ASM Sc. J., 21(1), 2026

https://doi.org/10.32802/asmscj.2026.0077

Designing a Cutting-Edge User-Centric Algorithmic Trading Platform with QuantConnect and Deep Learning

C.F.Tan[1], X.Y.Chew[1*], K.W.Khaw[2], W.L.Teoh[3], and Z.L.Chong[4]

[1]*School of Computer Science, Universiti Sains Malaysia, 11800 Gelugor, Penang, Malaysia*

[2]*School of Management, Universiti Sains Malaysia, 11800 Gelugor, Penang, Malaysia*

[3]*School of Mathematical and Computer Science, Heriot-Watt University, Malaysia*

[4]*Department of Electronic Engineering, Faculty of Engineering & Green Technology, Universiti Tunku Abdul Rahman, Jalan Universiti, Bandar Barat, 31900 Kampar, Perak, Malaysia.*

Rapid advances in big data analytics, machine learning (ML), and deep learning (DL) have transformed algorithmic trading (AT), enabling more accurate prediction of market trends and price movements. Once limited by technological constraints, AT is now widely adopted, with a large share of daily trading volume executed by hedge funds, insurance companies, and investment banks using proprietary algorithms. This institutional dominance has concentrated resources and innovation within large corporations, creating barriers that limit access to advanced trading tools for individual investors. This paper addresses this imbalance by proposing a user-centric, open-source algorithmic trading platform designed to democratise access to sophisticated trading technologies. The system integrates a custom-built deep learning model that analyses historical stock data alongside news sentiment to improve trading predictions. The trading infrastructure is implemented using the open-source QuantConnect framework, while decision-making is driven by DL and sentiment analysis techniques. The application was designed, deployed, and rigorously evaluated using standard software performance metrics. Results demonstrate that the proposed system enhances accessibility to state-of-the-art trading strategies, empowering individual investors and promoting a more inclusive, competitive, and innovative stock trading ecosystem.

Keywords: deep learning; investors; open-source; QuantConnect; stock analysis

I. INTRODUCTION

Algorithmic Trading (AT) has emerged as a prominent trend in today's dynamic stock market environment. Essentially, it involves the utilisation of sophisticated computer algorithms to autonomously execute various stages of the trading cycle, encompassing pre-trade analysis, trading signal generation, and trade execution (Banerjee & Banerjee, 2017; Mathur *et al.*, 2021; Sethi, 2015). Implementing AT offers significant advantages, unlocking new potential in trading strategies (Jain *et al.*, 2023). However, the complexity involved in executing such algorithms, particularly those leveraging Machine Learning (ML) and Deep Learning (DL) techniques, often limits their accessibility to institutional traders, such as hedge funds, insurance firms, and investment banks. Consequently, retail traders may need more exposure to this approach. They may even be unaware of its existence, especially as new financial instruments like ETFs, options, and leveraged systems (e.g., forex) are introduced (Sezer & Ozbayoglu, 2018).

The integration of advanced AI in finance continues to accelerate, with large language models (LLMs) and more sophisticated deep learning architectures emerging as key tools for market analysis (Brown & Yang, 2024; Garcia *et al.*, 2025). However, access to these technologies remains a

*Corresponding author's e-mail: xinying@usm.my

challenge for retail investors, reinforcing the need for open-source, user-centric platforms.

Despite its predominance among institutional traders, AT has witnessed a surge in demand from the retail sector. Li *et al.* (2019) underscored longstanding challenges in traditional trading methods, including the difficulty in extracting effective market representations and the disparity between classification (predicting market direction) and directly learning trading strategies. In addition, AT currently needs more resources and innovation within the industry. Stock trading has traditionally been characterised as a zero-sum game, where one party's gain equates to another's loss. Consequently, individuals who develop successful algorithms often opt to keep them proprietary or offer them at a premium price, limiting access to innovative strategies.

Essentially, having an AT model is insufficient; investors also require a trading platform capable of deploying such models effectively, enabling practical engagement in real-life scenarios. Effective AT deployment becomes especially pertinent amid the surge in popularity of investing and trading during the COVID-19 pandemic era and beyond (Chakrabarty & Pascual, 2023; Nakari, 2023). Much like proprietary AT solutions, proprietary trading platforms present challenges similar to those faced by individual investors. Over the years, the solution to proprietary software platforms has been the adoption of open-source alternatives. Open-source technology (OST) offers numerous advantages over proprietary solutions, making it a compelling choice for investors and traders alike. Among its many benefits are transparency and collaboration (openly accessible to the public for scrutiny, modification, and contribution). The advantage point of flexibility and customisation is also a gain for OST. Unlike proprietary platforms that may restrict users to predefined functionalities, open-source solutions empower users to tailor the software to suit their specific needs and preferences. This adaptability ensures that traders can efficiently implement their trading strategies using ML models and adapt to changing market conditions. However, potential challenges of open-source implementation include 1) lack of dedicated customer support, 2) complexity of deployment and operation, 3) security risks in the face of cyberattacks, and 4) computational requirements, which need to be critically addressed for an effective and efficient open-source trading application.

This paper, therefore, presents a QuantConnect-based trading application as a versatile solution tailored for implementing customised AT models. The proposed approach involves two key phases: firstly, customising and designing a web application using QuantConnect (Board, 2018), and secondly, integrating ML models into the web application. The ML models incorporated in this framework include Neural Network (NN) and MultiLayer Perceptron (MLP) algorithms for generating trading signal predictions, alongside the Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM), and CNN-BiLSTM for sentiment analysis of stock market news. The contributions of this paper are:

- Developing a user-centric trading platform based on the QuantConnect open-source solution.
- Utilising the QuantConnect-based application for training and deploying public ML and DL trading algorithms, with subsequent performance evaluation.

This approach prioritises transparency and offers flexibility and cost-effectiveness for individual investors. Moreover, it facilitates near-real-life engagement for both novice and seasoned investors, enabling them to execute trading activities, monitor market fluctuations, and implement AT models with ease.

II. LITERATURE REVIEW

As AT reaches a significant developmental milestone, it now boasts a comprehensive array of core functions, empowering users to effortlessly explore diverse algorithmic strategies tailored to their unique preferences and investment objectives (Taye, 2021). Despite the prevalence of AT platforms, there remains a need for more options available in the market.

One notable platform in this domain is Quanto, renowned for its cloud-based infrastructure and extensive suite of tools and resources. Users leverage Quanto to code, back-test, and deploy their algorithms, all within a unified environment. Furthermore, the platform offers access to both real-time and historical market data, providing users with invaluable inputs for refining their strategies. This comprehensive feature set positioned Quanto as a leading choice among trading

platforms (Maheshwari, 2020; Spörer, 2020; Taye, 2021). In a study by Chen, a sentiment analysis-based trading strategy was presented, leveraging the Quanto platform (Z. (Neil) Chen, 2023). This algorithm employed a rule-based dictionary approach to gauge market sentiment towards various stocks, using the insight weight method to construct portfolios based on sentiment analysis results. Back testing against historical data and optimisation efforts led to enhanced performance metrics. The results demonstrated that the proposed strategy exhibited comparable performance to the S&P 500 index across both bullish and bearish market conditions, with the potential for outperformance in favourable market environments. One of the drawbacks of Quanto is that its design was tailored more toward experienced traders with coding and programming skills than casual traders. This limitation significantly restricts the platform's potential and accessibility to the wider public, which this paper addresses.

While Quanto caters to retail traders with coding skills, Wyden, formerly known as AlgoTrader, targets institutional clients such as investment firms, hedge funds, and insurance companies (Taye, 2021). Wyden is a Java-based trading software deployed through Docker that offers cloud-based access via web browsers. It boasts a strong client base, including prominent companies like Coinbase, Binance, and Bloomberg. However, Wyden's accessibility to retail traders is severely limited, and it lacks explicit support for ML and DL models compared to Quanto. On the other hand, Streak is a mobile application exclusively available on the Android OS and is designed for retail trading. It stands out for its ability to conduct back testing during deployment and features like the scanner, allowing users to preset specific criteria. Streak is beginner-friendly, but its drawback lies in its limited trading strategies, which often need to be updated. Other notable trading platforms include TradeStation, PyAlgoTrade, Ninja Trader, and AmiBroker (Taye, 2021).

In the context of AT, various ML models, including cutting-edge DL techniques, have been proposed in the literature (Pothumsetty, 2020). For instance, Chen *et al.* (2019) investigated China's Shanghai and Shenzhen 300 stock index futures and developed a prediction model using CNN and Deep Belief Network (DBN). Findings show that combining CNN with DBN yielded higher accuracy than using CNN alone. Similarly, Xie *et al.* (2020) constructed a forecasting model to predict short-term stock indexes for future price movements. The proposed DL models demonstrate improved prediction accuracy over traditional methods. Hossain *et al.* (2018) introduced a stock price prediction model based on a hybrid deep recurrent neural network integrating LSTM and Gated Recurrent Unit (GRU).

Li *et al.* (2019) devised a transaction agent leveraging deep reinforcement learning, outperforming benchmark approaches. Liu *et al.* (2017) proposed a hybrid CNN-LSTM neural network model for quantitative stock market analysis, achieving superior returns compared to benchmark strategies. A recent survey by Smith & Jones (2024) categorises and evaluates over 50 deep learning-based trading models, highlighting the superior performance of hybrid architectures like CNN-LSTM and Transformer-based models in capturing complex market patterns. Jeong and Kim devised a trading system integrating Reinforcement Learning (RL) and Deep Neural Networks (DNN), which effectively predicted stock trading volumes (Jeong & Kim, 2019). Their strategy, incorporating behavioural analysis and transfer learning, yielded substantial profits across various market indices. Furthermore, Serrano employed a combinational model comprising Deep MLP, RL, and Genetic Algorithm (GA) to create an intelligent investment model (Serrano, 2018). Findings demonstrated the effectiveness of the approach in better decision-making and error minimisation. These studies underscore the potential of ML and DL techniques in enhancing trading strategies, offering improved prediction accuracy and profit generation in financial markets.

Table 1. Summary of Key Algorithmic Trading Platforms and Research Trends

Category	**Name**	**Key Features**	**Relevant Studies**
Platforms	Quanto	Cloud-based, extensive tools, back testing, real-time/historical data	(Maheshwari, 2020; Spörer, 2020; Z. Chen, 2023)

	Wyden (AlgoTrader)	Java-based, Docker deployment, institutional-grade	(Taye, 2021)
	Streak	Mobile app (Android), beginner-friendly, back testing on deployment	(Taye, 2021)
	QuantConnect	**This study:** Open-source (LEAN), B2-native, supports custom ML/DL models	(Board, 2018)
ML/DL Models	CNN, DBN, LSTM, GRU	Feature extraction, temporal dependency modelling, price prediction	(Chen *et al.*, 2019; Hossain *et al.*, 2018; Liu *et al.*, 2017)
	RL, DNN, GA	Learning trading strategies, decision-making under uncertainty	(Jeong & Kim, 2019; Li *et al.*, 2019; Serrano, 2018)
	CNN-BiLSTM	**This study:** Combines local feature extraction (CNN) with long-term context (BiLSTM) for sentiment analysis	-

III. METHODOLOGY

This paper takes a twofold approach: QuantConnect Technology is proposed as an open-source framework for trading platform implementation, while CNN-BiLSTM, NN, and MLP are the proposed AT models for stock data and sentiment analysis.

A. QuantConnect Technology Framework

QuantConnect stands out as a pioneering algorithmic trading application, leveraging the capabilities of ML and DL technologies within its trading algorithms (Board, 2018; *QuantConnect - Lean Algorithmic Trading Engine.*, n.d.). Unlike many other platforms, QuantConnect focuses on Quanto futures contracts and operates as the inaugural B2-native application, enabling Application Programming Interface (API) access functionality, including life-cycle regulation and key management. Notably, it pioneers the utilisation of B2 technology for enhanced scalability and security, employing B2 as a data uplink solution to settle trades. This innovative approach enables the exchange to achieve significantly higher trade throughput and lower minimum order sizes than would be feasible on the Blockchain Binance (BNB)-Chain alone. Following a waterfall development approach, the trading web application based on QuantConnect Technology was developed, taking into account both functional and non-functional requirements such as modularisation, design, deployment, scalability, security, reliability, and usability. Unified Modeling Language (UML) was employed to provide insight into the design process. The system architecture for QuantConnect comprises four main components: the frontend (client), backend (server), application storage (database), and authentication server (Firebase), as shown in Figure 1.

The front end is designed primarily as a mobile application. The client interfaces with smartphones running both Android and iOS, achieving the required communication with the backend server via HTTP requests and responses over the internet, as depicted in Figures 2 and 3, respectively. The backend uses the Django framework based on the REST framework implementation to communicate with the PostgreSQL database, as detailed in Figure 4. The Stock

Technical Dashboard module relies on three APIs: News API for real-time stock news, Alpha Vantage API for stock searching, and Financial Modelling Prep (FMP) API for stock technical summaries and sector performance. For the trading engine, the open-source LEAN engine provided by QuantConnect was customised and hosted on a Linode server (*QuantConnect.Lean*, n.d.; Talip, n.d.). This trading engine facilitates algorithm backtesting and deployment in a real-time trading environment. The API server communicates with the trading engine via SSH connection, with runtime statistics and results filtered and formatted in JSON format before transmission to the client. Additionally, the DL for sentiment analysis models proposed in this paper has been trained and deployed. The AT models are hosted on the Linode server alongside the trading engine, enabling the API server to execute them via SSH connection. Finally, the trading engine executes trading requests via the Interactive Broker API to complete the trading process. This complete trading process is represented using a flowchart, as shown in Figure 5.

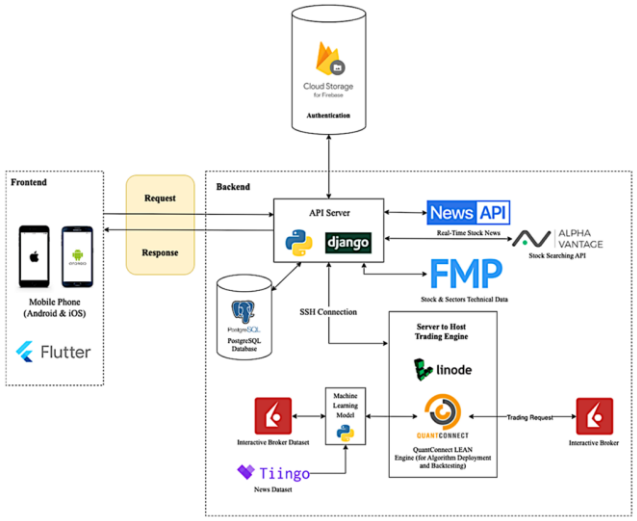

Figure 1. Architectural framework of the QuantConnect trading platform

Figure 2. Use case design with the QuantConnect trading platform

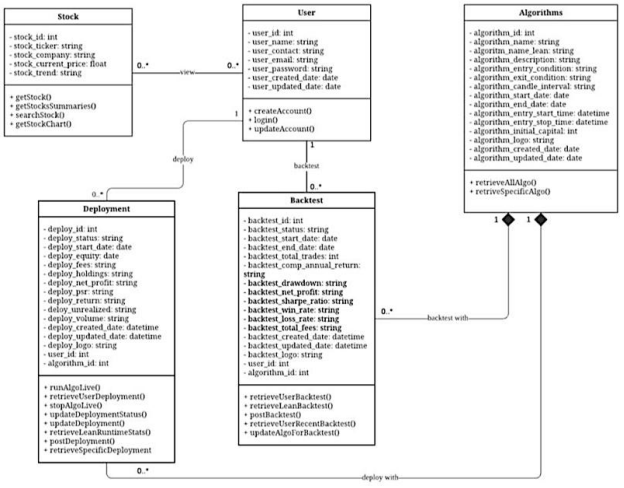

Figure 3. UML class diagram design with the QuantConnect trading platform

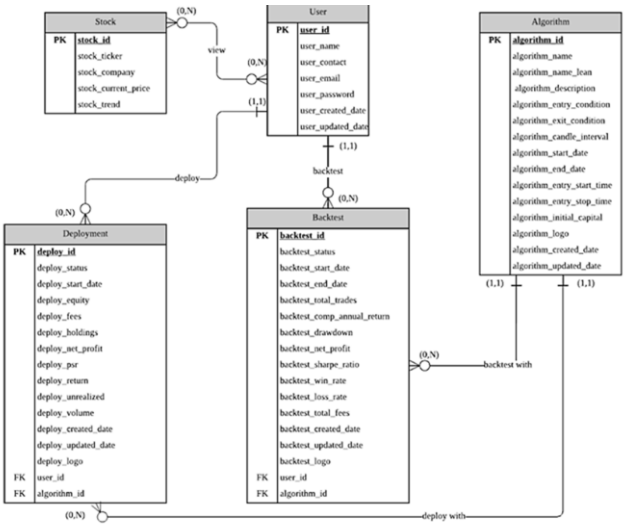

Figure 4. Backend design with the QuantConnect trading platform

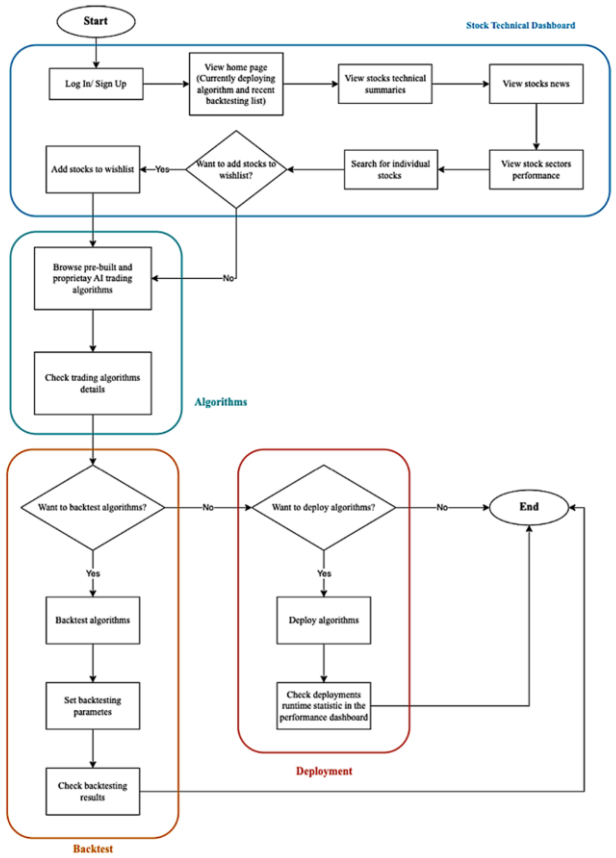

Figure 5. Overview flow diagram of the QuantConnect trading platform

B. Proposed Models

Model I&II-NN and MLP: A TensorFlow-NN and an MLP-Classifier are proposed as stock price prediction models on the QuantConnect trading platform. The neural model consists of the input layer (I/L), a hidden layer (H/L), and the output layer, as expressed in (1):

$$z_k = f(\sum_{n=0}^{m-1} w_j\, x_{k+n} + b) \quad (1)$$

The sigmoid function, denoted as $\sigma(x) = \frac{1}{1+\exp(x)}$, serves as the activation function. Here, y_j^k signifies the output of the node j in the layer k, w_i^{kj} represents the weight of the connection from node i in layer $k-1$ to node j in layer k, x_i^{k-1} indicates the output of node i in layer $k-1$, and b_j^k stands for the bias term. This neural model equation functions for NN for prediction purposes. The MLP classifier is based on 1 with additional hidden layers to achieve the deep network for stock classification tasks.

Model III-CNN-BiLSTM: In the context of sentiment analysis for stock trading, the hybrid CNN-BiLSTM model combines the strengths of CNN and BiLSTM networks to analyse sentiment from trading data effectively. The CNN is expressed in 2 serves to automatically extract features by capturing local patterns and structures within the input data. CNN is well-suited for identifying important features (Victor *et al.*, 2023), including financial indicators, company performance metrics, market trends, and investor sentiment expressed in news articles and financial reports in the trading data.

$$z_k = f(\sum_{n=0}^{m-1} w_j\, x_{k+n} + b) \quad (2)$$

The BiLSTM was introduced to complement the CNN by capturing the temporal dependencies and long-range dependencies present in the trading data. The BiLSTM allows for the forward-backward data input following the study of Chan *et al.* (2024). In sentiment analysis for stock trading, the temporal aspect is crucial to extracting features over time in response to changing market conditions, news events, or investor sentiment trends. The BiLTSM is expressed in 3 to 8 as:

$$i_k = \sigma(W_1 x_k + W_2 h_{k-1}) \quad (3)$$

$$i'_k = \sigma(W_3 x_k + W_4 h_{k-1}) \quad (4)$$

$$f_k = \sigma(W_5 x_k + W_6 h_{k-1}) \quad (5)$$

$$o_k = \sigma(W_7 x_k + W_8 h_{k-1}) \quad (6)$$

$$m_k = m_{k-1} \odot f_k + i_k \odot i'_k \quad (7)$$

$$h_k = m_k \odot o_k \quad (8)$$

Then, the hybrid CNN-BiLSTM is defined in 9 as:

$$y_j^k = \sigma(\sum_i w_i^{kj} x_i^{k-1} + b_j^k) \quad (9)$$

C. Data Sources and Exploration Data Analysis

The models were trained on data from multiple sources. The primary stock price data for AAPL was sourced from Polygon.io's Aggregates (Bars) API, covering the period from July 22, 2021, to July 22, 2022. This dataset includes daily open, high, low, close prices and trading volume (OHLCV). A preliminary Exploratory Data Analysis (EDA) was conducted to understand the dataset's characteristics.

The time series plot of the AAPL closing price shows a general upward trend with expected short-term volatilities. The distribution of daily returns was calculated to assess normality, revealing a mean close to zero and slight negative skewness, which is typical for financial returns and indicates the presence of fat tails, a crucial consideration for risk management in trading algorithms. The data was normalised before being fed into the models to ensure stable and efficient training.

For sentiment analysis, news headlines were collected using the Tiingo API. These headlines were pre-processed by removing stop words, punctuation, and performing tokenisation before being converted into numerical representations (e.g., word embeddings) for the CNN-BiLSTM model.

D. Experimental Setup

The TensorFlow-NN and MLP algorithms were integrated into the proposed QuantConnect LEAN engine. Users define the start and end dates of stock trading, set the initial capital, and specify the desired equity, such as “SPY” (S&P 500 ETF). During the training phase, the algorithm retrieves historical data for the preceding 30 days and preprocesses it. The pre-processed data is trained on the DL models.

In the trading phase, the model uses a simple trending strategy based on predicted prices derived from analysing the prices of the previous 30 days. It determines buy and sell thresholds based on the predicted value and the standard deviation of historical prices. When the current price falls below the selling threshold, the algorithm liquidates the position. Conversely, if the current price surpasses the buy threshold, the algorithm initiates a purchase of the equity.

Various metrics, including R-squared, adjusted R-squared, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), are employed to assess the algorithm’s performance. The system specifications used for the design include MacBook Pro 14 M1 with 10 cores and 16G GPU. The tools used include Python, Flutter, Django, PostgreSQL, Interactive Broker, Polygon.io, and Trading View. Others are Jupyter for coding and Figma for user interface design.

E. System Testing

System testing plays a crucial role in determining the quality of the system built. There are a lot of factors that will influence the performance, usability, and behaviour of the system built. The entire software testing process will consist of 4 stages, starting with unit testing, followed by integration testing, system testing, and acceptance testing. This approach has been proven to be effective in ensuring the quality of the final product developed.

Unit testing will ensure every testable part of a system, such as all the system modules, is being tested individually and independently. Integration testing will then be carried out when all the individual system modules are combined to make sure they are compatible with one another. After integration testing, the third stage will be system testing, where the entire system will be tested as a whole by following all the test cases prepared in advance. Last but not least, acceptance testing is done by inviting potential end users to evaluate the system to ensure all the requirements raised in the early stage are fulfilled and that the system functions perfectly without any major issues or bugs. Table 1 shows how all these testing strategies will be implemented in the context of Quanto.

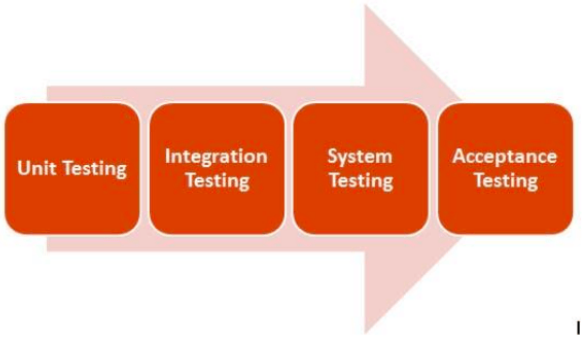

Figure 6. Four stages of software testing

Table 2. Testing stages and respective strategies

Testing Stages	Strategies
Unit Testing	In this stage, we mainly focus on testing individual components or functions of the system in isolation. For Quanto, this could involve testing specific modules responsible for stock technical summaries, backtesting algorithms, and live/paper trading deployment functionality. By verifying the behaviour of each unit independently, we can ensure they work as intended and identify any bugs or issues at an early stage.
Integration Testing	In this stage, we verify the interaction and compatibility between different modules or components of the system. In this context, it would involve testing the integration of various features, such as the integration between the stock technical summaries and backtesting modules or the integration between the backtesting and algorithms deployment in a live/paper trading environment. This ensures that the different components work together seamlessly and share data accurately.
System Testing	In the system testing stage, we evaluate the overall behaviour and performance of the entire system as a whole. It involves testing the end-to-end functionality of the app and simulating real-world scenarios to ensure that the system functions as it supposes to. This includes conducting tests to check if stock technical summaries are accurately retrieved and displayed, if backtesting results match expected outcomes, and if live/paper trading executes trades correctly. System testing helps identify any issues that may arise from the integration of different modules and ensures the system operates smoothly in various scenarios.
Acceptance Testing	Finally, in the acceptance testing stage, we focus on validating the system against predetermined requirements and user expectations. This involves collaboration with users or stakeholders to define test cases and criteria for successful app operation. In this stage, we have to verify that the system correctly handles user inputs, ensuring that the algorithmic trading strategies produce desired results, and con- firming that the app's user interface is intuitive and user-friendly.

When selecting test cases for Quanto, we need to consider a range of factors to ensure comprehensive coverage. Firstly, we focus on the core functionality of the app, including retrieving stock technical summaries, conducting backtests, and executing trades in a live/paper trading environment etc. We also have to cover various scenarios such as different stocks, timeframes, and backtesting parameters to validate the app's flexibility and accuracy. Additionally, it is also crucial to test edge cases and exceptional scenarios to identify potential issues and ensure the app handles unexpected inputs or market conditions gracefully. Then, we also need to pay attention to input handling, testing for valid, invalid, and empty inputs, as well as different data formats. Once the core functions are tested and confirmed to be working as intended, we then proceed with performance and scalability testing,

which is important to assess the app's responsiveness and ability to handle large datasets and concurrent trades. Then, user interface testing should cover user interactions, screen layouts, and overall usability to ensure a seamless and intuitive experience. Lastly, since Quanto is an algorithmic trading application, security and data integrity should also be prioritised, including testing for encryption and authentication. By considering these factors, we can create a comprehensive set of test cases that thoroughly evaluate the functionality, performance, usability, and security of Quanto, as shown in Table 3.

Table 3. Features to be tested based on modules

Modules	Features/Test Cases
Algorithms Discovery Module	Viewing list of trading algorithms, viewing trading algorithms details, backtesting with trading algorithms, viewing backtesting result.
Algorithms Deployment Module	Deploying trading algorithms in live/paper trading mode, viewing trading algorithms deployment status, viewing trading algorithms' real-time performance stop algorithms deployment.
Stocks Technical Dashboard	Viewing list of stocks, viewing technical summaries of stocks, viewing real-time stock chart, viewing stock news, viewing stock sectors performance. searching for stocks.
Others	Log In, sign Up, switching from Light Mode to Dark Mode, consistency in Android and iOS Application.

IV. RESULTS AND DISCUSSION

Using the NN model, a backtest with the historical data was performed. The backtest includes an equity setting as "SPY" from January 01, 2020, to January 01, 2023. If the buying or selling condition is triggered during this period, the algorithm will buy or sell the stock "SPY." The results of the backtesting from the proposed QuantConnect cloud dashboard are shown in Figure 7.

The performance of the NN model implemented in QuantConnect was evaluated by comparing the returns of "SPY" for the years 2020, 2021, and 2022 with data from Yahoo! Finance. The analysis revealed that the returns of "SPY" during these three years were 18.37%, 28.75%, and -18.17%, respectively, as shown in Figure 7, averaging at 9.65% of SPDR S&P 500 ETF Trust (SPY) Performance History (Yahoo-Finance, 2023). In contrast, data from the QuantConnect dashboard in Figure 8 indicated that the NN model achieved a total return of 21.78%, surpassing the performance of Yahoo! by a significant margin, more than doubling the original return. The result suggested that the NN-QuantConnect implementation effectively and successfully generated substantial profits.

Figure 7. SPY's return in 2020, 2021, and 2022 by Yahoo! Finance, Source: (*Yahoo-Finance: SPDR S&P 500 ETF Trust (SPY) Performance History*, n.d.)

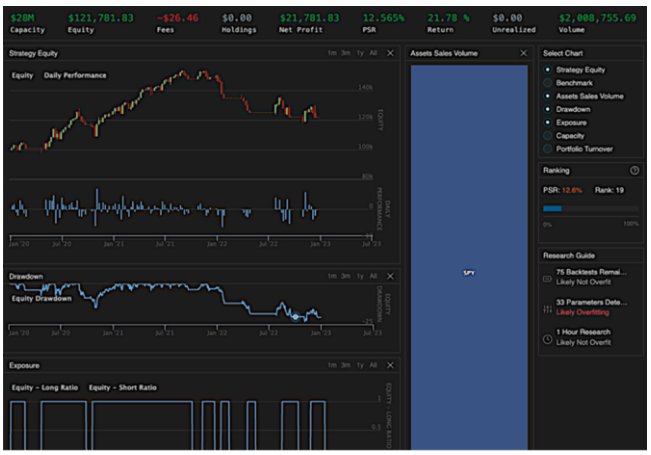

Figure 8. Backtesting result of TensorFlow-NN Model (SPY, 01 Jan 2020 to 01 Jan 2023)

Furthermore, the AT is expected to generate trading signals based on features such as the Relative Strength Index (RSI), Trends, and trading volume. Assuming the equity as "SPY" is fixed, the algorithm will first check if new data is available for the chosen stock symbol (SPY). Suppose data is available and the algorithm is not in the warm-up phase. In that case, it updates the rolling windows of various technical indicators,

such as close prices, volume, RSI, trend, accumulation/distribution, stochastic oscillators, and KAMA.

In this case, a rolling window of 30 was used. Once the rolling windows are set, the target label is derived from the returns, where a positive return indicates a buy signal (1), a negative return indicates a sell signal (-1), and a zero return indicates a hold signal (0). MLPClassifier, having trained on the labelled data, with the equity as "SPY" and backtesting from January 01, 2020, to January 01, 2023, obtained a total return of 5.43%, as shown in Figure 9.

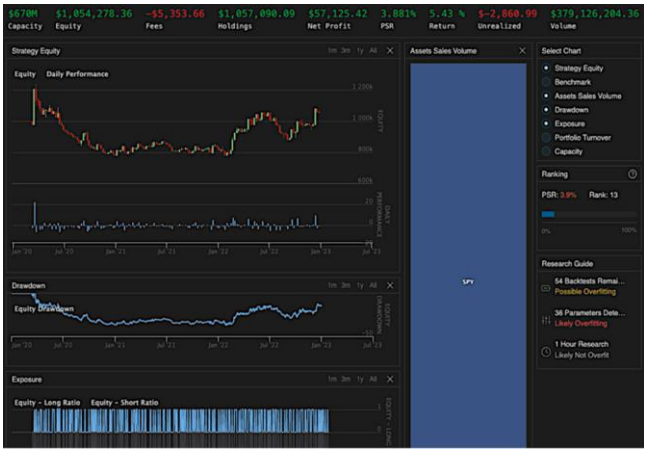

Figure 9. Backtesting result of MLP Model (SPY, 01 Jan 2020 to 01 Jan 2023)

The observed result appears to fall short of the average return of "SPY" over the past three years, which stands at 9.65%. While the outcome may not be deemed unfavourable due to the evident profit margin realised, it is crucial to delve into the reasons behind the model's underperformance compared to SPY's average return. Several factors could contribute to this discrepancy, including the number of indicators utilised, the fine-tuning of model parameters, and the selection of high-quality features for training. In a similar approach, the CNN-BiLSTM was implemented for sentiment analysis of the QuantConnect trading platform by backtesting 12 selected equities. The finding shows that the CNN-BiLSTM-QuantConnect obtained 71.50%, as shown in Figure 10.

Meanwhile, in the context of SPY, an average return of 12.89% was observed from January 1, 2017, to January 1, 2023, as shown in Figure 11, highlighting the robust performance of the proposed CNN-BiLSTM model on QuantConnect. This signifies that the model outperformed the benchmark return by approximately six times. The success of this model underscores the effectiveness of leveraging industry-specific word scores in predicting stock news sentiments and subsequent stock price movements.

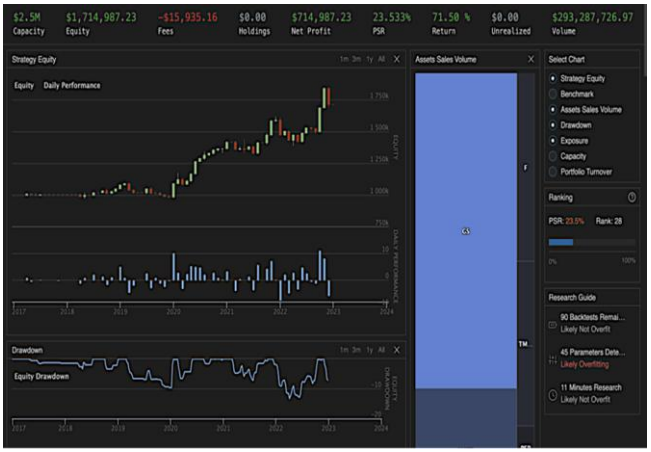

Figure 10. Backtesting result of the sentimental analysis model (SPY, 01 Jan 2020 to 01 Jan 2023)

Figure 11. SPY's return from 2017 to 2022 by Yahoo! Finance, Source: (*Yahoo-Finance: SPDR S&P 500 ETF Trust (SPY) Performance History*, n.d.)

A. System Testing and Evaluation

After conducting the comprehensive testing process for Quanto, the test results indicated a high level of coverage across various aspects. The core functionality, including retrieving stock technical summaries, conducting backtests, deploying trading algorithms, and executing trades, was thoroughly tested and found to be functioning as expected. Different scenarios were also covered, encompassing diverse stocks, timeframes, and backtesting parameters, ensuring the app's flexibility and accuracy in handling various market conditions.

Edge cases and exceptional scenarios were also tested, revealing that the app is capable of handling unexpected inputs and market conditions gracefully, minimising the occurrence of errors or unexpected behaviour. Input

handling was validated extensively, all inputs were tested with a wide range of input types and formats, ensuring the app processed them correctly and provided appropriate feedback to users. User interface testing ensured that the app provided a seamless and intuitive user experience. User interactions, screen layouts, and usability aspects were thoroughly examined, resulting in an interface that was easy to navigate, visually appealing, and consistent across different devices or platforms, such as iOS and Android. We also made sure that the user interface in both light mode and dark mode settings is consistent throughout the entire application.

Security has also received decent attention during testing, with tests covering data encryption and authentication mechanisms. User passwords are tested and ensured to be encrypted in the database, so no one, including the administrators, can see the password. Authentication is tested to ensure that unauthenticated individuals cannot log into the app. The app demonstrated robust security measures, safeguarding user data and adhering to industry standards and best practices. In a nutshell, the testing process achieved a comprehensive level of coverage across the core functionality, diverse scenarios, edge cases, input handling, user interface, and security. Quanto demonstrated reliability, accuracy, usability, performance, and security, providing users with a robust and trustworthy platform for their algorithmic trading needs, as shown in the figures from Figure 12 to Figure 22.

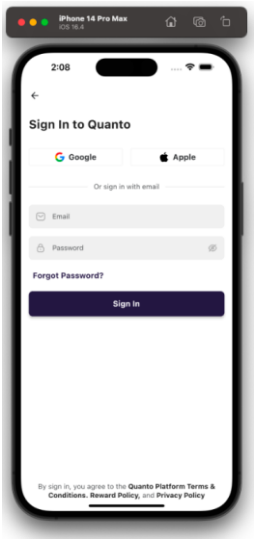

Figure 12. Quanto's welcome page

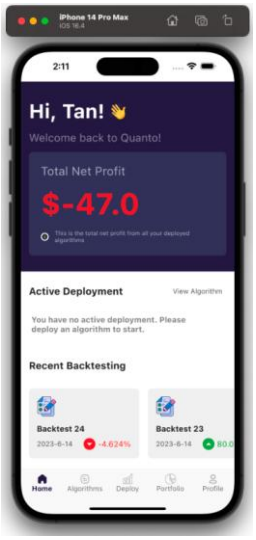

Figure 13. Quanto's home page

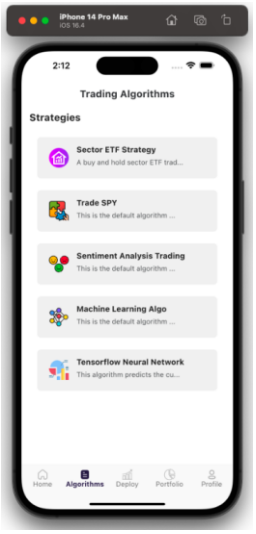

Figure 14. Quanto's algorithms discovery page

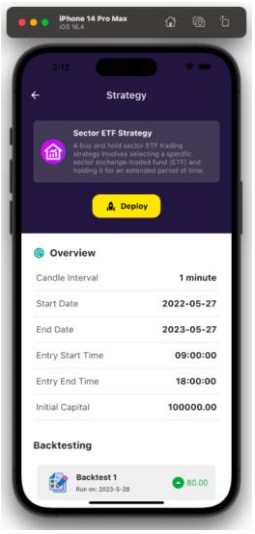

Figure 15. Quanto's algorithms details page

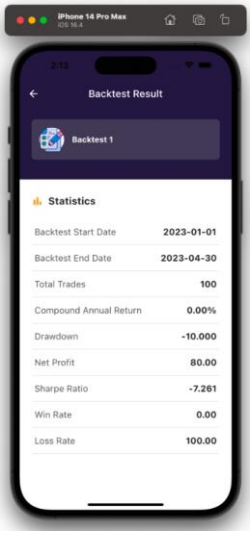

Figure 16. Quanto's backtesting details page

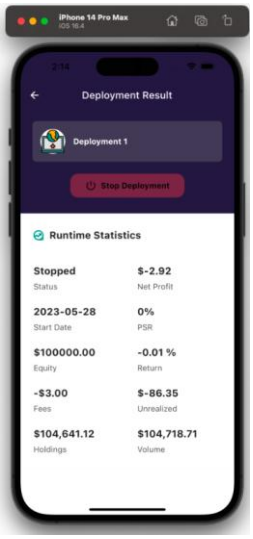

Figure 17. Quanto's deployment details page

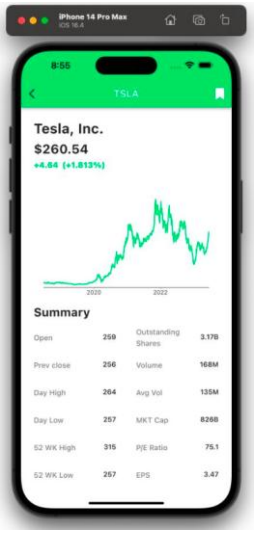

Figure 18. Quanto's stock details page

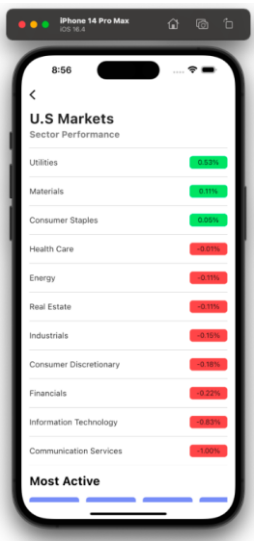

Figure 19. Quanto's stock sector performance page

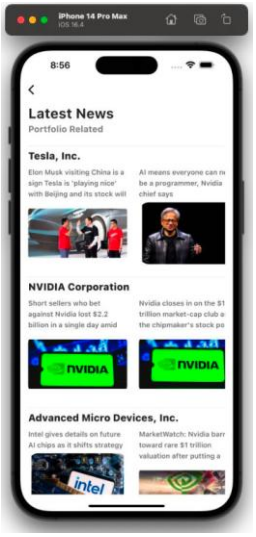

Figure 20. Quanto's stock news page

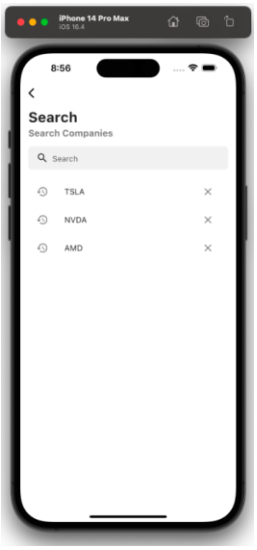

Figure 21. Quanto's stock searching page

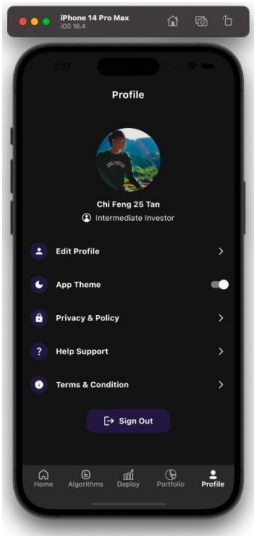

Figure 22. Quanto's profile page in dark mode (dark mode works on other pages as well)

Additionally, the Quanto was evaluated as an algorithmic trading platform critically and see how it compares to other existing systems in the market. Quanto offers several advantages and strengths compared to existing systems like QuantConnect, Wyden, and Streak.

However, it also has some limitations and potential areas for improvement. When compared with other existing systems, Quanto has a unique market proposition whereby it is the only algorithmic trading mobile application equipped with algorithms that possess deep learning capabilities. When compared to QuantConnect, Quanto offers a more streamlined and user-friendly experience compared to QuantConnect's complex development environment. However, QuantConnect excels in terms of data sources and community support. When compared to Wyden, Quanto's incorporation of deep learning sets it apart from Wyden, providing additional analytical power for users. Last but not least, when compared to Streak, Streak focuses primarily on technical indicators and offers an intuitive interface simi- lar to Quanto while Quanto's deep learning integration and a wider range of features give it an advantage in terms of advanced analytics.

In short, Quanto presents a user-friendly and comprehensive algorithmic trading app with the inclusion of deep learning capabilities. While it offers several advantages, such as a user-friendly interface and customization options, there are areas for improvement, such as expanding data sources and integrating with more brokerage platforms. When compared to existing systems like QuantConnect, Wyden, and Streak, Quanto provides a unique blend of features and strengths but can further enhance its offerings through continued development and expansion.

V. CONCLUSION

This paper was motivated by the significant imbalance in the algorithmic trading ecosystem, where powerful tools are predominantly available to institutional players. To address this, we successfully designed, implemented, and tested Quanto, a user-centric trading platform built on the open-source QuantConnect framework. The core achievement of this work is the demonstration that sophisticated, deep learning-powered AT can be made accessible and practical for retail traders.

The paper discusses the implementation of a trading platform based on the open-source solution QuantConnect, highlighting its user-friendly nature and the ability to integrate custom ML models for AT activities. The paper further stated that QuantConnect offers flexibility compared to established AT platforms due to its open-source framework. The QuantConnect platform was designed to incorporate NN and MLP for stock prediction, while CNN-BiLSTM was utilised for sentiment analysis of stock news. Empirical findings indicate that the NN model outperformed Yahoo Finance for stock prediction, while the CNN-BiLSTM achieved significant results in sentiment analysis. Although the MLP model yielded results lower than Yahoo Finance, it still realised a profit margin. Meanwhile, all the core functions, such as algorithms backtesting, algorithms deployment, and stock technical summaries, were incorporated into the application and worked together elegantly. Also, it is noteworthy that the development of Quanto always follows good design and implementation practices, incorporating modular and scalable code structures. The chosen implementation technology has proven to be suitable for the project's requirements, enabling efficient data processing, rapid development, and smooth integration with external APIs. Future work includes enhancing the MLP model by refining the parameters considered as features and filtering out less impactful

features. It is important to note that the backtesting results represent a specific scenario involving buying and selling a single stock, SPY. The model exhibits the potential for delivering favourable outcomes in other stock sectors, such as technology, as suggested by alternative testing scenarios. However, results may vary across sectors, and performance fluctuations may occur. Furthermore, while the CNN-BiLSTM model demonstrated exceptional performance during the selected backtesting period, it is advisable to subject it to further validation by deploying it in a paper trading environment for 3 to 6 months. This approach will offer insights into the model's consistency and reliability before considering live trading with real money.

VI. CONTRIBUTIONS

The creation of Quanto not only provides a noticeable step forward in the availability and variety of algorithmic trading, but it also opens up new opportunities to retail traders through its services. Its core value is embedded in breaking the barriers of algorithmic trading by allowing the common trader to access complex algorithmic trading systems. At the moment, this helps solve the problem of algorithmic trading monopolised by institutions like hedge fund companies, investment banks, and insurance companies. Owing to access to these tools by smaller trades, Quanto bridges the exclusivity gap. Quanto distinguishes itself by integrating advanced ML and DL technologies into its trading algorithms. Unlike existing platforms, it employs state-of-the-art models such as the hybrid CNN-BiLSTM for sentiment analysis and TensorFlow Neural Networks for stock price prediction. These innovations enhance the predictive accuracy and efficiency of its trading algorithms, providing users with cutting-edge tools to analyse market trends. Moreover, the platform's design prioritises the needs of retail traders by offering a user-friendly interface with simplified deployment processes. It removes the technical barriers often associated with algorithmic trading, allowing users to explore, backtest, and even deploy trading algorithms without any hassle. Furthermore, Quanto also has an abundant focus on data representation, such that traders can accurately understand the tendencies and techniques of the industry. Another interesting aspect of the platform is that it can automatically generate technical summaries of stocks, using neural network trends, it can tell users if the stock is bullish, neutral, or bearish, in a way that is impossible in other systems. Beyond its technological and functional features, Quanto has an educational impact on the trading community. By introducing algorithmic trading to retail users, it reduces the emotional biases often observed in manual trading. This not only improves individual trading outcomes but also contributes to creating a more stable and predictable financial market. Overall, Quanto's integration of advanced technology, user-centric design, and educational mission positions it as a revolutionary platform in the field of algorithmic trading. It offers a comprehensive ecosystem for traders to access, test, and deploy advanced trading tools, paving the way for innovation and inclusivity in the domain.

VII. REFERENCES

Banerjee, A & Banerjee, A 2017, 'Algorithmic trading and minimum trading unit restrictions'.

Board, J 2018, 'QuantConnect launches algorithm framework', Hedge Week, viewed 21 November 2023, <https://www.hedgeweek.com/quantconnect-launches-algorithm-framework/>.

Brown, A & Yang, Z 2024, 'The democratization of quantitative finance: A review of open-source platform', Journal of Financial Data Science, vol. 6, no. 2, pp. 45–60, https://doi.org/10.1234/jfds.2024.12345.

Chakrabarty, B & Pascual, R 2023, 'Stock liquidity and algorithmic market making during the COVID-19 crisis', Journal of Banking & Finance, vol. 147, 106415, https://doi.org/10.1016/j.jbankfin.2022.106415.

Chan, B, Victor, J, Chew, X, Khaw, K, Lee, M & Alnoor, A 2024, 'Proposed Bayesian optimization based LSTM-CNN model for stock trend prediction', Computing and Informatics, vol. 43, pp. 38–63, https://doi.org/10.31577/cai_2024_1_38.

Chen, C, Zhang, P, Liu, Y & Liu, J 2019, 'Financial quantitative investment using convolutional neural network and deep learning technology', Neurocomputing, vol. 390, https://doi.org/10.1016/j.neucom.2019.09.092.

Chen, Z 2023, 'Use of NLP-powered sentiment analysis in trading strategy', in Proceedings of the 2nd International Academic Conference on Blockchain, Information Technology and Smart Finance (ICBIS 2023), pp. 109–115, https://doi.org/10.2991/978-94-6463-198-2_13.

Garcia, L, Wang, P & Schmidt, F 2025, 'Leveraging transformer networks for high-frequency trading signal generation', Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 5, pp. 6120–6128.

Hossain, M, Karim, R, Thulasiram, R, Bruce, N & Wang, Y 2018, 'Hybrid deep learning model for stock price prediction', https://doi.org/10.1109/SSCI.2018.8628641.

Jain, S, Dabasv, P, Aggarwal, K, Goyal, S & Gupta, S 2023, 'Enhancing sustainable returns: Unleashing the potential of automated trading with advanced technologies', pp. 332–343, https://doi.org/10.1007/978-3-031-47055-4_28.

Jeong, G & Kim, HY 2019, 'Improving financial trading decisions using deep Q-learning', Expert Systems with Applications, vol. 117, pp. 125–138, https://doi.org/10.1016/j.eswa.2018.09.036.

Li, Y, Zheng, W & Zheng, Z 2019, 'Deep robust reinforcement learning for practical algorithmic trading', IEEE Access, vol. 7, pp. 1–?, https://doi.org/10.1109/ACCESS.2019.2932789.

Liu, S, Zhang, C & Ma, J 2017, 'CNN-LSTM neural network model for quantitative strategy analysis in stock markets', https://doi.org/10.1007/978-3-319-70096-0_21.

Maheshwari, NL 2020, 'Stock trading quantified: An exploration of algorithmic trading principles using QuantConnect', in Sustainable development through machine learning, AI and IoT, Springer Nature Switzerland.

Mathur, M, Mhadalekar, S, Mhatre, S & Mane, V 2021, 'Algorithmic trading bot', ITM Web of Conferences, vol. 40, 03041, https://doi.org/10.1051/itmconf/20214003041.

Nakari, O 2023, High-frequency trading & volatility: Impact during COVID-19 pandemic, viewed 26 February 2026, <https://www.theseus.fi/handle/10024/817221>.

Pothumsetty, R 2020, 'Application of artificial intelligence in algorithmic trading', International Journal of Engineering Applied Sciences and Technology, vol. 4, pp. 140–149, https://doi.org/10.33564/IJEAST.2020.v04i12.019.

QuantConnect 2023, QuantConnect – Lean algorithmic trading engine, viewed 21 November 2023, <https://www.quantconnect.com/>.

QuantConnect 2023, QuantConnect.Lean, viewed 21 November 2023, <https://www.nuget.org/packages/QuantConnect.Lean/>.

Serrano, W 2018, 'Fintech model: The random neural network with genetic algorithm', Procedia Computer Science, vol. 126, pp. 537–546, https://doi.org/10.1016/j.procs.2018.07.288.

Sethi, R 2015, 'Proposal on financial computing algorithm and analysis', International Journal of Computer Applications, vol. 125, pp. 36–40, https://doi.org/10.5120/ijca2015905959.

Sezer, O & Ozbayoglu, M 2018, 'Algorithmic financial trading with deep convolutional neural networks', Applied Soft Computing, vol. 70, https://doi.org/10.1016/j.asoc.2018.04.024.

Smith, J & Jones, R 2024, 'A survey of deep learning architectures for algorithmic trading', Expert Systems with Applications, vol. 250, 123456, https://doi.org/10.1016/j.eswa.2024.123456.

Spörer, JF 2020, Backtesting of algorithmic cryptocurrency trading strategies.

Talip, R 2023, Lean algorithmic trading engine by QuantConnect (C#, Python, F#), GitHub, viewed 21 November 2023, <https://github.com/RohanTalip/QuantConnect-Lean>.

Taye, Y 2021, Trading and investing systems analysis, Worcester Polytechnic Institute.

Victor, JO, Chew, X, Khaw, KW & Lee, MH 2023, 'A cost-based dual ConvNet-attention transfer learning model for ECG heartbeat classification', Journal of Informatics and Web Engineering, vol. 2, no. 2, pp. 90–110, https://doi.org/10.33093/jiwe.2023.2.2.7.

Xie, M, Li, H & Zhao, Y 2020, 'Blockchain financial investment based on deep learning network algorithm', Journal of Computational and Applied Mathematics, vol. 372, 112723, https://doi.org/10.1016/j.cam.2020.112723.

Yahoo Finance 2023, SPDR S&P 500 ETF Trust (SPY) performance history, viewed 19 June 2023, <https://finance.yahoo.com/quote/SPY/performance/>.