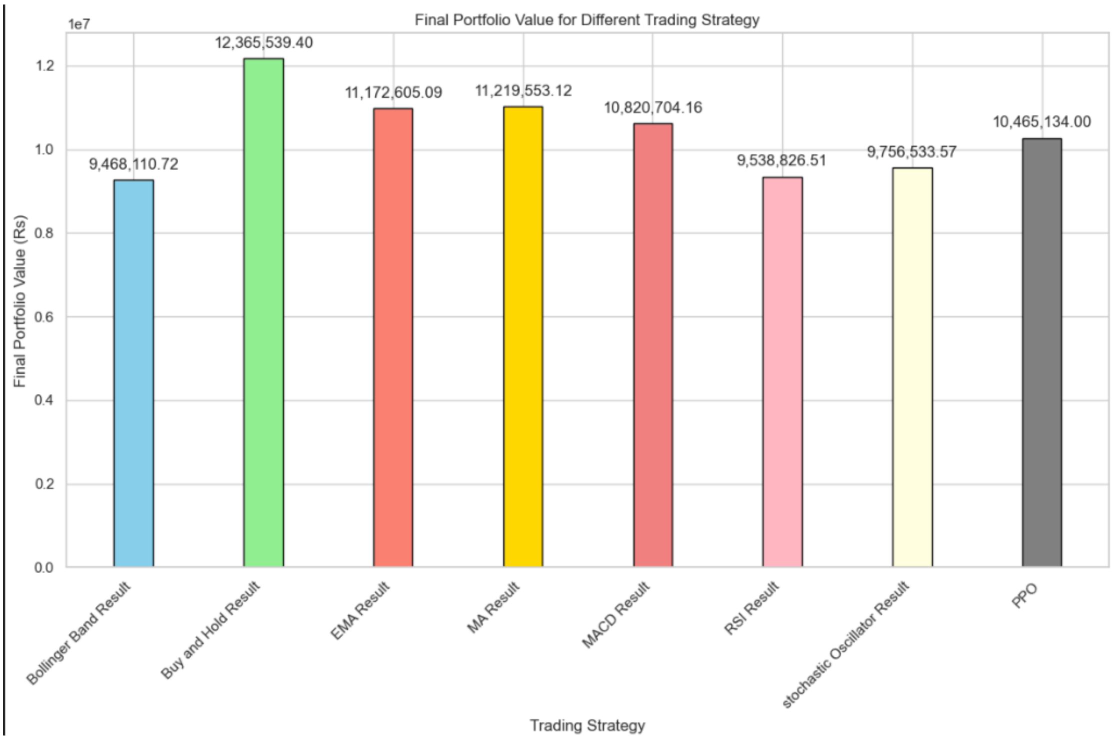

Figure 5.1: Bar Graph of Return of Different Strategy Case I

Figure 5.1, shows the comparison of return of all the strategies in a bar graph.

The **Buy and Hold** method resulted in the highest final portfolio value and cumulative return. However, it also exhibited the highest annual volatility, which led to a relatively lower Sharpe Ratio despite the strong returns. The maximum drawdown was significant at -10.15%, indicating potential for large losses.

The **Bollinger Band** strategy underperformed with a low final portfolio value and cumulative return. The Sharpe Ratio was quite low at 0.47, and the maximum drawdown was slightly negative, reflecting minimal success in limiting losses.

The **EMA** and **MA** strategies both showed moderate performance with relatively high annual returns and reasonable Sharpe Ratios. Their maximum drawdown values were also moderate, indicating a balanced risk-return profile.

The **MACD** strategy achieved a decent final portfolio value and annual return but had a lower Sharpe Ratio compared to EMA and MA, suggesting slightly lower risk-adjusted returns.

The **RSI** strategy performed poorly, with a low final portfolio value and cumulative return. The Sharpe Ratio was low at 0.65, and the maximum drawdown was minimal, indicating that the strategy did not take on much risk, but also did not generate significant returns.

The **SO (Stochastic Oscillator)** strategy produced better returns than RSI but still underperformed compared to others like EMA and MA. The Sharpe Ratio and maximum drawdown suggest moderate success in risk management.

The **PPO** method achieved a final portfolio value of 10,265,134.00 and a cumulative return of 12.80%. The annual return of 84.76% indicates significant yearly gains, though the annual volatility of 29.77% suggests that these returns came with substantial risk.

The Sharpe Ratio of 1.89 shows that PPO provided a decent return for the amount of risk taken, though it's lower compared to some other methods. A maximum drawdown of -8.24% indicates that PPO experienced some drawdowns, but these were not as severe as those seen with the Buy and Hold strategy.

The Buy and Hold strategy generated the highest returns, primarily because the test conditions were in a bull market, which naturally favored this approach. However, it also came with significant volatility and risk. Strategies like EMA and MA provided a more balanced approach, achieving good returns with moderate risk. Indicators like RSI and Bollinger Bands, on the other hand, were less successful in this market environment.

The PPO method offered solid returns with a good balance of risk, as indicated by its Sharpe Ratio. Although it didn't achieve the highest returns overall, PPO presented a safer alternative with moderate drawdowns. This makes it a viable option for those who prioritize risk-adjusted returns while managing potential volatility.

The performance of PPO highlights its effectiveness in providing a balanced approach, especially for investors who are concerned with managing risk alongside achieving reasonable returns.

5.3.3 Analysis of Each Strategy's Overall Profitability, Risk-Adjusted Returns, and Volatility For Case I

Understanding the performance of various trading strategies in terms of profitability, risk-adjusted returns, and volatility is crucial for assessing their effectiveness in different market conditions.

This analysis delves into these aspects by comparing the Proximal Policy Optimization (PPO) algorithm against traditional technical indicators and the Buy and Hold strategy. By examining key metrics such as final portfolio value, cumulative return, Sharpe ratio, and maximum drawdown, the study provides insights into how each strategy navigates the complexities of a bullish market, highlighting the strengths and weaknesses of algorithmic versus traditional approaches.

1. Buy and Hold:

- **Profitability:** The Buy and Hold strategy excelled in a bullish market, achieving a final portfolio value of 12,165,539.40 and a cumulative return of 33.69%. This approach fully leveraged the upward market trend by holding positions without trading frequently.
- **Risk-Adjusted Returns:** It recorded an exceptionally high Sharpe ratio of 7.48,

indicating that the returns were well-compensated for the risk involved.

- **Volatility:** Despite having a high annual volatility of 43.22%, typical in a strong bullish market, the substantial returns justified the increased risk.

2. Bollinger Bands:

- **Profitability:** The Bollinger Bands strategy showed weaker performance with a final portfolio value of 9,268,110.72 and a cumulative return of only 1.85%. This strategy is more effective in volatile or range-bound markets rather than a steadily rising one.
- **Risk-Adjusted Returns:** The Sharpe ratio was 0.47, suggesting that the strategy's returns did not sufficiently reward the risk taken, possibly due to premature sell signals.
- **Volatility:** It had a low annual volatility of 3.12% and a minimal maximum drawdown of -0.52%, indicating lower risk but also missed opportunities for profit in a bullish trend.

3. Exponential Moving Average (EMA) & Moving Average (MA):

- **Profitability:** Both EMA and MA strategies performed well in capturing the bullish trend, with final portfolio values of 10,972,605.09 (EMA) and 11,019,553.12 (MA), and cumulative returns of 20.58% (EMA) and 21.09% (MA). They were effective in following the market's upward movement.
- **Risk-Adjusted Returns:** They both achieved strong Sharpe ratios of 4.52 (EMA) and 4.55 (MA), reflecting their ability to manage risk while capitalizing on the trend.
- **Volatility:** Both strategies had moderate annual volatility (35.65% for EMA and 37.16% for MA) and low maximum drawdowns, indicating a good balance between risk and return.

4. Moving Average Convergence Divergence (MACD):

- **Profitability:** The MACD strategy achieved a final portfolio value of 10,620,704.16 and a cumulative return of 16.71%. Although it captured some gains, it underperformed compared to EMA and MA due to more frequent trading signals.
- **Risk-Adjusted Returns:** With a Sharpe ratio of 3.32, the returns were less favorable compared to the risk, possibly due to false signals in a strong uptrend.
- **Volatility:** The annual volatility was moderate at 36.28%, with a maximum drawdown of -10.24%, indicating higher risk without corresponding returns.

5. Relative Strength Index (RSI):

- **Profitability:** The RSI strategy showed a final portfolio value of 9,338,826.51 and a cumulative return of 2.62%. This underperformance is likely due to premature sell signals in a continuously rising market.
- **Risk-Adjusted Returns:** The Sharpe ratio of 0.65 suggests that the returns did not adequately compensate for the risk, as the strategy likely exited positions too early.
- **Volatility:** It had low annual volatility of 5.17% and a small maximum drawdown of -1.07%, showing conservative behavior but at the expense of missing out on the market's gains.

6. Stochastic Oscillator (SO):

- **Profitability:** The SO strategy, with a final portfolio value of 9,556,533.57 and a cumulative return of 5.02%, performed better than Bollinger Bands and RSI but still lagged behind other strategies. It also likely suffered from early sell signals in the bullish market.
- **Risk-Adjusted Returns:** A Sharpe ratio of 1.49 indicates moderate returns relative to the risk, better than Bollinger Bands and RSI but still not optimal.

- **Volatility:** With annual volatility of 13.37% and a maximum drawdown of -3.29%, the SO strategy was moderately risky but did not fully capitalize on the upward trend.

7. Proximal Policy Optimization (PPO):

- **Profitability:** PPO achieved a final portfolio value of 10,265,134.00 and a cumulative return of 12.80%. While it did not match Buy and Hold's performance, it outperformed most traditional strategies by adapting to the bullish market.
- **Risk-Adjusted Returns:** With a Sharpe ratio of 1.89, PPO provided a solid balance between return and risk, benefiting from its adaptive learning capabilities.
- **Volatility:** PPO had moderate annual volatility of 29.77% and a maximum drawdown of -8.24%, reflecting a balanced approach between risk and return.

From the analysis we found that Buy and Hold Strategy was the most profitable strategy in a bullish market, as it capitalized fully on the market's upward trend with just buying at the start and holding the stock till the end. EMA and MA strategies were also effective, capturing most of the market's gains with moderate risk, making them suitable for trend-following in a bullish market. PPO performed well by balancing profitability with risk, making it a solid choice for trading in a rising market, though it was slightly more conservative than Buy and Hold.

Strategies like Bollinger Bands, RSI and SO were less effective in a bullish market, as they tend to signal exits during upward trends, leading to missed opportunities. Their lower returns reflect their inability to fully exploit the sustained market growth.

5.3.4 Performance Evaluation and Methodology Assessment

PPO Algorithm demonstrated notable strengths and some weaknesses, which are critical to understanding its overall performance and potential areas for improvement.

Strengths of PPO Agent Compared to Benchmark Strategies:

- **Risk-Adjusted Returns:** PPO demonstrated a balanced approach to risk and reward, as evidenced by its Sharpe Ratio of 1.89 . Although the ratio were lower than the Buy and Hold strategy, they were competitive and indicated a good balance between returns and risk. The PPO method managed drawdowns effectively with a maximum drawdown of -8.24%, which is less severe compared to the -10.15% seen in the Buy and Hold strategy.

- **Adaptability:** PPO's performance benefits from its adaptability to changing market conditions. Unlike static strategies such as Buy and Hold or simple technical indicators like Bollinger Bands and RSI, PPO continuously learns and refines its trading policy based on new data. This adaptability helps PPO handle volatility better and adjust to varying market scenarios, which is reflected in its moderate drawdown and relatively high annual return.

- **Dynamic Decision-Making:** The PPO algorithm's reinforcement learning approach allows it to make dynamic decisions based on current market conditions, rather than relying on fixed rules. This dynamic approach contributed to PPO's higher annual return of 84.76% compared to many traditional strategies, though it came with higher volatility.

Areas Where PPO Agent Underperformed:

- **Return Comparisons:** While PPO achieved a solid cumulative return of 12.80%, it fell short compared to the Buy and Hold strategy, which yielded a cumulative return of 33.69%. This discrepancy is partly due to the market conditions during the test, which favored a long-term holding approach. PPO's performance, though impressive, did not match the extreme returns of the Buy and Hold strategy.

- **Volatility and Risk:** The annual volatility for PPO was 29.77%, which is relatively high. This high volatility indicates that the PPO strategy experienced considerable fluctuations in returns. In contrast, strategies like EMA and MA provided high returns with lower volatility, suggesting better performance in terms of stability.

- **Data and Strategy Specificity:** PPO's performance can be sensitive to the quality and scope of training data. If the data used does not represent future market conditions well, the PPO model might not perform as expected. Traditional strategies like EMA and MACD use straightforward rules that can be more robust to different data sets, as they rely on well-established technical indicators.

5.3.5 Comparison of PPO Agent with Benchmark Strategy For Case II

Table 5.2 compares the performance of different trading methods, including Bollinger Bands, Buy and Hold, EMA, MA, MACD, RSI, Stochastic Oscillator (SO), and PPO for Case II where we take 21 hydropower stocks and train the model till date December 29, 2021 and test it from January 2, 2022 to July 29, 2024. It shows metrics such as Final Portfolio Value, Cumulative Return, Annual Return, Annual Volatility, Sharpe Ratio, and Max Drawdown for each method. These metrics provide insight into each strategy's overall profitability, risk-adjusted returns, and volatility.

In this case the agent was provided Rs 1,00,000 for each of the hydropower stock to trade totalling Rs 21,00,000 total initial investment.

Table 5.2: Comparison of Return of PPO algorithm with Buy and Hold and other Strategy For Case II

Method	**Final Portfolio Value**	**Cumulative Return (%)**	**Annual Return (%)**	**Annual Volatility(%)**	**Sharpe Ratio**	**Max Drawdown (%)**
Bollinger Band	1565147.74	-25.47	-11.62	30.88	-0.44	-47.52
Buy and Hold	1532797.70	-27.01	-12.83	44.54	-0.35	-62.20
EMA	2144155.54	2.10	0.38	25.24	-0.12	-33.84
MA	1919038.40	-8.62	-4.95	31.08	-0.23	-45.53
MACD	1957961.83	-6.76	-3.32	27.49	-0.24	-40.92
RSI	1437996.09	-31.52	-14.86	34.90	-0.48	-49.76
SO	1385400.95	-34.03	-16.23	32.40	-0.58	-52.06
PPO at Beginning(25%)	1839849.46	-12.39	-8.16	27.45	-0.10	-44.24
PPO at Middle 1(50%)	2190870.45	10.10	5.92	9.78	0.47	-8.78
PPO at Middle 2(75%)	2302858.16	9.66	5.21	25.98	0.29	-31.69
PPO at End(100%)	2449491.83	16.64	8.22	35.83	0.29	-41.77

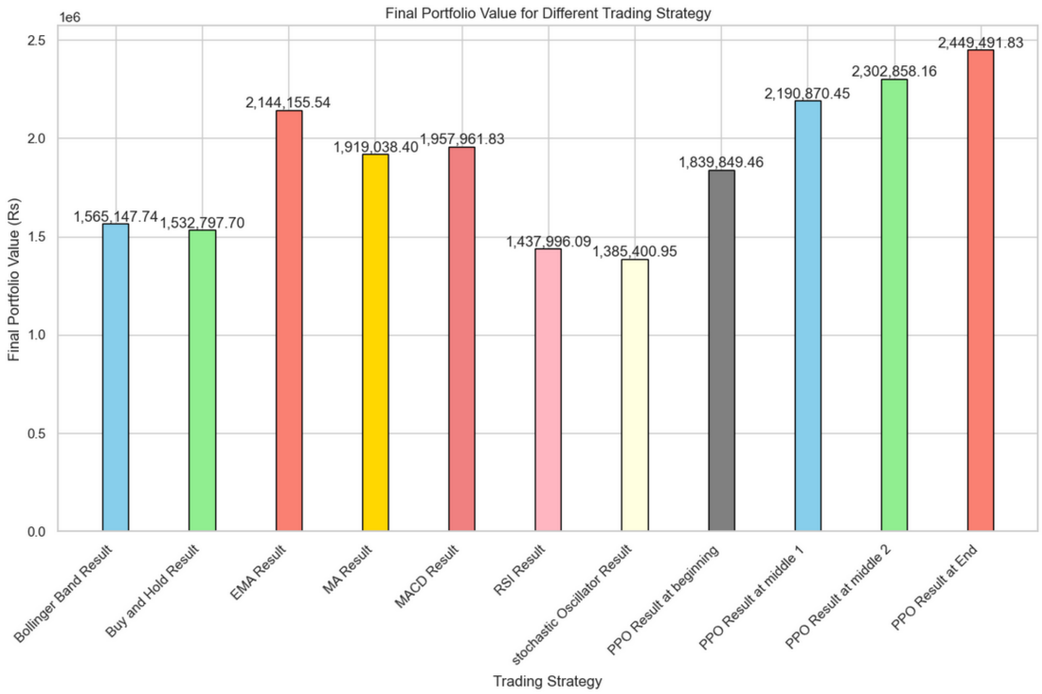

Figure 5.2: Bar Graph of Return of Different Strategy Case II

The trading strategies' performance in fig 5.2 reveals significant differences in their effectiveness for Case II with longer test period.

The **Bollinger Band** strategy ended with a final portfolio value of Rs1,565,147.74, showing a cumulative loss of 25.47% and an annual return of -11.62%. This strategy exhibited a high annual volatility of 30.88%, resulting in a negative Sharpe ratio of -0.44 and a substantial maximum drawdown of 47.52%. Despite being one of the better performers among the losing strategies, its risk-adjusted return was poor, indicating that the strategy did not compensate well for the risk taken.

In contrast, the **Buy and Hold** approach fared slightly worse, with a final portfolio value of $1,532,797.70 and a cumulative return of -27.01%. It had a higher annual volatility of 44.54% and a Sharpe ratio of -0.35, reflecting significant risk without adequate returns. The maximum drawdown was a steep 62.20%, the worst among the strategies, highlighting the inherent risk of simply holding assets through market fluctuations.

The **EMA** strategy stood out with a final portfolio value of Rs2,144,155.54, managing a small cumulative gain of 2.10% and an annual return of 0.38%. This strategy had lower annual volatility at 25.24% and a less negative Sharpe ratio of -0.12, suggesting better risk management. The maximum drawdown of 33.84% was also moderate compared to other strategies, indicating a more resilient approach during market downturns.

The **MA** and **MACD** strategies, while performing better than the Bollinger Band and Buy and Hold approaches, still ended in losses. The MA strategy had a final portfolio value of Rs1,919,038.40 with a cumulative return of -8.62%, and the MACD strategy closed at Rs1,957,961.83 with a cumulative return of -6.76%. Both strategies showed moderate volatility and drawdowns but failed to achieve positive returns, with Sharpe ratios of -0.23 and -0.24, respectively, indicating suboptimal risk-adjusted performance.

On the lower end, the **RSI** and **Stochastic Oscillator (SO)** strategies performed poorly. The RSI strategy had a final portfolio value of Rs 1,437,996.09 with a cumulative return of -31.52%, while the SO strategy ended at Rs1,385,400.95 with a cumulative return of -34.03%. Both strategies experienced high volatility and significant drawdowns, leading to negative Sharpe ratios, with the SO strategy being the worst performer overall.

The **PPO** strategy shows four phases: model at 25% train duration, at 50% of train duration, 75% of train duration and the end of training. By the end of training, the PPO strategy demonstrated the best performance with a final portfolio value of Rs 2449491.83, a cumulative return of 16.64%, and an annual return of 8.22%. This was achieved with moderate volatility (35.83%) and a positive Sharpe ratio of 0.29, indicating that the strategy effectively managed risk and delivered positive returns. The maximum drawdown of 41.77%, although significant, was less severe compared to some of the other strategies.

In summary, the PPO strategy at the end of training was the most successful in case II, offering positive returns and a reasonable balance between risk and reward. The EMA strategy also showed resilience, although its returns were modest. Meanwhile, the traditional technical indicators like Bollinger Bands, RSI, and SO, along with the Buy and Hold approach, struggled to provide favorable outcomes, often leading to substantial losses and poor risk-adjusted returns.

5.3.6 Areas for Improvement in the Current Model

- **Data Quality and Quantity:** The performance of the PPO model is closely linked to the quality and diversity of the training data. To improve, we can expand the dataset to include a broader range of market scenarios, such as different economic conditions or market events.

- **Parameter Tuning:** Fine-tuning hyperparameters is crucial for optimizing the PPO model's performance. Conduct systematic experiments with different parameter settings, such as learning rates, batch sizes, and discount factors.

- **Real-Time Adaptation:** To enhance the model's ability to adapt to rapidly changing market conditions, we can also consider implementing real-time learning techniques. This might involve updating the model more frequently based on recent market data or incorporating adaptive learning rates. Such improvements can help the model stay responsive to sudden market shifts and maintain its effectiveness in dynamic environments.

- **Scalability:** As the model is scaled to handle larger datasets or more complex market environments, performance may suffer. To address this, we can focus on optimizing the model's computational efficiency. This can include techniques like distributed training, reducing the dimensionality of input features, or using more efficient data processing methods. Ensuring that the model remains effective and manageable as its scope expands is key to maintaining performance.

By addressing these areas, the current PPO model can be significantly improved in terms of generalization, adaptability, and scalability, leading to better overall performance in real-world trading scenarios.

5.3.7 Advantages of PPO Algorithm over other methods:

1. **Reinforcement Learning Framework:** Our methodology leverages the PPO algorithm within a reinforcement learning framework, allowing the model to learn and adapt trading strategies iteratively. Unlike traditional approaches that rely on static rules or heuristic strategies, PPO enables dynamic decision-making based on evolving market conditions and feedback from simulated trading experiences.

2. **Adaptability to Market Dynamics:**

   The PPO algorithm excels in handling non-linear and complex market dynamics, including sudden shifts in volatility or investor sentiment. By continuously updating its policy through interactions with the simulated environment, our methodology adapts more effectively to changing market scenarios compared to static models.

3. **Risk Management Integration:**

   Deep Reinforcement Learning incorporates robust risk management strategies directly into the reinforcement learning framework. Through reward shaping and penalty mechanisms, the model learns to prioritize capital preservation while seeking profitable trading opportunities, enhancing overall portfolio resilience.

6 FUTURE ENHANCEMENT

As the stock market evolves, trading strategies must adapt to changing conditions and new information. The Proximal Policy Optimization (PPO) algorithm applied in this project has shown potential in optimizing trading decisions for hydropower stocks on the Nepal Stock Exchange (NEPSE). However, like any model, there is room for improvement and enhancement to increase the strategy's effectiveness and profitability.

To improve the project's results, the dataset can be enriched by incorporating additional data sources such as economic indicators, news sentiment, and alternative data like social media sentiment, which can provide a more comprehensive view of market trends. Expanding the dataset to include all NEPSE stocks will enhance the model's ability to generalize across diverse market conditions. Continuous learning mechanisms and regular dataset updates will ensure the model remains responsive and accurate in evolving market environments.

Adjusting the reward function to account for factors like market impact may also result in more realistic and profitable trading decisions. Lastly, using a combination of different models might capture various aspects of the data, improving overall accuracy.

For future researchers pursuing similar research in stock trading using reinforcement learning, it is crucial to prioritize data quality and model interpretability. Researchers should begin by thoroughly exploring and cleaning the dataset to ensure it accurately represents the market conditions being modeled. Incorporating diverse data sources, such as economic indicators, news sentiment, and alternative data, can enhance the model's understanding of market dynamics. Care should be taken to avoid overfitting, particularly when working with small or specialized datasets like those from NEPSE. Experimenting with different reward functions that closely align with real-world trading objectives is also recommended.

7 CONCLUSION

This project successfully applied the Proximal Policy Optimization (PPO) algorithm to trade hydropower stocks listed on the Nepal Stock Exchange (NEPSE). The major findings indicate that the PPO algorithm can effectively learn and adapt to market conditions, making profitable trading decisions. The integration of technical indicators, combined with the reinforcement learning approach, allowed the model to navigate the complexities of stock trading. The results demonstrated that the algorithm was capable of generating consistent profits, highlighting its potential as a viable trading strategy. Additionally, the visualization tools provided clear insights into the model's decision-making process and performance, further validating the effectiveness of the approach.

The primary objectives of this project were to develop a machine learning model using PPO algorithm that could accurately predict buy, sell, and hold signals for hydropower stocks and to create a trading strategy that maximizes profits while minimizing risks. These objectives were met by first preprocessing the raw data, then applying the PPO algorithm to learn trading strategies, and finally analyzing the model's output to assess its performance. The project achieved its goals by demonstrating that the PPO algorithm could make informed trading decisions that led to profitable outcomes. Through thorough backtesting and analysis, the project confirmed that the model met the initial expectations, providing a strong foundation for future work in this area.

APPENDIX A

A.1 Project Schedule

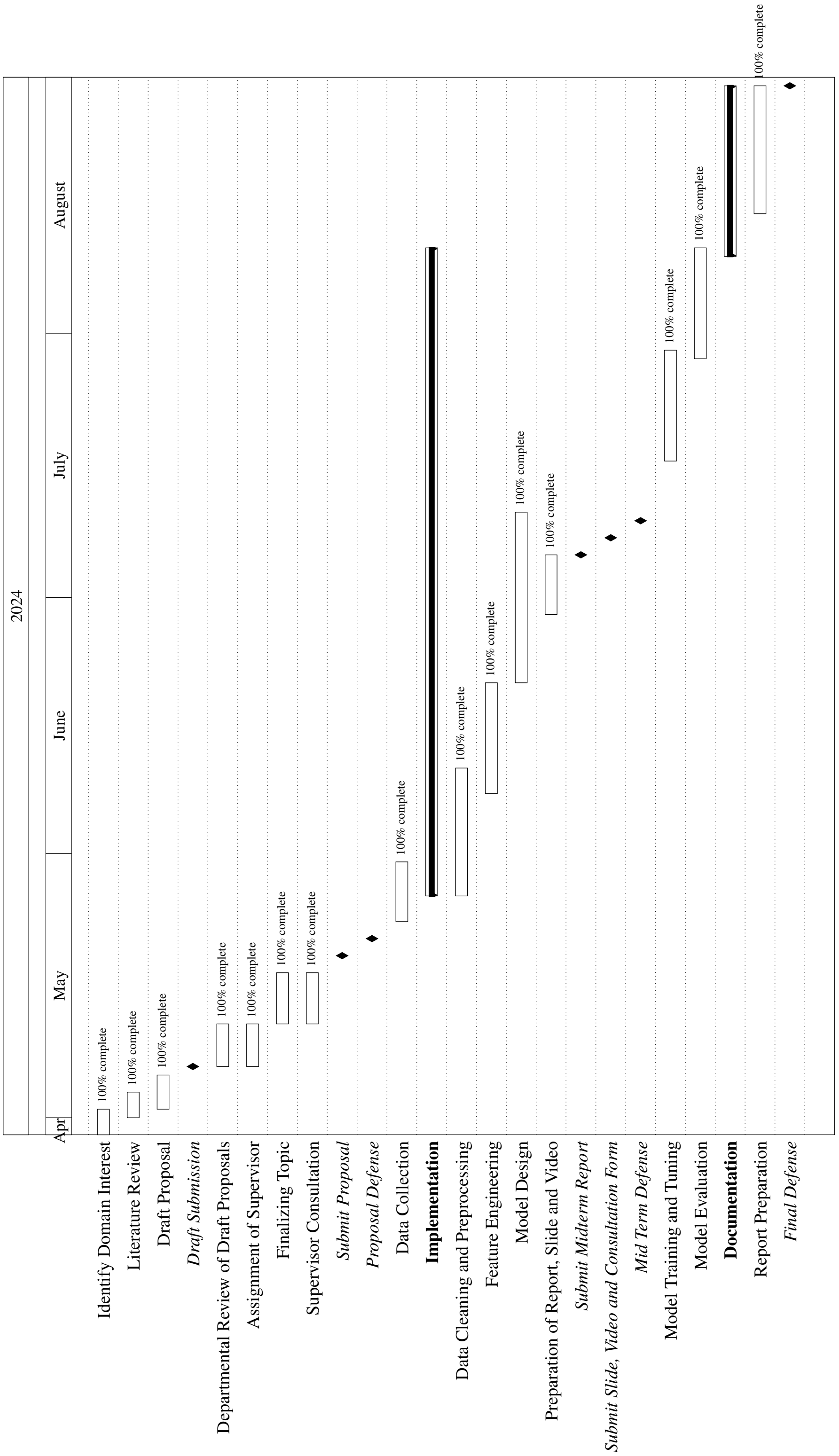

Figure A.1: Gantt Chart showing Expected Project Timeline

A.2 Literature Review of Base Paper- I

Author(s)/Source: Janak Kumar Lal	
Title: Design of Stock Trading Agent Using Deep Reinforcement Learning [3]	
Website: `https://elibrary.tucl.edu.np/JQ990gQIizUxyjI9nB0on90yLkqsGIf4/api/core/bitstreams/9f6fddb8-03a8-4b9c-b2d5-321c85f5c8a2/content`	
Publication Date: September 2022	**Access Date: May, 2024**
Journal: Tribhuvan University Central Library	**Place:** n/a
Volume: n/a	**Article Number:** n/a
Author's position/theoretical position: Phd Student of Tribhuvan University	
Keywords: Reinforcement learning, Double Deep Q-Learning, CNN, NEPSE	
Important points, notes, quotations	**Page No.**
1. employs the Double Deep Q learning algorithm to develop a trading strategy for the stocks of four commercial banks listed in NEPSE.	**17**
2. effectiveness of the Double Deep Q learning agent was evaluated by comparing its performance with that of different baseline trading strategies	**25**
Essential Background Information: develop a stock trading agent that leverages reinforcement learning and technical indicators to optimize trading strategies and improve profitability in the Nepal Stock Exchange (NEPSE).	
Overall argument or hypothesis: a deep reinforcement learning agent, utilizing Double Deep Q-Learning (DDQL), can be effectively designed to optimize stock trading strategies and maximize portfolio returns in the Nepal Stock Exchange (NEPSE) by leveraging technical indicators and a well-constructed stock trading environment	
Conclusion: Double Deep Q-Learning agent, trained on stock data from NEPSE, successfully outperformed baseline models in all tested scenarios, demonstrating its effectiveness in identifying profitable trading strategies through pattern recognition in stock price data	
Supporting Reasons	
1. integrates various technical indicators such as the Stochastic Oscillator, On Balance Volume (OBV), Moving Average Convergence Divergence (MACD), and the Average Directional Index (ADX) to identify trading signals and market trends	**2.** involves training two neural networks (policy and target networks) to reduce overestimation bias and improve decision-making accuracy .
3. Empirical results from experiments demonstrate that the DDQL agent outperforms baseline models in terms of net worth and profitability across different stock data from NEPSE	**4.** learn patterns in stock price data through Convolutional Neural Networks (CNNs) within the policy network, which allows the agent to make informed trading decisions based on these patterns and receive rewards accordingly
Strengths of the line of reasoning and supporting evidence: Integration of multiple technical indicators, an advanced DDQL framework with CNNs, empirical validation on NEPSE stock data, innovative methodologies like experience replay and target networks, and robust quantitative performance metrics demonstrating the agent's superior effectiveness.	
Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: risk of overfitting, narrow evaluation metrics focused primarily on returns, and the complexity of the DDQL algorithm which may hinder practical application and replication.	

A.3 Literature Review of Base Paper- II

Author(s)/Source: Kabbani, Taylan and Duman, Ekrem	
Title: Deep reinforcement learning approach for trading automation in the stock market[4]	
Website: `https://ieeexplore.ieee.org/abstract/document/9877940`	
Publication Date: 05 September 2022	**Access Date: May, 2024**
Journal: IEEE Access	**Place:** n/a
Volume: 10	**Article Number:** pages 93564–93574
Author's position/theoretical position: Engineers and Researchers from Department of Industrial Engineering, Özyein University, Istanbul, Turkey	
Keywords: Autonomous agent , deep reinforcement learning , MDP , sentiment analysis , stock market , technical indicators , twin delayed deep deterministic policy gradient	
Important points, notes, quotations — **Page No.** 1. Formulates the trading problem as a Partially Observed Markov Decision Process (POMDP) **2** 2. Uses the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm. **6** 3. Integrates prediction and decision-making for fully automated trading. **2**	
Essential Background Information: The use of Markov Decision Processes and advanced algorithms like Twin Delayed Deep Deterministic Policy Gradient (TD3), offers a more integrated and effective approach to automating trading by simultaneously addressing prediction and decision-making challenges.	
Overall argument or hypothesis: Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, can significantly enhance trading automation in the stock market by effectively integrating price prediction and portfolio allocation while considering market constraints, thus optimizing investor returns and minimizing risks.	
Conclusion: The model effectively automates stock market trading by combining prediction and decision-making processes, leading to improved investment performance and risk management.	
Supporting Reasons	
1. Combines price prediction and portfolio allocation in a unified approach, addressing both aspects simultaneously.	**2.** Considers market constraints like liquidity and transaction costs, which are often ignored in traditional supervised learning models.
3. The Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm demonstrates strong performance, achieving a Sharpe Ratio of 2.68 on the test dataset.	**4.** Optimizes the overall trading objective of the investor by integrating prediction accuracy with decision-making for portfolio management.
Strengths of the line of reasoning and supporting evidence: Comprehensive integration of price prediction and portfolio management, the robust handling of market constraints, and the demonstrable performance improvements achieved by the TD3 algorithm, as evidenced by the high Sharpe Ratio on the test dataset.	
Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: Potential over-reliance on historical data without sufficient testing in diverse market conditions, and the lack of consideration for unforeseen market disruptions or changes in market behavior that may not be adequately captured by the model.	

A.4 Literature Review of Base Paper- III

Author(s)/Source: Xing Wu, Haolei Chen, Jianjia Wang, Luigi Troiano, Vincenzo Loia, Hamido Fujita

Title: Adaptive Stock Trading Strategies with Deep Reinforcement Learning Methods [5]

Website: `https://www.sciencedirect.com/science/article/abs/pii/S0020025520304692`

Publication Date: 13 May 2020

Access Date: May, 2024

Publisher or Journal: Information Sciences

Place: n/a

Volume: 538

Issue Number: 0020-0255

Author's position/theoretical position: Student and Researcher of School of Computer Engineering and Science, Shanghai University, Shanghai, China

Keywords: stock trading strategy, gated recurrent unit, deep Q-learning, deep deterministic policy gradient

Important points, notes, quotations **Page No.**

1. Gated Recurrent Unit is proposed to extract informative features from raw financial data. **2**
2. Reward function is designed with risk-adjusted ratio for trading strategies for stable returns in the volatile condition. **2**
3. Two adaptive stock trading strategies are proposed for quantitative stock trading. **2**
4. The system outperforms the Turtle trading strategy and achieve more stable returns. **18**

Essential Background Information: Use Gated Recurrent Units (GRUs) to extract informative features from raw financial data and technical indicators, coupled with reinforcement learning strategies GDQN and GDPG, to achieve stable returns in volatile stock markets

Overall argument or hypothesis: Adaptive stock trading strategies utilizing Gated Recurrent Units (GRUs) and deep reinforcement learning methods, specifically GDQN and GDPG, can outperform traditional trading strategies and provide more stable returns in volatile stock markets

Conclusion: The proposed GDQN and GDPG trading strategies, which leverage GRUs for feature extraction and deep reinforcement learning for decision-making, significantly outperform traditional and state-of-the-art methods, offering more stable and robust returns in various market conditions

Supporting Reasons

1. GRUs extract informative features from raw financial data and technical indicators.

2. GDQN and GDPG use reinforcement learning to improve trading decisions based on extracted features .

3. GDQN and GDPG outperform traditional methods in different markets (U.S., U.K., and China) in terms of Sortino ratio and return rate.

4. GDPG shows greater stability and lower loss in volatile markets compared to GDQN.

5. DQN and GDPG strategies outperform the Turtle trading strategy and the DRL trading strategy in terms of rate of return in trending and volatile markets.

6. The strategy adapt to the ever-changing market environment through their self-learning ability, making them effective and efficient in high-density environments

Strengths of the line of reasoning and supporting evidence: The strengths of the line of reasoning and supporting evidence include the effective use of GRUs for feature extraction, the application of reinforcement learning for optimal trading decisions, robust performance across various markets, stability in volatile conditions, superior performance compared to traditional strategies, and adaptive self-learning capabilities in dynamic environments .

Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: The simplification of stock quantities to a nominal value of one share per trade, the omission of transaction costs, the potential for overfitting to historical data, and the lack of consideration for real-world market impact and liquidity constraints.

A.5 Literature Review of Base Paper- IV

Author(s)/Source: Xiao, Chenglin and Xia, Weili and Jiang, Jijiao	
Title: Stock price forecast based on combined model of ARI-MA-LS-SVM[6]	
Website: `https://link.springer.com/article/10.1007/s00521-019-04698-5`	
Publication Date: 2020	**Access Date: May, 2024**
Publisher or Journal: Neural Computing and Applications	**Place:**
Volume: 32	**Issue Number:** 10
Author's position/theoretical position: Student of School of Management, Northwestern Polytechnical University, Xi'an, 710129, Shaanxi, China	
Keywords: Stock price forecasting , Support vector machine, Least squares ,Attribute reduction ,Cumulative auto-regressive moving average	
Important points, notes, quotations **Page no.** 1. ARI-MA-LS-SVM integrates ARI-MA and LS-SVM for stock prediction. **2** 2. Outperforms individual models, indicating superior predictive capability. **7** 3. Multi-model fusion algorithm validated, ensuring universal applicability. **4** 4. Offers practical guidance for investors and regulators, advancing forecasting methodologies.**9**	
Essential Background Information: Study examines a novel combined forecasting model's effectiveness in predicting stock market trends, integrating ARI-MA and LS-SVM methodologies.	
Overall argument or hypothesis: The study posits that the ARI-MA-LS-SVM synthesis model offers superior performance compared to individual forecasting methods, enhancing stock market trend prediction across diverse market conditions. .	
Conclusion: ARI-MA-LS-SVM model provides valuable insights for investors and regulators, offering enhanced predictive accuracy and practical applicability in stock market forecasting.	
Supporting Reasons	
1. Demonstrates consistent superiority over individual models.	**2.** Adaptable across various market conditions.
3. Addresses deficiencies in existing methods, providing comprehensive insights.	**4.** Exhibits stability through consistent performance.
5. Offers practical guidance for investors and regulators.	
Strengths of the line of reasoning and supporting evidence: Empirical evidence demonstrating the ARI-MA-LS-SVM model's superior performance, adaptability, and practical relevance.	
Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: The argument overlooks potential data biases and lacks discussion on practical implementation challenges, casting doubt on the model's real-world applicability in Nepalses context.	

A.6 Literature Review of Base Paper- V

Author(s)/Source: Kyoung-Sook, Moon and Hongjoong, KIM

Title: Performance of Deep Learning in Prediction of Stock Market Volatality[7]

Website: `https://scholarworks.bwise.kr/gachon/handle/2020.sw.gachon/2881`

Publication Date: 2019	**Access Date: May, 2024**
Publisher or Journal: Economic Computation & Economic Cybernetics Studies & Research	**Place:**
Volume: 53	**Issue Number:** n/a

Author's position/theoretical position: Student

Keywords: LSTM,Stock market prediction,Volatility forecasting,Hybrid momentum,

Important points, notes, quotations	Page no.
1. Moon and Kim employed LSTM for stock market prediction.	**1**
2. Study spanned five indices from 2010 to 2016.	**2**
3. Used Technical analysis to predict price like Moving average, exponential moving average, momentum and hybrid momentum.	**3**
4. Hybrid momentum yielded highest accuracy.	**9**

Essential Background Information: Moon and Kim's 2019 study employed LSTM deep learning to predict stock market indices and volatility, highlighting the efficacy of hybrid momentum and the unique characteristics of volatility forecasting.

Overall argument or hypothesis: Moon and Kim hypothesize that utilizing LSTM deep learning with hybrid momentum can accurately predict stock market indices and associated volatility, while emphasizing the distinct nature of volatility prediction compared to index prediction.

Conclusion: Moon and Kim's study underscores the potential of LSTM-based algorithms, particularly when coupled with hybrid momentum, to provide valuable insights into stock market prediction and volatility forecasting

Supporting Reasons

1. Robust dataset spanning seven years across five indices.

2. Independence from traditional financial variables necessitates specialized techniques.

3. Highest prediction accuracy achieved with hybrid momentum.

4. nsights into volatility challenges bolster deep learning's role in financial forecasting.

5. Advanced methods like LSTM required for accurate predictions.

Strengths of the line of reasoning and supporting evidence: Moon and Kim's line of reasoning is strengthened by comprehensive empirical testing across multiple indices, coupled with superior prediction accuracy using hybrid momentum, supporting the efficacy of their LSTM-based approach.

Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: The research focuses on S&P500, NASDAQ, German DAX, Korean KOSPI200, and Mexico IPC, from 2010 to 2016 which lacks recency and doesn't focus on stock exchange of Nepal.

REFERENCES

[1] Xu Wang, Sen Wang, Xingxing Liang, Dawei Zhao, Jincai Huang, Xin Xu, Bin Dai, and Qiguang Miao. Deep reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):5064–5078, 2024.

[2] Yuhui Wang, Hao He, and Xiaoyang Tan. Truly proximal policy optimization. In *Uncertainty in Artificial Intelligence*, pages 113–122. PMLR, 2020.

[3] Janak Kumar Lal. *Design of Stock Trading Agent Using Deep Reinforcement Learning*. PhD thesis, IOE Pulchowk Campus, 2022.

[4] Taylan Kabbani and Ekrem Duman. Deep reinforcement learning approach for trading automation in the stock market. *IEEE Access*, 10:93564–93574, 2022.

[5] Xing Wu, Haolei Chen, Jianjia Wang, Luigi Troiano, Vincenzo Loia, and Hamido Fujita. Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences*, 538:142–158, 2020.

[6] Chenglin Xiao, Weili Xia, and Jijiao Jiang. Stock price forecast based on combined model of ari-ma-ls-svm. *Neural Computing and Applications*, 32(10):5379–5388, 2020.

[7] Moon Kyoung-Sook and KIM Hongjoong. Performance of deep learning in prediction of stock market volatility. *Economic Computation & Economic Cybernetics Studies & Research*, 53(2), 2019.