International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
doi : https://doi.org/10.32628/IJSRST523103163

A Survey on Machine Learning Algorithms for Risk-Controlled Algorithmic Trading

Soham Pathak[1], Antara Pawar[1], Shruti Taware[1], Sarthak Kulkarni[1], Afsha Akkalkot[2]

[1]Third Year Student, Department of Computer Engineering, Zeal College of Engineering and Research, Pune, Maharashtra, India

[2]Assistant Professor, Department of Computer Engineering, Zeal College of Engineering and Research, Pune, Maharashtra, India

ARTICLEINFO

Article History:
Accepted: 10 June 2023
Published: 28 June 2023

Publication Issue
Volume 10, Issue 3
May-June-2023

Page Number
1069-1089

ABSTRACT

Machine learning algorithms have emerged as powerful tools for risk control in algorithmic trading, enabling traders to analyze vast amounts of market data, detect patterns, and make informed trading decisions. In today's fast-paced and data-driven financial markets, effective risk management is essential to navigate market uncertainties and optimize trading performance. Traditional risk control methods often struggle to capture complex market dynamics and adapt to rapidly changing conditions, leading to the adoption of machine learning algorithms. These algorithms excel in processing large volumes of data, uncovering hidden patterns, and making accurate predictions, enabling traders to develop proactive risk management strategies. Machine learning algorithms offer several advantages in risk control for algorithmic trading. They can analyze diverse data sources such as historical price data, news sentiment, and economic indicators, providing valuable insights for risk assessment and decision-making. Additionally, these algorithms can handle time series data, capturing temporal dependencies and adapting to dynamic market conditions. They provide real-time risk monitoring and early warning capabilities, enabling traders to respond quickly to emerging risks and implement risk mitigation measures. Furthermore, machine learning algorithms offer the potential to optimize portfolio management, dynamically adjusting portfolio weights based on risk-return profiles and optimizing asset allocation strategies. Machine learning algorithms have revolutionized risk control in algorithmic trading by providing advanced analytics, predictive capabilities, and real-time monitoring. These algorithms enhance risk management strategies, improve decision-making processes, and enable traders to navigate the complexities of financial markets.

Keywords: Machine Learning, Risk-Controlled, Algorithmic Trading, Support Vector Machines(SVMs), Gradient Boosting Models (GBMs), Random Forests, Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) Networks, Generative Adversarial Networks (GANs), Long Short-Term Memory (LSTM) Networks, Generative Adversarial Networks (GANs), Backtesting, Overfitting.

I. INTRODUCTION

Machine learning algorithms have transformed the landscape of risk control in algorithmic trading by providing advanced analytical tools to tackle complex market dynamics. In today's fast-paced financial markets, traditional risk control methods often struggle to keep up with the ever-changing conditions. Machine learning algorithms offer a solution by analyzing vast amounts of data, extracting meaningful patterns, and making accurate predictions. These algorithms excel in processing time series data, capturing temporal dependencies, and adapting to dynamic market environments. By leveraging machine learning algorithms, traders can enhance risk management strategies, optimize trading decisions, and mitigate potential losses.

Furthermore, machine learning algorithms enable real-time risk monitoring and early detection of market anomalies. By processing streaming market data, these algorithms can detect unusual patterns, identify potential risk events, and provide timely alerts to traders. This empowers traders to respond quickly to emerging risks, adjust their trading positions, and implement risk mitigation measures. Additionally, machine learning algorithms offer the potential to optimize portfolio management by dynamically adjusting portfolio weights based on risk assessments and optimizing asset allocation strategies. This enhances the overall risk control framework and helps traders achieve better risk-adjusted returns in algorithmic trading.

A. Background

Algorithmic trading has gained popularity in financial markets due to its speed and efficiency. However, effective risk control is crucial in automated trading systems. Traditional methods often struggle to handle complex patterns and adapt to changing market conditions. Machine learning algorithms offer advanced data analysis techniques that can learn from historical data, predict market trends, and optimize risk management strategies. These algorithms excel in handling time series data and provide real-time insights for informed decision-making in algorithmic trading.

B. Motivation

Machine learning algorithms are motivated by the desire to improve risk management strategies and trading performance in the complex and dynamic financial market. Traditional methods often struggle to capture intricate market patterns and adapt to changing conditions. By utilizing machine learning algorithms, traders can analyze vast amounts of data, make accurate predictions, and respond quickly to market fluctuations. These algorithms uncover hidden patterns, enhance decision-making, and offer the potential for more robust trading strategies.

C. Objective

The objective of employing machine learning algorithms for risk control in algorithmic trading is to enhance the accuracy and effectiveness of risk management strategies. By leveraging machine learning techniques, the aim is to analyze historical and real-time market data, identify patterns, and develop predictive models that enable traders to make informed decisions to mitigate risks. The objective is to optimize portfolio allocation, detect anomalies and potential risks in real-time, and improve overall risk control measures to achieve better trading performance and reduce potential losses.

II. Risk-Controlled Algorithmic Trading

A. Definition and Concept

Risk-controlled algorithmic trading is a methodology that involves the utilization of automated trading systems equipped with risk management mechanisms to mitigate potential losses and preserve capital during the execution of trading strategies. It entails the application of machine learning algorithms and mathematical models to analyze market data, identify patterns, and make well-informed trading decisions while effectively managing associated risks.

The fundamental concept underlying risk control in algorithmic trading lies in acknowledging the inherent unpredictability and uncertainties prevalent in financial markets. By implementing risk management techniques and harnessing the power of machine learning algorithms, traders aim to minimize potential losses and maximize profitability within predefined risk tolerance thresholds.

B. Importance of Risk-Controlled

Risk control is of utmost importance in algorithmic trading due to the uncertainties and volatility present in financial markets. Effective risk management mechanisms help preserve capital by setting risk limits and employing techniques such as stop-loss orders and position sizing. They ensure stability and consistency in trading performance over time, reducing the likelihood of catastrophic losses. Risk control also aims to optimize risk-adjusted returns by striking a balance between profitability and risk exposure. It mitigates emotional biases by removing subjective decision-making and adhering to predefined rules. Compliance with regulations is facilitated through the implementation of risk management practices. Risk control contributes to the long-term sustainability of trading strategies by avoiding significant drawdowns and attracting investor confidence. Overall, risk control is essential for protecting capital, achieving stability, complying with regulations, and inspiring investor trust in algorithmic trading.

C. Challenges in Risk-Controlled

Implementing effective risk control in algorithmic trading faces several challenges. First, ensuring data quality and availability is crucial, as financial markets produce vast amounts of data that may contain errors or missing values. Second, risk models must be robust and adaptable to changing market conditions and unforeseen events. Third, capturing complex interactions and dependencies between different market factors poses a challenge in risk control modeling. Fourth, accounting for black swan events and tail risks, which have low probabilities but severe consequences, is difficult. Fifth, executing risk control measures in high-frequency trading environments can be challenging due to the need for rapid execution. Sixth, finding the optimal trade-off between risk and reward is a persistent challenge. Seventh, navigating regulatory and compliance requirements adds complexity to risk control practices. Addressing these challenges requires ongoing research, development, and adaptation of risk management techniques while considering market dynamics and incorporating real-time data. Continuous monitoring and evaluation are essential for effective risk control in algorithmic trading.

III. Machine Learning in Algorithmic Trading

A. Overview of Machine Learning

Machine learning (ML) has emerged as a powerful tool in algorithmic trading, revolutionizing the way financial markets are analyzed and trading decisions are made. ML algorithms enable the extraction of insights from vast amounts of historical and real-time market data, empowering traders to make data-driven predictions and enhance their trading strategies.

Machine learning algorithms are designed to automatically learn and improve from experience without being explicitly programmed. They leverage statistical techniques and mathematical models to identify patterns, relationships, and anomalies within complex financial data. By processing large datasets, ML algorithms can uncover hidden patterns and extract valuable insights that may not be readily apparent to human traders.

B. Machine Learning techniques for algorithmic trading

A. Supervised Learning

Supervised machine learning is a subfield of machine learning that plays a significant role in algorithmic trading. It involves training algorithms on labelled historical data, where the desired outputs or targets are known, to make predictions or classifications on new, unseen data.

B. Unsupervised Learning

Unsupervised machine learning is an important aspect of algorithmic trading. It involves training algorithms on unlabelled data to discover patterns, anomalies, and market states. This helps traders in clustering similar stocks, detecting unusual behaviour, reducing data complexity, and identifying market regimes. Proper implementation requires data pre-processing and model evaluation. Unsupervised learning provides valuable insights for informed trading decisions and strategy development.

C. Reinforcement Learning

Reinforcement learning is a promising approach in algorithmic trading. It involves training agents to make sequential decisions to maximize long-term rewards. Agents learn through trial and error, adapting their strategies based on market feedback. Reinforcement learning algorithms can adapt to changing market conditions and discover optimal trading strategies. However, successful implementation requires careful consideration of factors like reward design and risk management. Further research is needed to fully exploit the potential of reinforcement learning in algorithmic trading.

D. Deep Learning

Deep learning is a powerful tool in algorithmic trading. It involves training neural networks to analyze historical market data and make predictions. Deep learning models can predict price movements, assess risks, analyze sentiment, and enable high-frequency trading. They require large amounts of quality data and computational resources for training. Proper data preparation and model tuning are essential for success. Deep learning enhances trading strategies and provides a competitive edge in algorithmic trading.

E. Data Pre-processing and Feature Engineering

Data pre-processing and feature engineering are essential steps in algorithmic trading. Data pre-processing involves cleaning and transforming raw market data to ensure its quality and suitability for analysis. This includes handling missing values, removing outliers, and normalizing data. Feature engineering focuses on selecting and creating informative features that capture relevant market characteristics. Traders need to identify features that have predictive power and can capture meaningful patterns in the data. Effective feature engineering requires domain knowledge and an understanding of the specific trading problem. Well-prepared data and well-engineered features provide a solid foundation for accurate predictions and informed trading decisions. Continuous monitoring and refinement of these processes are necessary to adapt to changing market conditions and ensure the effectiveness of algorithmic trading strategies.

F. Evaluation Metrics for Trading Strategies

Evaluation metrics are essential tools for assessing the performance and effectiveness of trading strategies in algorithmic trading. These metrics provide quantitative measures that enable traders to analyze and compare different strategies based on their profitability, risk, and other relevant factors.

Evaluation metrics used in algorithmic trading:

Return on Investment (ROI): ROI measures the profitability of a trading strategy by calculating the percentage gain or loss relative to the initial investment. It provides a straightforward assessment of the strategy's overall performance.

Sharpe Ratio: The Sharpe Ratio evaluates the risk-adjusted return of a strategy by considering the excess return generated per unit of risk taken. It provides insight into how well a strategy compensates for the level of risk involved.

Maximum Drawdown: Maximum drawdown measures the largest peak-to-trough decline in the strategy's value over a specific period. It represents the maximum loss an investor would have experienced during the strategy's worst-performing period.

Win Rate: Win rate calculates the percentage of winning trades in a strategy. It helps determine the strategy's consistency in generating profitable trade.

Risk-Adjusted Return: Risk-adjusted return measures the return generated per unit of risk taken. It accounts for both the strategy's profitability and its volatility, providing a more comprehensive assessment of its performance.

Alpha and Beta: Alpha measures the excess return generated by a strategy compared to a benchmark, while beta quantifies the strategy's sensitivity to market movements. These metrics help evaluate the strategy's performance relative to the overall market.

IV. RISK MANAGEMENT TECHNIQUES

Following are the Techniques used to reduce risk of algorithmic trading strategy.

A. Stop-Loss Orders: Stop-loss orders are risk management techniques widely used in algorithmic trading to help traders limit their potential losses. A stop-loss order is a pre-defined instruction that automatically triggers a market order to sell a security when its price reaches a specified level, known as the stop price.

The primary purpose of a stop-loss order is to protect traders from significant losses in case the market moves against their positions. By setting a stop price below the current market price for long positions or above the current market price for short positions, traders can limit their downside risk.

When the stop price is reached, the stop-loss order is activated, and a market order is executed to sell the security at the prevailing market price. This allows traders to exit their positions and cut their losses before they escalate further.

Stop-loss orders can be implemented in various ways in algorithmic trading, depending on the specific trading platform and strategy. They can be set based on fixed price levels, percentage thresholds, or technical indicators. Additionally, traders can use dynamic stop-loss orders that adjust their stop prices based on market conditions or the performance of the trading strategy.

While stop-loss orders are effective risk management tools, they also have limitations. During highly volatile market conditions or periods of rapid price movements, the execution price of the stop-loss order may deviate from the expected price, resulting in slippage. Traders should also consider the possibility of false breakouts or temporary price fluctuations that trigger the stop-loss order but do not represent a significant change in the overall market trend.

B. Position Sizing and Portfolio Allocation: Position sizing and portfolio allocation are key components of risk management in algorithmic trading. They involve determining the optimal size of each trade and allocating capital across different trading strategies or assets within a portfolio.

Position sizing refers to the determination of the appropriate quantity or value of a security to buy or sell in a trading position. It aims to balance risk and reward by considering factors such as the trader's risk tolerance, available capital, and the expected risk and return of the trading strategy. Position sizing helps traders control their exposure to individual trades and manage overall portfolio risk.

Several position sizing techniques are commonly used in algorithmic trading, including:

1) **Fixed Fractional Position Sizing**: This technique involves allocating a fixed percentage of the trading capital to each trade. The position size is determined based on the trader's risk tolerance and the expected risk of the strategy. Higher-risk strategies receive a smaller allocation, while lower-risk strategies receive a larger allocation.

2) **Fixed Dollar Position Sizing**: In this approach, a fixed dollar amount is allocated to each trade, regardless of the trading strategy's risk. The position size is determined by dividing the allocated capital by the expected risk per trade.

Portfolio allocation focuses on diversifying capital across different trading strategies or assets within a portfolio. It aims to reduce risk by spreading capital among uncorrelated or negatively correlated strategies, which can help mitigate losses if one strategy underperforms. Portfolio allocation involves determining the optimal allocation weights for each strategy based on factors such as historical performance, risk characteristics, and the trader's investment goals.

In summary, position sizing and portfolio allocation are essential components of risk management in algorithmic trading. They involve determining the appropriate size of each trade and allocating capital across different strategies or assets within a portfolio. Position sizing helps control risk exposure in individual trades, while portfolio allocation diversifies capital to manage overall portfolio risk.

C. Risk Parity Models: Risk parity models are widely used in algorithmic trading to allocate capital among different assets or strategies based on their risk contributions. The goal of a risk parity model is to create a balanced portfolio where each component contributes an equal amount of risk.

In a risk parity model, the allocation of capital is determined by the relative riskiness of the assets or strategies rather than their expected returns. The underlying principle is that diversification should be based on risk rather than the absolute dollar value or market capitalization of the assets.

Risk parity models typically involve the following steps:

1) Risk Measurement
2) Risk Contribution Calculation
3) Capital Allocation

Risk parity models offer a systematic approach to portfolio allocation in algorithmic trading. They aim to create a balanced portfolio where each asset or strategy contributes an equal amount of risk. Risk parity models provide diversification benefits but do not consider expected returns.

D. Value-at-Risk (VaR) and Expected Shortfall (ES): VaR and ES are widely used risk management measures in algorithmic trading. They help traders assess and control the potential losses associated with their trading positions.

Value-at-Risk (VaR) is a statistical measure that estimates the maximum potential loss of a portfolio or position over a given time horizon at a specified confidence level. VaR provides a single number that represents the potential loss in a worst-case scenario.

Expected Shortfall (ES), also known as Conditional VaR (CVaR), goes beyond VaR by providing an estimate of the average loss beyond the VaR level. ES represents the expected value of losses that exceed the VaR threshold. It gives traders an idea of the potential magnitude of losses when extreme events occur.

Both VaR and ES are calculated based on historical data and statistical models. The calculations take into account factors such as volatility, correlation, and confidence levels. As They are based on historical data and statistical assumptions, which may not accurately capture extreme market conditions or unforeseen events.

E. Dynamic Hedging Strategies: Dynamic hedging strategies are commonly employed in algorithmic trading to manage and mitigate risks associated with trading positions. These strategies involve continuously adjusting the hedge positions in response to market movements and changes in the underlying risk factors.

The main objective of dynamic hedging is to offset the exposure to unwanted risks, such as price fluctuations, volatility, or changes in interest rates. By dynamically rebalancing the hedge positions, traders aim to maintain a more stable and controlled risk profile.

Dynamic hedging strategies typically involve the following steps:

Risk Identification: Traders first identify the key risk factors associated with their trading positions. This could include factors such as price movements, interest rate changes, or currency fluctuations.

Risk Monitoring: Traders continuously monitor the market conditions and the behavior of the risk factors that affect their positions. This involves real-time data analysis and the use of algorithms to identify and assess risks.

Hedging Instrument Selection: Based on the identified risks, traders select suitable hedging instruments such as options, futures, or derivatives contracts. These instruments should have a high correlation with the underlying risks and provide effective risk mitigation.

Dynamic Rebalancing: Traders dynamically adjust the hedge positions based on market movements and changes in risk factors. This involves recalculating the optimal hedge ratios and executing trades to rebalance the portfolio.

Risk Assessment and Optimization: Traders regularly evaluate the performance of their dynamic hedging strategies and make adjustments as needed. They analyze the effectiveness of the hedges in reducing risk, optimizing costs, and achieving desired risk-return profiles.

Dynamic hedging strategies offer several benefits in algorithmic trading. They help traders minimize the impact of market fluctuations on their positions, enhance risk management, and potentially increase profitability. By continuously adapting to changing market conditions, these strategies can provide a more robust and flexible approach to risk control.

However, dynamic hedging strategies also have challenges and limitations. They require real-time data feeds, advanced analytics, and efficient execution systems to react quickly to market movements.

V. ML ALGORITHMS FOR RISK CONTROL

A. Support Vector Machines (SVM)

Support Vector Machines (SVM) is a powerful and widely used supervised machine learning algorithm for classification and regression tasks. It is particularly effective in solving complex problems with high-dimensional feature Spaces. Here is a systematic breakdown of the operational principles behind Support Vector Machines(SVM) in the context of algorithmic trading:

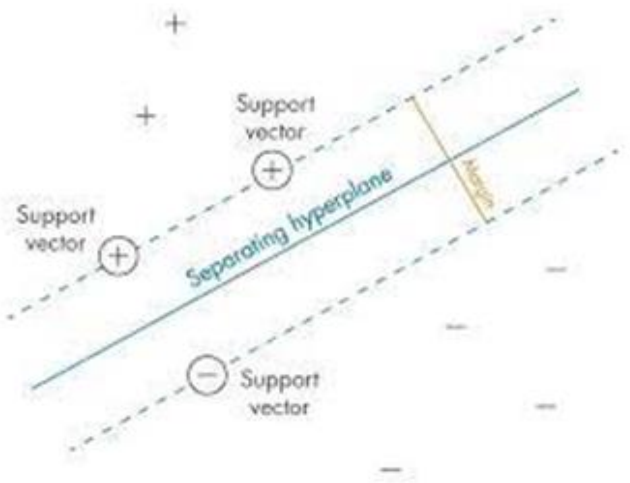

Fig.1: Support Vector Machine

1. Risk Assessment and Prediction:
SVMs can be trained to predict the likelihood of certain risk events occurring in the market. By analyzing historical data and relevant features, such as price movements, volume, technical indicators, and economic indicators, an SVM model can estimate the probability of specific risks, such as market downturns or volatility spikes. This risk assessment can provide valuable information for decision-making in algorithmic trading.

2. Decision-Making and Trade Execution:
Based on the risk assessment from the SVM model, algorithmic trading systems can make informed decisions on trade execution and risk control measures. For example, the system can adjust position sizes, set stop-loss orders, or dynamically adapt trading strategies based on the predicted risk levels. SVMs can help guide these decision-making processes by providing risk-based insights.

3. Portfolio Optimization:
SVMs can assist in optimizing portfolio allocations and risk diversification. By training SVM models on historical data, including correlations between different assets, the algorithmic trading system can use the predictions to determine the optimal allocation of assets in a portfolio. This can help manage risk by spreading investments across different assets and minimizing exposure to specific risks.

4. Anomaly Detection and Risk Mitigation:
SVMs can be utilized for anomaly detection to identify abnormal market behaviour that may pose risks to trading strategies. By training an SVM model on historical data, the system can learn patterns of normal market behaviour. When the SVM detects deviations from these patterns, it can trigger risk mitigation actions, such as suspending trading or adjusting positions, to minimize potential losses.

5. Performance Evaluation and Adaptive Learning: SVMs can be employed to evaluate the performance of trading strategies and adapt them based on risk control objectives. By analyzing the outcome of trades and comparing them to the SVM predictions, the system can assess the effectiveness of risk management measures. This evaluation can inform adjustments to the SVM model or the overall algorithmic trading strategy to improve risk control capabilities.

It's important to note that integrating SVMs into algorithmic trading requires careful consideration of data quality, model selection, feature engineering, and proper evaluation of model performance. Risk control in algorithmic trading is a dynamic process, and SVMs can serve as valuable tools within the broader framework of risk management in algorithmic trading systems.

B. Random Forests

Random Forest (RF) is a machine learning algorithm widely used in algorithmic trading for its predictive power and ability to handle complex market dynamics. In algorithmic trading, RF can be employed in the following ways:

1. Classification and Prediction:
RF can be utilized for classification tasks, such as predicting market movements or identifying trading opportunities. By training the RF model on historical data labelled with target variables (e.g., price movements), the model can learn patterns and relationships that help predict future market conditions. Traders can use these predictions to inform their trading decisions, such as determining the direction of trades or identifying potential entry and exit points.

2. Risk Management and Control:
RF plays a crucial role in risk management by providing risk assessments and aiding in risk control decisions. By incorporating relevant features such as volatility, market indicators, and asset correlations, RF can estimate the probabilities or levels of risk associated with different market scenarios. Traders can leverage these risk assessments to adjust their trading strategies, dynamically allocate portfolio weights, or implement stop-loss mechanisms to mitigate potential losses and manage risk exposure.

3. Portfolio Optimization:
RF can contribute to portfolio optimization in algorithmic trading. By training the RF model on historical market data and considering factors such as asset prices, volatilities, and correlations, the model can help determine optimal asset allocation. RF can assist traders in constructing diversified portfolios that balance risk and return, considering factors such as asset performance, risk profiles, and market conditions.

4. Feature Importance and Interpretability:
RF provides valuable insights into feature importance, highlighting the relevance of different market variables for predicting outcomes. By evaluating the impact of each feature on the model's performance, RF can assist traders in identifying key drivers of market behaviour. This information can be used to refine trading strategies, prioritize data sources, or enhance risk control mechanisms.

5. Model Validation and Ensemble Techniques:
RF is often combined with other machine learning models and techniques in algorithmic trading. Ensemble methods, such as combining RF with other algorithms like Gradient Boosting Machines or Support Vector Machines, can leverage the strengths of different models and improve overall performance. Model validation techniques, such as cross-validation or out-of-sample testing, are crucial to ensure the RF model's reliability and effectiveness in real-world trading scenarios.

6. Real-Time Adaptation:
RF models can be updated and adapted in real-time to account for changing market conditions. Traders can retrain the RF model periodically or incorporate new data to capture evolving market dynamics and improve the accuracy of predictions. Real-time adaptation ensures that the model remains relevant and effective in dynamic trading environments.

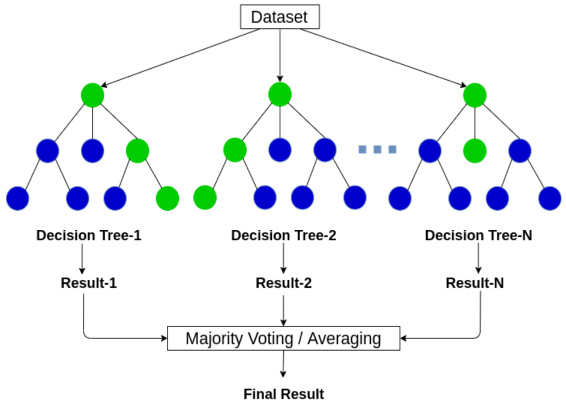

Fig.2: Random Forest Classifier

It's important to note that while RF is a powerful tool for algorithmic trading, it should be used alongside other risk management techniques, market insights, and human judgment. RF models are based on historical data and patterns, and unforeseen events or structural changes in the market can introduce risks that may not be fully captured by the model alone. A comprehensive approach that combines various strategies, risk control mechanisms, and market expertise is essential for successful algorithmic trading.

C. Gradient Boosting Models (GBMs)

Gradient Boosting Machines (GBMs) work in algorithmic trading for risk control by iteratively combining weak learners (decision trees) to create a powerful predictive model. Here is a systematic breakdown of the operational principles behind Gradient Boosting Machines (GBMs) in the context of algorithmic trading:

1. Data Collection: Historical market data is collected, including relevant features such as price, volume, technical indicators, and economic data.

2. Data Pre-processing: The collected data is pre-processed to handle missing values, normalize or standardize the features, and split it into training and testing sets.

3. Labelling: The historical data is labelled with appropriate risk levels or outcomes to train the GBM model. For example, the data could be labelled with risk categories such as low risk, medium risk, and high risk based on historical price movements, volatility, or other risk indicators.

4. Training the GBM: The GBM model is trained using the labelled historical data. The process starts with the creation of an initial weak learner, which is typically a shallow decision tree. This tree is fitted to the training data, attempting to predict the risk levels based on the input features.

5. Iterative Training: In subsequent iterations, new weak learners (decision trees) are added to the GBM. Each new tree is trained to correct the errors made by the ensemble of trees constructed so far. The model focuses on the data instances where the previous trees performed poorly and assigns higher weights to those instances to improve the model's performance.

6. Gradient Descent Optimization: During each iteration, the GBM optimizes a loss function by applying gradient descent. The loss function measures the discrepancy between the predicted risk levels and the actual labels. The GBM adjusts the model's

parameters to minimize the loss function, moving in the direction of steepest descent.

7. Ensemble Prediction: The final prediction of the GBM is the sum of the predictions from all the weak learners. Each weak learner contributes to the prediction based on its individual weight. The combination of multiple weak learners enables the GBM to capture complex relationships in the data and make accurate risk predictions.

8. Risk Assessment and Prediction: Once the GBM model is trained, it can be used to assess and predict risks in real-time market conditions. By feeding new market data into the model, it estimates the probabilities or levels of risk associated with different scenarios. These risk assessments guide risk control decisions in algorithmic trading, such as adjusting position sizes, setting stop-loss levels, or dynamically adapting trading strategies based on the predicted risks.

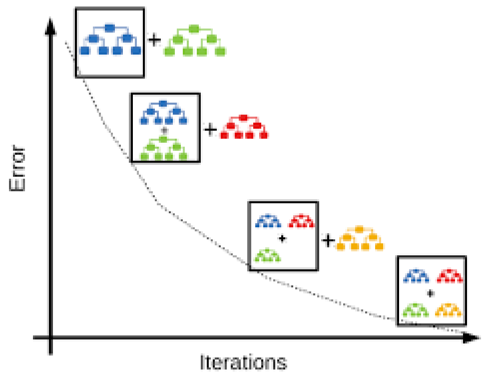

Fig.3: Gradient Boosting Models

GBMs excel at capturing nonlinear relationships and handling large feature spaces. They are highly flexible and can adapt to different types of risk control tasks in algorithmic trading. However, it's important to consider model evaluation, hyperparameter tuning, and the ongoing monitoring and adaptation of the GBM model to ensure its performance and reliability in real-world trading scenarios.

D. Recurrent Neural Networks (RNNs)

RNN stands for Recurrent Neural Network. It is a type of artificial neural network that is designed to process sequential data or data with temporal dependencies. RNNs are widely used in various domains, including natural language processing, speech recognition, and time series analysis, making them relevant to algorithmic trading. The basic architecture of an RNN consists of recurrent units that take an input at each time step and produce an output and hidden state. The hidden state is then fed back into the network as input for the next time step, creating a feedback loop. This recurrent structure allows the network to remember information from previous time steps and utilize it for subsequent predictions. In the context of algorithmic trading, RNNs, especially LSTM and GRU, are commonly used to model and predict financial time series data. They can learn complex patterns and relationships from historical price data, technical indicators, and other relevant factors, enabling traders to make predictions about future price movements or identify trading opportunities.

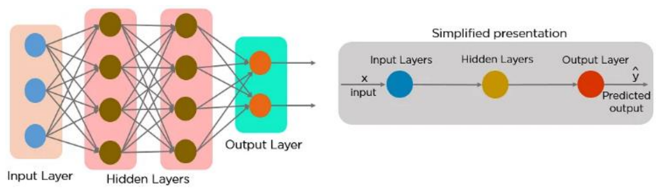

Fig.4: Recurrent Neural Network

Here is a systematic breakdown of the operational principles behind Recurrent Neural Network (RNN) in the context of algorithmic trading:

1. Data Collection:
Historical market data is collected, including price, volume, technical indicators, and economic data, specific to the assets or markets of interest. Data sources can include market feeds, financial databases, or specialized data providers.

2. Data Pre-processing:
The collected data undergoes pre-processing steps tailored to algorithmic trading. This includes handling missing values, outlier detection and treatment, normalizing or standardizing the features, and splitting the data into training, validation, and testing sets. Special attention is given to maintaining the sequential structure of the data.
3. Architecture of RNN:
The RNN architecture comprises recurrent units that enable information to flow and persist across different time steps. In algorithmic trading, popular RNN variants like Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRU) are commonly used due to their ability to capture long-term dependencies and mitigate the vanishing gradient problem.

4. Training the RNN Model:
The RNN model is trained using historical data to learn the patterns and relationships within the data. The sequential nature of the data is leveraged to capture temporal dependencies and exploit trends, seasonality, and other time-related patterns in the financial data.

5. Risk Assessment and Prediction:
Once the RNN model is trained, it can be used for risk assessment and prediction in real-time market conditions. By feeding new market data into the trained RNN, it generates predictions about future risk levels or outcomes. These predictions can guide risk control decisions, such as adjusting position sizes, setting stop-loss levels, or dynamically adapting trading strategies based on the predicted risks.

6. Sequential Pattern Recognition:
RNNs excel at recognizing and exploiting sequential patterns in financial time series data. They can identify short-term and long-term trends, detect market regimes, and capture other relevant temporal relationships, all of which are critical for risk control in algorithmic trading.

7. Signal Generation and Trading Strategy Execution:
RNNs can be used to generate trading signals based on risk predictions. These signals can trigger buy/sell decisions or modify existing positions. They can also be combined with other models or trading strategies to form a comprehensive trading system. Execution algorithms and risk management techniques are applied to ensure proper implementation of the trading strategies.

8. Model Evaluation and Validation:
The trained RNN model is evaluated and validated to assess its performance and robustness. Evaluation involves metrics specific to algorithmic trading, such as profitability, risk-adjusted returns, and statistical measures of performance (e.g., Sharpe ratio, maximum drawdown). The model is rigorously tested using out-of-sample data to evaluate its generalization ability and to avoid overfitting.

9. Real-Time Adaptation and Monitoring:
RNN models in algorithmic trading require ongoing monitoring and adaptation to changing market conditions. This involves retraining the model periodically with new data to capture evolving market dynamics. Hyperparameter optimization, regularization techniques, and feature selection methods are employed to fine-tune the model's performance.

10. Risk Management and Control:
RNNs play a vital role in risk management by providing timely risk assessments and predictions. These inform risk control decisions, including portfolio allocation, position sizing, stop-loss levels, and the selection of risk mitigation strategies. Risk management algorithms are employed to ensure that risk exposure remains within predefined limits.

RNNs offer the ability to capture complex temporal dependencies in financial data, making them well-suited for risk control in algorithmic trading. However, it's important to consider the limitations of RNNs, such as sensitivity to hyperparameters and potential difficulties in training with long historical sequences.

E. Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) is a variant of Recurrent Neural Networks (RNNs) that has proven effective in risk control in algorithmic trading. LSTMs excel at capturing long-term dependencies in sequential data, making them well-suited for analyzing financial time series data. Here's a more detailed explanation of how LSTM works specifically for risk control in algorithmic trading, incorporating additional algorithmic trading-based information:

1. Data Collection: Historical market data, including price, volume, technical indicators, and economic data, is collected for training the LSTM model. The data is typically organized as a time series with sequential data points.

2. Data Pre-processing: The collected data undergoes pre-processing steps tailored to algorithmic trading. This includes handling missing values, outlier detection and treatment, normalization or standardization of features, and splitting the data into training, validation, and testing sets. The sequential structure of the data is preserved during pre-processing.

3. Architecture of LSTM: The LSTM architecture comprises memory cells that allow information to flow across different time steps while maintaining a long-term memory. LSTMs contain various gates, such as the input gate, forget gate, and output gate, which control the flow of information and enable the network to selectively retain or discard information over time.

4. Training the LSTM Model: The LSTM model is trained using the historical data to learn the patterns and relationships within the data. The sequential nature of the data allows the LSTM to capture temporal dependencies and exploit trends, seasonality, and other time-related patterns in the financial data.

5. Risk Assessment and Prediction: Once the LSTM model is trained, it can be used for risk assessment and prediction in real-time market conditions. By feeding new market data into the trained LSTM, it generates predictions about future risk levels or outcomes. These predictions guide risk control decisions, such as adjusting position sizes, setting stop-loss levels, or dynamically adapting trading strategies based on the predicted risks.

6. Capturing Long-Term Dependencies: LSTMs are particularly effective in capturing long-term dependencies in financial time series data. They can recognize and exploit patterns that extend over a significant number of time steps, such as trend reversals, cyclic behaviour, or macroeconomic factors influencing the market.

7. Signal Generation and Trading Strategy Execution: LSTMs can be used to generate trading signals based on risk predictions. These signals can trigger buy/sell decisions or modify existing positions. They can also be combined with other models or trading strategies to form a comprehensive trading system. Execution algorithms and risk management techniques are applied to ensure proper implementation of the trading strategies.

8. Model Evaluation and Validation: The trained LSTM model is evaluated and validated using metrics specific to algorithmic trading, such as profitability, risk-adjusted returns, and statistical measures of performance (e.g., Sharpe ratio, maximum drawdown). Robust validation techniques, including walk-forward analysis and out-of-sample testing, are employed to assess the model's generalization ability and avoid overfitting.

9. Real-Time Adaptation and Monitoring: LSTM models in algorithmic trading require continuous monitoring and adaptation to changing market conditions. This involves retraining the model periodically with new data to capture evolving market dynamics. Hyperparameter optimization, regularization techniques, and feature selection methods are employed to fine-tune the model's performance.

10. Risk Management and Control: LSTMs play a crucial role in risk management by providing timely risk assessments and predictions. These inform risk control decisions, including portfolio allocation, position sizing, stop-loss levels, and the selection of risk mitigation strategies. Risk management algorithms are employed to ensure that risk exposure remains within predefined limits.

LSTMs offer powerful capabilities for risk control in algorithmic trading by effectively capturing long-term dependencies and patterns in financial time series data.

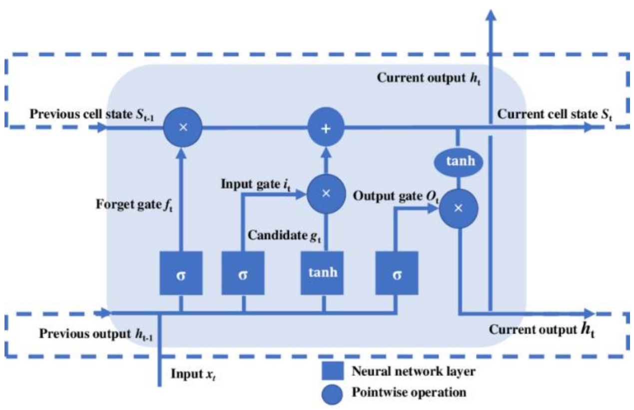

Fig.5 : Long Short-Term Memory (LSTM) Networks

However, it's important to consider the challenges associated with LSTMs, such as the potential for overfitting, the need for careful hyperparameter tuning, and the management of sequential data pre-processing and normalization. Robust risk control in algorithmic trading typically involves combining multiple models and techniques, where LSTMs can serve as a valuable component within a comprehensive trading framework.

F. Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) are a class of machine learning models that consist of two components: a generator and a discriminator. GANs have been explored for risk control in algorithmic trading to generate synthetic data that mimics the behaviour of real financial data.

Here is a systematic breakdown of the operational principles behind Generative Adversarial Networks(GANs) in the context of algorithmic trading:

1. Data Collection and Pre-processing:
Algorithmic trading requires collecting a vast amount of historical market data. This data includes not only price and volume information but also various technical indicators, fundamental data, news sentiment, and other relevant factors. The collected data is pre-processed to handle missing values, outliers, and ensure consistency in data formats.

2. GAN Architecture Selection:
Different GAN architectures can be explored based on the specific requirements of algorithmic trading. Variants such as Deep Convolutional GANs (DCGANs) or Conditional GANs (CGANs) can be considered depending on the type of data and desired outputs.

3. Generator Network:
The generator network in the GAN model learns to generate synthetic financial data that resembles the real market data. It takes random noise as input and produces synthetic samples as output. The generator network's architecture can be designed to capture the temporal dependencies, patterns, and statistical properties of the financial time series data.

4. Discriminator Network:
The discriminator network is trained simultaneously with the generator network. It learns to distinguish between real and synthetic data. The discriminator helps guide the generator network to generate more realistic and accurate synthetic data by providing feedback on the quality of the generated samples.

5. Training Process:
The GAN model is trained iteratively through a competitive process. The generator network generates synthetic data, and the discriminator network evaluates the authenticity of both real and synthetic samples. The networks are updated based on the discriminator's feedback to improve the generator's ability to produce data that closely resembles the real financial data.

6. Risk Assessment and Synthetic Data Generation:
The generator network can generate synthetic financial data that captures the statistical properties, temporal dependencies, and patterns observed in the real data. This synthetic data can be used for various risk assessment purposes, such as backtesting trading strategies, assessing portfolio risk, or evaluating the impact of different market scenarios.

7. Model Evaluation and Validation:
The quality and effectiveness of the generated synthetic data are evaluated using a combination of statistical measures and domain-specific metrics. The synthetic data is compared with the real data to assess its similarity, statistical properties, and the ability to capture the underlying market dynamics accurately. The performance of risk models trained on the augmented dataset is also evaluated to ensure its reliability and generalization capabilities.

8. Risk Control Strategies:
The insights gained from the synthetic data generation and risk modelling can be utilized in various risk control strategies. The augmented dataset can be used to refine and enhance risk models, allowing for better risk estimation and mitigation. The synthetic data can also be incorporated into risk management algorithms to dynamically adjust trading positions, implement stop-loss mechanisms, or optimize portfolio allocation based on anticipated risks.

9. Adaptive Learning and Dynamic Updates:
GANs can facilitate adaptive learning and dynamic updates in algorithmic trading systems. The GAN model can be continually retrained with new market data to generate more accurate synthetic data and adapt to changing market conditions. This enables risk control measures to evolve in real-time and ensures that the system remains responsive to emerging risks and opportunities.

10. Combination with Other Techniques:
GANs can be combined with other machine learning techniques, such as LSTM or SVM, to enhance risk control in algorithmic trading. For example, GAN-generated synthetic data can be used as additional input for LSTM models to improve their predictive capabilities. The combination of multiple techniques allows for a more comprehensive and robust risk control framework.

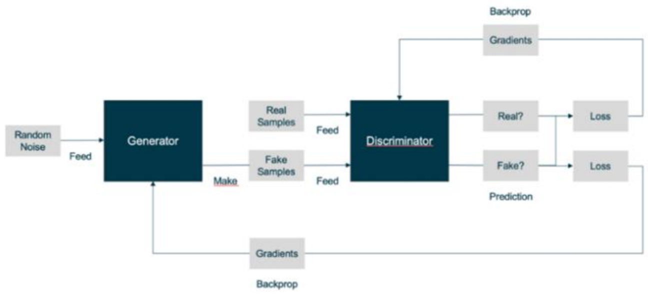

Fig.6 : Generative Adversarial Network

It's important to note that while GANs have shown promise in generating synthetic financial data, careful validation and testing are essential to ensure that the generated data accurately reflects the characteristics and behaviour of the real market.

Additionally, risk control strategies based on GAN-generated data should be thoroughly back tested and validated before deployment in live trading environments.

VI. PERFORMANCE EVALUATION AND CHALLENGES

A. Backtesting and Simulation

Backtesting and simulation are crucial components of risk control in algorithmic trading. They involve the historical simulation of trading strategies using past market data to assess their performance, evaluate risk metrics, and validate the effectiveness of risk control measures. Here's an overview of backtesting and simulation in the context of risk control in algorithmic trading:

1. Data Selection and Pre-processing:
Historical market data is selected and collected, including price, volume, and other relevant data points. The data is pre-processed to handle missing values, adjust for corporate actions (e.g., stock splits), and ensure consistency in the data format.

2. Strategy Development:
Trading strategies, including risk control measures, are developed based on specific objectives and risk tolerance. These strategies can include entry and exit signals, position sizing rules, stop-loss levels, risk mitigation techniques, and other risk management parameters.

3. Simulation Environment:
A simulation environment is created to replicate real market conditions. This environment incorporates the selected historical data, accounting for transaction costs (such as commissions and slippage), market liquidity, and other trading constraints.

4. Execution and Performance Measurement:
The trading strategy is executed in the simulation environment using historical data. The performance of the strategy is measured in terms of key metrics, including returns, risk-adjusted returns (such as Sharpe ratio), maximum drawdown, win/loss ratio, and other relevant performance indicators.

5. Risk Assessment and Analysis:
Risk metrics, such as Value at Risk (VaR), Expected Shortfall (ES), or downside risk measures, are calculated to evaluate the potential losses and downside risk associated with the trading strategy. The risk control measures implemented in the strategy are assessed for their effectiveness in mitigating risk.

6. Parameter Optimization:
The parameters of the trading strategy and risk control measures are optimized through parameter sweeps or optimization algorithms. This process aims to find the combination of parameters that maximizes returns while minimizing risks within the defined risk control framework.

7. Sensitivity Analysis:
Sensitivity analysis is performed to assess the robustness of the trading strategy and risk control measures to changes in key assumptions or market conditions. This analysis helps identify potential vulnerabilities and provides insights into the strategy's performance under different scenarios.

8. Out-of-Sample Testing:
To validate the strategy's performance and robustness, out-of-sample testing is conducted using a separate set of historical data that was not used in the initial development or parameter optimization. This testing ensures that the strategy can generalize well to unseen market conditions.

9. Risk Control Adjustments:
Based on the results of backtesting and simulation, risk control measures and parameters may be adjusted to enhance the strategy's risk-adjusted performance. This iterative process allows for continuous improvement and refinement of risk control mechanisms.

10. Monitoring and Adaptation:
Risk control in algorithmic trading is an ongoing process. After deployment, the trading strategy and risk control measures are continually monitored and adapted to changing market conditions. Regular updates to the strategy may be required to maintain its effectiveness in managing risk.

Backtesting and simulation provide valuable insights into the performance and risk characteristics of algorithmic trading strategies. They help traders and investors evaluate the potential risks, refine risk control mechanisms, and make informed decisions to optimize their trading strategies.

B. Overfitting and Generalization

Performance evaluation through backtesting and simulation is a critical step in assessing the effectiveness of risk control in algorithmic trading. It involves measuring the performance of trading strategies under historical market conditions to understand their risk-adjusted returns, profitability, and risk management capabilities. Here are key aspects of performance evaluation in algorithmic trading risk control:

1. Metrics for Performance Evaluation:
Various metrics are used to evaluate the performance of trading strategies. Common metrics include:
- Returns: Calculate the overall profitability of the strategy, such as total return, annualized return, or compounded return.
- Risk-Adjusted Returns: Measure the returns relative to the risk taken by considering metrics like Sharpe ratio, Sorting ratio, or information ratio.
- Drawdowns: Assess the magnitude and duration of peak-to-trough declines in the strategy's equity curve, reflecting potential losses.
- Win/Loss Ratio: Determine the ratio of winning trades to losing trades, indicating the strategy's ability to generate profitable trades.
- Risk Measures: Utilize risk metrics like Value at Risk (VaR), Expected Shortfall (ES), or maximum portfolio loss to quantify downside risk.

2. Backtesting Process: Backtesting involves simulating the trading strategy using historical data to assess its performance. The process typically includes the following steps:
- Data Selection: Choose relevant historical market data, including price, volume, and other essential indicators required for the strategy.
- Data Pre-processing: Cleanse and adjust the data for splits, dividends, and other events that may affect the analysis.
- Strategy Implementation: Execute the trading strategy using predetermined rules, such as entry and exit signals, position sizing, and risk management parameters.
-

Transaction Costs: Incorporate realistic transaction costs like commissions, slippage, and spread to accurately reflect real-world trading conditions.
- Portfolio Management: Simulate the management of multiple assets or positions, including rebalancing, portfolio diversification, and position sizing strategies.
- Performance Measurement: Calculate performance metrics, track equity curves, and evaluate risk management measures over the backtesting period.

3. Out-of-Sample Testing: To validate the strategy's performance and robustness, it's important to conduct out-of-sample testing. This involves using a separate

set of historical data that was not used in the initial development or parameter optimization of the strategy. Out-of-sample testing helps assess the strategy's ability to generalize to unseen market conditions and verify its effectiveness beyond the backtesting period.

4. Statistical Significance: When evaluating performance, it's essential to determine whether the observed results are statistically significant or simply due to chance. Statistical tests, such as t-tests or bootstrapping, can help assess the significance of performance metrics and validate the strategy's performance.

5. Risk Assessment and Management: Backtesting and simulation provide insights into the risk management capabilities of the trading strategy. By analyzing risk metrics, drawdowns, and risk-adjusted returns, traders can assess whether the strategy effectively controls risk and aligns with their risk tolerance.

6. Benchmarking: Comparing the strategy's performance against relevant benchmarks, such as market indices or peer strategies, helps provide context and determine if the strategy outperforms or underperforms the broader market or similar trading approaches.

7. Iterative Improvement: Backtesting and simulation are not one-time activities. As new data becomes available, traders can refine and enhance their strategies based on performance evaluation results. This iterative process allows for continuous improvement and adaptation to changing market dynamics.

Remember, while backtesting and simulation provide valuable insights, they have limitations. The results are based on historical data and assumptions, and past performance does not guarantee future results. It's important to consider factors such as market conditions, transaction costs, and slippage when interpreting the performance of a strategy and managing risks in algorithmic trading.

VII. CASE STUDIES AND APPLICATION

A. Stock Market Prediction and Trading

Stock market prediction and trading is a prominent application of machine learning algorithms in algorithmic trading. By utilizing historical price data, trading volumes, and other relevant indicators, machine learning models can be trained to predict stock prices and generate trading signals. These predictions can assist traders in making informed decisions regarding buying, selling, or holding stocks. Additionally, machine learning techniques can be used to identify patterns, trends, and anomalies in the stock market, enabling the development of sophisticated trading strategies.

B. Cryptocurrency Trading

Cryptocurrency trading has gained significant attention in recent years, and machine learning algorithms have found applications in this domain as well. The high volatility and complex dynamics of cryptocurrency markets make them suitable for machine learning-based trading strategies. Machine learning models can analyze historical cryptocurrency price data, trading volumes, sentiment analysis from social media, and other relevant factors to predict future price movements and identify profitable trading opportunities. These models can assist traders in optimizing their cryptocurrency trading strategies and managing risks associated with this emerging asset class.

C. Foreign Exchange (Forex) Trading

Foreign exchange (Forex) trading involves the buying and selling of different currencies. Machine learning algorithms have been widely employed in Forex trading due to the vast amount of historical data available. These algorithms can analyze currency exchange rates, economic indicators, geopolitical events, and other factors to predict short-term or long-term movements in currency pairs. By incorporating machine learning models into Forex trading strategies, traders can make more accurate predictions, develop robust risk management techniques, and potentially increase their profitability in the highly liquid Forex markets.

These case studies demonstrate the diverse applications of machine learning algorithms in algorithmic trading. From stock market prediction and cryptocurrency trading to Forex trading, machine learning techniques offer valuable insights and tools to traders seeking to capitalize on market opportunities and effectively manage risks. As the field of algorithmic trading continues to evolve, machine learning algorithms are expected to play an increasingly important role in developing innovative trading strategies and enhancing overall trading performance.

VIII. LIMITATIONS

While machine learning algorithms have shown great promise in risk control for algorithmic trading, they also come with certain limitations that need to be considered:

A. Data quality and reliability:

Machine learning algorithms heavily rely on the quality and reliability of the data used for training. If the input data contains errors, outliers, or biases, it can adversely impact the performance and accuracy of the algorithms. Therefore, careful data pre-processing and validation are essential to ensure the reliability of the results.

B. Changing market dynamics:

Financial markets are dynamic and subject to evolving conditions. Machine learning algorithms trained on historical data may struggle to adapt to rapidly changing market dynamics. Ongoing monitoring and updating of the algorithms are necessary to ensure their effectiveness and relevance in different market environments.

C. Model risk and limitations:

Machine learning algorithms are not infallible and can still be subject to modeling errors and limitations. Traders need to be aware of the assumptions, biases, and potential weaknesses of the models they employ. It is essential to perform thorough validation, stress testing, and sensitivity analysis to assess the robustness and limitations of the algorithms.

IX. CONCLUSION

In conclusion, this survey paper has provided a comprehensive overview of machine learning algorithms for risk-controlled algorithmic trading. The importance of risk control in algorithmic trading has been highlighted, emphasizing the need for robust techniques to manage and mitigate risks in the financial markets.

The paper explored various machine learning algorithms applicable to algorithmic trading, including supervised learning algorithms such as Support Vector Machines and Reinforcement Learning algorithms like Deep Learning. Each algorithm's strengths, limitations, and applications in algorithmic trading were discussed, providing insights into their potential use cases.

Additionally, the paper covered essential aspects of algorithmic trading, such as data pre-processing, feature engineering, evaluation metrics for trading strategies, risk management techniques, and performance evaluation through backtesting and simulation. These topics provided a holistic view of the key components involved in developing and assessing algorithmic trading strategies.

Throughout the paper, the challenges associated with risk-controlled algorithmic trading and the application of machine learning algorithms were addressed. These challenges included data quality, overfitting, assumptions, and the need for forward testing to validate the performance of strategies in real-time market conditions.

By gaining a deeper understanding of machine learning algorithms and risk management techniques in algorithmic trading, traders and researchers can make informed decisions and develop robust strategies that enhance profitability while effectively managing risks.

Overall, this survey paper serves as a valuable resource for individuals interested in the field of algorithmic trading and machine learning. It provides a foundation of knowledge, highlights the key concepts, and challenges, and offers insights into the application of machine learning algorithms for risk-controlled algorithmic trading. With the continuous advancements in machine learning and the ever-evolving financial markets, this paper sets the stage for further research and innovation in the field.

X. REFERENCES

[1]. S. Carta, D. R. Recupero, R. Saia, and M. M. Stanciu, "A general approach for risk-controlled trading based on machine learning and statistical arbitrage,'' in Proc. 6th Int. Conf. Mach. Learn., Optim., Data Science (LOD) in Lecture Notes in Computer Science, vol. 12565, 2020.

[2]. J. Jayko, 'A demon of our own design: Markets, hedge funds, and the perils of financial innovation,' J. Pension Econ. Finance, vol. 7, no. 3, p. 363, 2008. G. Vidyamurthy, Pairs Trading: Quantitative Methods and Analysis. Hoboken, NJ, USA: Wiley, 2004.

[3]. M. T. Leung, H. Daouk, and A.-S. Chen, "Forecasting stock indices: A comparison of classification and level estimation models,'' Int. J. Forecasting, vol. 16, no. 2, pp. 173–190, 2000.

[4]. G. S. Atsalakis and K. P. Valavanis, 'Surveying stock market forecasting techniques—Part II: Soft computing methods,' Expert Syst. Appl., vol. 36, no. 3, pp. 5932–5941, Apr. 2009.

[5]. N. Huck, "Large data sets and machine learning: Applications to statistical arbitrage,' Eur. J. Oper. Res., vol. 278, no. 1, pp. 330–342, Oct. 2019

[6]. R. Almeida, L. F. Brandão, and D. G. da Silva, "Machine learning for high-frequency trading: a practical approach." In Proceedings of the 20th European Conference on Artificial Intelligence, 2012.

[7]. J. Moody and M. Saffell, "Learning to trade." Technical Report, The University of Michigan, 2000.

[8]. H. Sargolzaei, R. A. Javidan, and A. Poon, "Machine learning for algorithmic trading: predicting and optimizing execution strategies." In Proceedings of the 2015 International Conference on Machine Learning and Applications (ICMLA), 2015.

[9]. Y. Sun, D. Liu, J. Chen, and Y. Wang, "An empirical study of machine learning techniques for risk control in algorithmic trading." In Proceedings of the 2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA), 2017.

[10]. X. Zhang, Y. Yu, and Y. Tang, "A comprehensive review of machine learning techniques for algorithmic trading strategies." Artificial Intelligence Review, vol. 52, no. 4, pp. 2275-2295, 2019.

[11].H. Chen, T. Zheng, and Y. Zhang, "Machine learning techniques for risk control in high-frequency trading." In Proceedings of the 2016 International Conference on Progress in Informatics and Computing (PIC), 2016.

[12].Y. Hong, X. Li, and B. Zhang, "Risk control in algorithmic trading: a survey on performance evaluation, risk assessment, and regulation." Journal of Systems Science and Complexity, vol. 34, no. 1, pp. 1-19, 2021.

[13].L. Wu, Y. Zhang, J. Shang, and Q. Zhang, "Risk control of algorithmic trading based on deep reinforcement learning." In Proceedings of the 2020 International Conference on Intelligent Transportation, Big Data and Smart City (ICITBS), 2020.

Cite this article as :

Soham Pathak, Antara Pawar, Shruti Taware, Sarthak Kulkarni, Afsha Akkalkot, "A Survey on Machine Learning Algorithms for Risk-Controlled Algorithmic Trading", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 10 Issue 3, pp. 1069-1089, May-June 2023. Available at doi : https://doi.org/10.32628/IJSRST523103163
Journal URL : https://ijsrst.com/IJSRST523103163